Studies in Computational Intelligence 556

Pramod Kumar Pisharady
Prahlad Vadakkepat
Loh Ai Poh

Computational Intelligence in Multi-Feature Visual Pattern Recognition

Hand Posture and Face Recognition using Biologically Inspired Approaches

Springer

Studies in Computational Intelligence

Volume 556

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Pramod Kumar Pisharady
Prahlad Vadakkepat · Loh Ai Poh

Computational Intelligence in Multi-Feature Visual Pattern Recognition

Hand Posture and Face Recognition using Biologically Inspired Approaches

Springer

Pramod Kumar Pisharady
Institute of High Performance Computing, A*STAR
Singapore
Singapore

Prahlad Vadakkepat
Loh Ai Poh
Electrical and Computer Engineering
National University of Singapore
Singapore
Singapore

ISSN 1860-949X
ISBN 978-981-287-055-1
DOI 10.1007/978-981-287-056-8
Springer Singapore Heidelberg New York Dordrecht London

ISSN 1860-9503 (electronic)
ISBN 978-981-287-056-8 (eBook)

Library of Congress Control Number: 2014938213

© Springer Science+Business Media Singapore 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Dedicated in Loving Memories to Our Beloved Mothers

Preface

Visual pattern recognition systems have wide applications in surveillance, video gaming, human–robot interaction, and virtual reality. High computational complexity, abundance of pattern features, sensitivity to size and shape variations, and poor performance against complex backgrounds are prominent issues related to robust recognition of visual patterns. A collection of computational intelligence algorithms addressing these issues in visual pattern recognition is provided in this book. The book is positioned as a reference in the field of computer vision and pattern recognition.

Image feature extraction, feature selection, and classification are different stages in visual pattern recognition. Efficiency of pattern recognition algorithm depends on the individual efficiencies of these stages. The focus of Part II of the book is on feature selection and classification aspects, whereas Part III focuses on feature extraction algorithms. The main application area considered is hand posture recognition. In addition, the book discusses the utility of the algorithms in other visual as well as nonvisual pattern recognition tasks including face recognition, general object recognition, and cancer/tumor classification. The feature selection and classification algorithms presented in Part II are useful for predictive gene identification and classification of cancer/tumor. The experiments with gene expression-based cancer and tumor datasets have shown the utility of the algorithms in multi-feature classification problems in biomedical field.

The book contains eight chapters, divided into three parts. Part I (Chaps. 1–3) contains the necessary background to understand the material presented. Chapter 1 introduces the book with a discussion on various research issues in the field. Chapter 2 describes the computational intelligence tools utilized. A comprehensive survey of the related literature is provided in Chap. 3. Part II (Chaps. 4–6) focuses on computational intelligence algorithms for pattern recognition. Two fuzzy-rough set-based feature selection and classification algorithms applicable to visual as well as nonvisual pattern recognition tasks are presented in Chaps. 4 and 5. The algorithms are named as fuzzy-rough single cluster (FRSC) and multi-cluster (FRMC) classification algorithms. The FRSC and FRMC algorithms are discriminative and fast in feature selection and classification of multiple feature datasets. Applications include image pattern recognition and cancer classification. Chapter 6 describes a boosting-based fuzzy-rough multi-cluster (BFRMC) classification algorithm, combining the fuzzy-rough approach with genetic

algorithm based on iterative rule learning and boosting. Part III (Chaps. 7 and 8) presents algorithms for hand posture recognition using neurobiologically inspired approaches. Computational model of visual cortex-based algorithms are presented in Chap. 7, addressing the problems in hand posture recognition. The visual cortex model-based features have invariance with respect to appearance and size of the hand, and provide good inter-class discrimination. Chapter 8 presents an attention-based segmentation and recognition (ASR) algorithm combining the model of visual cortex with a Bayesian model of visual attention, addressing the complex background problem in hand posture recognition.

The book provides performance comparisons for the algorithms outlined, with existing standard methods in machine learning.

Singapore, January 2014

Pramod Kumar Pisharady
Prahlad Vadakkepat
Loh Ai Poh

Contents

Part I Computational Intelligence in Visual Pattern Recognition

1 Visual Pattern Recognition 3
1.1 Introduction 3
1.2 Overview 4
1.2.1 The Visual Pattern Recognition Pipeline 4
1.3 Hand Gestures: Variation in Appearance, Large Number of Features and Complex Backgrounds 6
1.4 The Algorithms: Fuzzy-Rough Classifier, Biologically Inspired Feature Extraction and Visual Attention. 7
References 8

2 Computational Intelligence Techniques 11
2.1 Fuzzy and Rough Sets 11
2.1.1 Fuzzy-Rough Sets 13
2.1.2 Feature Selection and Classification Using Fuzzy-Rough Sets 13
2.1.3 Genetic Algorithm. 14
2.2 Computational Model of Visual Cortex 15
2.2.1 Biologically Inspired Feature Extraction System 16
References 17

3 Multi-Feature Pattern Recognition 21
3.1 Feature Selection and Classification of Multi-Feature Patterns 21
3.2 Cancer Classification 22
3.3 Face Recognition. 22
3.4 Hand Gesture Recognition 23
3.4.1 Hand Gesture Recognition Techniques 24
3.4.2 Hand Gesture Databases 32
3.4.3 Comparison of Methods. 32
References 34

Part II Feature Selection and Classification

4 Fuzzy-Rough Discriminative Feature Selection and Classification . . . 41
4.1 Introduction . . . 42
4.2 Fuzzy-Rough Single Cluster Feature Selection and Classification Algorithm . . . 42
4.2.1 Training Phase: Discriminative Feature Selection and Classifier Rules Generation . . . 43
4.2.2 The Testing Phase: The Classifier . . . 49
4.2.3 Computational Complexity Analysis . . . 50
4.3 Performance Evaluation and Discussion . . . 52
4.3.1 Cancer Classification . . . 52
4.3.2 Image Pattern Recognition . . . 57
4.4 Summary . . . 59
References . . . 60

5 Hand Posture and Face Recognition Using Fuzzy-Rough Approach . . . 63
5.1 Introduction . . . 64
5.2 The Fuzzy-Rough Multi Cluster Classifier . . . 64
5.2.1 Training Phase: Identification of Feature Cluster Centers and Generation of Classifier Rules . . . 65
5.2.2 Genetic Algorithm Based Feature Selection . . . 69
5.2.3 FRMC Classifier Testing Phase . . . 72
5.2.4 Computational Complexity Analysis . . . 72
5.3 Experimental Evaluation of FRMC Classifier . . . 72
5.3.1 Face Recognition . . . 74
5.3.2 Hand Posture Recognition . . . 76
5.3.3 FRMC Classifier Online Implementation . . . 78
5.4 Summary . . . 79
References . . . 79

6 Boosting Based Fuzzy-Rough Pattern Classifier . . . 81
6.1 Introduction . . . 81
6.2 Fuzzy-Rough Sets for Classification . . . 82
6.3 Boosting Based Fuzzy-Rough Multi-cluster Classifier . . . 83
6.3.1 Stage 1: Membership Functions from Cluster Points . . . 84
6.3.2 Stage 2: Generation of Certain Rules from Membership Functions . . . 84
6.3.3 Stage 3: Generation of Possible Rules from Membership Functions . . . 86

6.4 Performance Evaluation and Discussion 89
6.5 Conclusion 90
References 91

Part III Biologically Inspired Approaches in Hand Posture Recognition

7 Hand Posture Recognition Using Neuro-Biologically Inspired Features 95
7.1 Introduction 95
7.2 Graph Matching Based Hand Posture Recognition: C1 Features 96
7.2.1 Elastic Graph Matching Based Hand Posture Classifier 96
7.2.2 Experimental Results 98
7.2.3 Summary: Graph Matching Based Hand Posture Recognition Algorithm 100
7.3 Feature Extraction and Selection for Hand Posture Recognition: C2 Features 101
7.3.1 Discriminative Feature Extraction and Selection 101
7.3.2 Real-time Implementation and Experimental Results 103
7.3.3 Summary: Feature Extraction and Selection for Hand Posture Recognition 105
References 105

8 Attention Based Segmentation and Recognition Algorithm for Hand Postures Against Complex Backgrounds 107
8.1 Introduction 108
8.2 The Feature Extraction System and the Model of Attention 110
8.2.1 Extraction of Shape and Texture Features 110
8.2.2 Bayesian Model of Visual Attention 112
8.3 Attention Based Detection, Segmentation and Recognition 114
8.3.1 Image Pre-processing 114
8.3.2 Extraction of Color, Shape and Texture Features 117
8.3.3 Feature Based Visual Attention and Saliency Map Generation 119
8.3.4 Hand Segmentation and Classification 121
8.4 Experimental Results and Discussion 123
8.4.1 The Dataset and the Experimental Set-up 123
8.4.2 Hand Posture Detection 124
8.4.3 Hand Region Segmentation 124
8.4.4 Hand Posture Recognition 125

8.4.5 Performance with Human Skin and Body Parts as Noises 127
8.4.6 Comparison of the Recognition Time 128
8.4.7 Recognition of Hand Postures with Uniform Backgrounds. 129
8.5 Summary 129
8.5.1 Possible Extension and Application of ASR Algorithm. 130
References 130

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 133

Index 137

Acronyms

ASR	Attention-based Segmentation and Recognition
BFRMC	Boosting-based Fuzzy-Rough Multi-Cluster
BNT	Bayes Net Toolbox
DLLE	Distributed Locally Linear Embedding
EGM	Elastic Graph Matching
FRMC	Fuzzy-Rough Multi-Cluster
FRSC	Fuzzy-Rough Single Cluster
GA	Genetic Algorithm
HCI	Human–Computer Interaction
HMM	Hidden Markov Model
HRI	Human–Robot Interaction
IOHMM	Input Output Hidden Markov Model
LLE	Locally Linear Embedding
MC	Margin of Classification
PNN	Probabilistic Neural Network
RBF	Radial Basis Function
RIL	Recursive Induction Learning
RNN	Recurrent Neural Networks
SCFG	Stochastic Context Free Grammar
SLLE	Supervised Locally Linear Embedding
SMF	Standard Model Features
SVM	Support Vector Machines

Part I
Computational Intelligence in Visual Pattern Recognition

Chapter 1
Visual Pattern Recognition

A mathematician, like a painter or poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas

G. H. Hardy

Abstract An overview of the visual pattern recognition process and associated key issues are presented in this chapter. The varying scales and shapes, inter-class similarity, large number of features, and complex backgrounds are issues related to visual pattern recognition. The book focuses on these issues. The chapter introduces different algorithms addressing these issues.

Keywords Visual pattern recognition · Feature selection · Classification · Hand posture recognition

1.1 Introduction

Visual pattern recognition systems have vast applications in surveillance, interactive systems, video gaming, and virtual reality. Visual interaction is non-contact in type and is not affected by audible noise. Image feature extraction, feature selection, and classification are the different stages in visual pattern recognition. The efficiency of the overall system depends on the efficiencies of the individual stages.

Recognition of visual patterns for real world applications is a complex process. Varying and complex backgrounds, badly lit environments, person dependence, and the computational costs are some of the associated issues. Addressing these issues reliably and efficiently in realistic settings makes visual pattern recognition research challenging and demanding. The release of the motion sensing device Kinect [1] by Microsoft demonstrates the utility of tracking and recognition of human gestures in entertainment.

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_1,
© Springer Science+Business Media Singapore 2014

Real world application of visual pattern recognition systems requires algorithms with invariance to size and shape, good real-time performance, and robustness to cluttered and complex backgrounds. Developing robust algorithms is challenging as there are several unresolved issues. The algorithms presented in this book address issues in multi-feature pattern recognition with a focus on visual patterns. Hand gesture recognition is the main application considered. The algorithms are applied to other visual (face and object) and non-visual (cancer and tumor) pattern recognition tasks as well.

1.2 Overview

The visual processing and pattern recognition capabilities of the primate brain is yet to understand well. The human visual system rapidly and effortlessly recognizes 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 extend the capabilities of conventional pattern recognition systems. For example, the selectivity and invariance aspects of the visual cortex model are useful in improving the hand gesture recognition algorithms. The visual cortex model based features are invariant to size and appearance of the hand, leading to person independent performance.

1.2.1 The Visual Pattern Recognition Pipeline

The main stages in a visual pattern recognition task involves feature extraction and selection, and pattern classification. A typical visual pattern recognition pipeline is shown in Fig. 1.1.

The book presents computationally efficient and accurate pattern recognition algorithms. Areas of applications considered include hand gesture and face recognition, and cancer classification. Novel algorithms which utilize computational intelligence techniques are introduced. Fuzzy-rough feature selection and classification algorithms, and biologically inspired feature extraction algorithms are presented.

1.2.1.1 Feature Selection and Classification

Real world classification problems often involve continuous data making the development of reliable classifiers difficult. One way to handle continuous data is by partitioning the data into crisp or discrete intervals. The 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 sensitive to the subsets of the

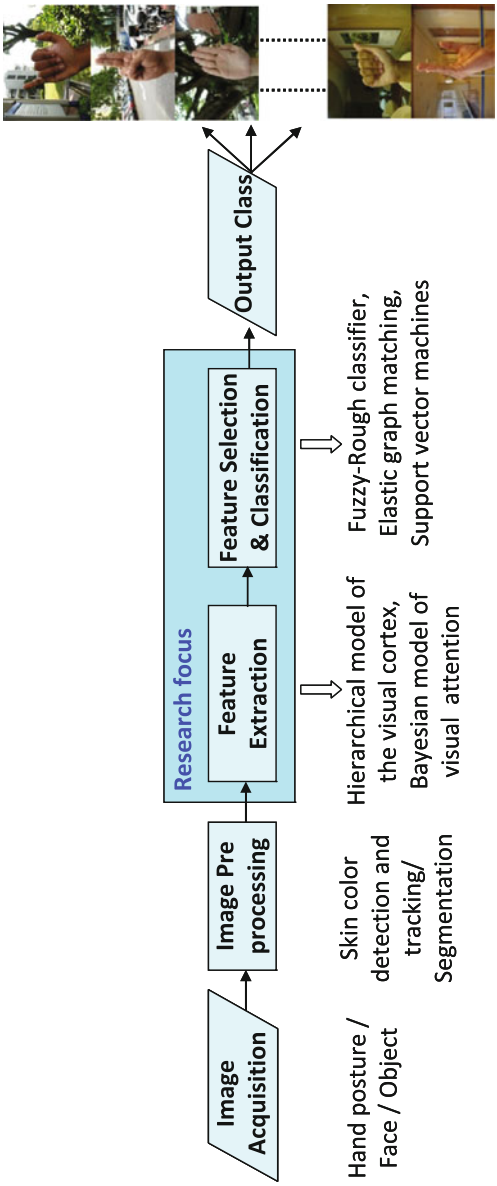


Fig. 1.1 The visual pattern recognition pipeline

information system 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 [2], offers solutions for tackling the difficulties associated with continuous data. Fuzzy discretization of the feature space alleviates the issues associated with crisp discretization.

Fuzzy and rough sets are computational intelligence tools useful for decision making in uncertain situations [2–10]. Fuzzy and rough set theories are considered complementary in that they both deal with uncertainty: vagueness for fuzzy sets and indiscernibility for rough sets [11]. These two theories can be combined to form rough-fuzzy sets or fuzzy-rough sets [11, 12]. Combining the two theories provides the concepts of lower and upper approximations of fuzzy sets by similarity relations, which are useful in classification problems.

The presence of large number of features makes the classification of patterns difficult. An abundance of features increases the size of the search space, thereby increasing the classification time. Fuzzy-rough approach is useful in selecting relevant predictive features from multi-feature datasets. The feature selection and classification algorithms presented 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 applied in cancer and tumor classification as well.

1.2.1.2 Feature Extraction

The visual pattern recognition algorithms presented utilize a biologically inspired feature extraction system. The visual cortex model has good selectivity and invariance capabilities. The extracted features provide maximum inter class discrimination among multiple classes. In addition these features have invariance with respect to appearance, perspective, and person.

The complex natural background problem in hand posture recognition is addressed using a Bayesian model of visual attention [13]. The feature based visual attention helps to detect and identify the hand region in the images. A saliency map is generated using a combination of high and low level image features. 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 and YCbCr color spaces, and the similarity to skin color map.

1.3 Hand Gestures: Variation in Appearance, Large Number of Features and Complex Backgrounds

Hand gesture is a non-verbal mode of communication in human interactions. Hand gestures are natural and widely useful in human-machine interfaces, sign languages, and virtual reality. Hand postures are person dependent as the human hand is highly

articulable, deformable and varying in size. Other factors which affect the appearance of the hand postures are the view point, scale, illumination and 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 algorithms which are 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 postures looking similar. The biologically inspired object recognition models provide a trade-off between the selectivity and invariance.

The features extracted using the visual cortex model provide good recognition accuracy. A major limitation of the model in real-world applications is its processing speed [14]. From a classification perspective, both relevant and irrelevant features are pooled together in the model. One of the ways to improve the real-time performance of the model is by extracting a set of reduced and discriminative features. In addition, the features extracted from images of different classes which are visually similar, can have vague and indiscernible classification boundaries. Discriminative feature selection and computationally simple classifiers capable of classifying vague and indiscernible data are essential to address these issues. Fuzzy-rough approach is suitable in discriminative feature selection and classification, which can improve the real-time performance.

Complex sceneries and cluttered backgrounds make the recognition of hand gestures difficult. Skin color based segmentation improves the performance to certain extent. However, the conventional skin color based algorithms fail when the complex background contains skin colored regions. Visual attention models are useful in segmentation and recognition of hand gestures against complex backgrounds.

1.4 The Algorithms: Fuzzy-Rough Classifier, Biologically Inspired Feature Extraction and Visual Attention

The research reported in this book is motivated by the 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 among similar classes, and complex backgrounds are the key issues which need better solutions.

Initial part of the book (Chaps. 4 and 5) presents two classification and feature selection algorithms for multiple feature datasets. The two algorithms basically differ in the number of identified feature cluster centers. Data sets with large number of samples can have multiple feature cluster centers. The number of identified cluster centers is dependent on the clustering radius considered. The feature selection and classification algorithm presented in Chap. 4 utilizes only the main cluster center. The features selected in an unsupervised manner lead to speedy training. Multiple cluster centers are identified and utilized in the modified algorithm in Chap. 5.

Additional cluster centers are identified by reducing the clustering radius, resulting in additional classification rules and better recognition accuracy. The multiple clusters identified can have different densities. The distance from cluster center to respective classification boundary can differ in clusters. The unsupervised feature selection algorithm is not effective in such scenarios. Supervised feature selection algorithm is useful in addressing these issues.

The concepts of fuzzy and rough sets are combined to develop computationally efficient and accurate classifier algorithms. The classification algorithms have a polynomial time complexity. Combining fuzzy and rough sets helps in classifying vague and indiscernible data. The algorithms are capable of identifying the discriminative features in the dataset. The classification rules are generated by calculating the fuzzy lower and upper approximations around the data cluster centers. A genetic algorithm based on iterative rule learning is employed in combination with a boosting technique, for generating the possible rule set from fuzzy upper approximation, incrementally by optimizing one fuzzy classifier rule at a time (Chap. 6).

The later part of the book (Chap. 7) discusses the development of hand posture recognition algorithms using the standard model of the ventral stream of visual cortex [14–16]. The hand posture recognition algorithms discussed have good discriminative power between different posture classes. The visual cortex model provides two kinds features namely C_1 and C_2. The C_1 features are invariant edge features and the C_2 features are the learned shape descriptors. The invariant edge features are useful in improving the performance of graph matching based pattern recognition algorithms. The graph algorithms capture the geometrical distribution of features. The shape descriptors have good discriminative power in classifying hand shapes and patterns.

Chapter 8 elaborates a visual attention based algorithm for the recognition of hand postures against complex backgrounds. A combination of color, texture, and shape features improve the recognition accuracy. A saliency map generated using Bayesian inference segments the hand region. The postures are classified using the shape and texture based features of the hand with a Support Vector Machines (SVM) classifier.

References

1. Z. Zhang, Microsoft kinect sensor and its effect. IEEE Multi Media **19**(2), 04–10 (2012)
2. A. Lotfi, Zadeh. Fuzzy Sets. Inf. Control **8**(3), 338–353 (1965)
3. Z. Pawlak, Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
4. Z. Pawlak, Rough classification. Int. J. Man Mach. Stud. **20**, 469–483 (1984)
5. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about data* (Kluwer Academic Publishers, Dordrecht, 1991)
6. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition, PhD Thesis, National University of Singapore (August, 2011)
7. P.K. Pisharady, P. Vadakkepat, S. Ganesan, and Ai Poh Loh, *Boosting based fuzzy-rough pattern classifier, Trends in Intelligent Robotics: Proceedings of the 15th Robot World Cup and Congress, FIRA 2010, Bangalore, India, September15-19, 2010* vol. 103 (2010), pp. 306–313

8. P.K. Pisharady, P. Vadakkepat, and A.P. Loh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft Comput. **11**(4), 3429–3440 (2011)
9. P.K. Pisharady, P. Vadakkepat, and A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **7**(3), 331–356 (2010)
10. H.J. Zimmermann, *Fuzzy Set Theory and Its Applications* (Kluwer Academic Publishers, Boston, 1991)
11. D. Dubois, H. Prade, in *Putting Rough Sets and Fuzzy Sets together, Intelligent Decision Support: Handbook of Applications and Advances in Rough Sets Theory* ed. by R. Slowinski, Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 11 (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992), pp. 203–232
12. D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets. Int. J. Gen. Syst. **17**, 191–209 (1990)
13. P.K. Pisharady, P. Vadakkepat, and A.P. Loh, Attention based detection and recognition of hand postures against complex backgrounds. Int. J. Comput. Vision **101**(3), 403–419 (2013)
14. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
15. P.K. Pisharady, Q.S.H. Stephanie, P. Vadakkepat, and A.P. Loh, *Hand posture recognition using neuro-biologically inspired features, Trends in Intelligent Robotics: 15th Robot World Cup and Congress, FIRA 2010, Bangalore, India, September 15-19, 2010, Proceedings*, vol. 103, (2010) pp. 290–297
16. P.K. Pisharady, 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

Chapter 2
Computational Intelligence Techniques

The true sign of intelligence is not knowledge but imagination

Albert Einstein

Abstract The chapter explains the computational intelligence techniques utilized in the algorithms presented in the book. The fuzzy and rough sets, fuzzy-rough sets, genetic algorithm and, feature selection and classification using the fuzzy-rough sets are detailed. The biologically inspired feature extraction system utilized in the presented algorithms is explained.

Keywords Fuzzy sets · Rough sets · Fuzzy-rough sets · Genetic algorithm · Biologically inspired features · Classification

2.1 Fuzzy and Rough Sets

In classical set theory, an element of the universe either belongs or not belongs to a set. The belongingness of the element is crisp: either yes (in the set) or no (not in the set). In fuzzy sets, belongingness can lie between yes and no; for example, a set of *tall* persons. We cannot classify a person as tall into yes / no category, as there exists no well-defined boundary for the set *tall*. Attributes used for short and tall classifications namely height is a fuzzy attribute and the sets short and tall are fuzzy sets. Each fuzzy set is characterized by a membership function $m(x)$ defining the degree of membership of the element x, usually in the range [0, 1] [1].

Fuzzy inference systems have been successfully applied in control, pattern classification, decision analysis, expert systems, manufacturing and computer vision [2]. The concept of fuzzy sets is important in pattern classification, due to its similarity with human reasoning. Traditionally, the rules in a fuzzy inference system are generated from expert knowledge. If no expert knowledge is available, the usual approach is to identify and train fuzzy membership functions in accordance with the data clusters in a training set.

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_2,
© Springer Science+Business Media Singapore 2014

The rough set theory was introduced in the early eighties as a tool to handle inconsistencies among data [3, 4]. A rough set is a formal approximation of a vague concept by a pair of precise concepts, called lower and upper approximations [5]. Rough sets handle uncertainty by computing the lower and upper approximations. Objects belonging to the same category characterized by the same attributes are indistinguishable and indiscernible.

Let $I = (U, A)$ be an information system, where U is the universe of discourse and A is a non-empty finite set of features such that $a : U \rightarrow V_a, \forall a \in A$, V_a being the set of values for feature a. In most applications, the outcome of classification is known and represented by a special *decision attribute* set. Consider a decision support system, $A = C \cup D$ where C is a set of conditional attributes and D is a set of decision attributes. With any $P \subseteq A$ there associated is an equivalence relation $IND(P)$:

$$IND(P) = (x, y) \in U^2 | \forall a \in P, a(x) = a(y). \tag{2.1}$$

If $(x, y) \in IND(P)$, then x and y are indiscernible with the features from P. The equivalence classes of P-indiscernibility relation are denoted by $[x]_P$. For any $X \subseteq U$, X can be approximated using the information contained only in P by constructing *P-lower* and *P-upper approximations* of X, denoted by $\underline{P}X$ and $\overline{P}X$ respectively,

$$\underline{P}X = x | [x]_P \subseteq X \tag{2.2}$$

$$\overline{P}X = x | [x]_P \cap X \neq \phi \tag{2.3}$$

If P and Q are equivalence relations over U, then the positive region $POS_P(Q)$ is defined as,

$$POS_P(Q) = \bigcup_{x \in U/Q} \underline{P}X. \tag{2.4}$$

For pattern classification purposes, the positive region contains all objects of U that can be classified into classes of U/Q using the knowledge in attribute set P.

Several measures of uncertainty exist in rough set theory [6]. The most widely used is the quality of the lower and upper approximations. For a given set of data samples X, not necessarily definable by a set A of attributes, the quality of lower approximation is the ratio of the number of all elements in the lower approximation of X to the total number of samples. It is interpreted as the ratio of the number of all *certainly classified* samples by attributes from A as being in X to the number of all samples. Similarly, the quality of upper approximation is the ratio of the number of all elements in the upper approximation of X to the total number of samples. It is taken as the ratio of the number of all *possibly classified* samples by attributes from A as being in X to the total number of samples.

2.1.1 Fuzzy-Rough Sets

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 different output classes. A combination of fuzzy and rough sets, namely fuzzy-rough sets [7, 8], 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 with the fuzzy set theory to form fuzzy equivalence class [8]. 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 [9].

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 containing patterns from different output classes. The definite and possible members of the output class are identified using the lower and upper approximations [3] of the fuzzy equivalence classes. The description of a fuzzy set C_c (output class) by means of the fuzzy partitions in the form of lower and upper approximations $\underline{C_c}$ and $\overline{C_c}$ is as follows [9] :

$$\begin{aligned} \mu_{\underline{C_c}}(F_j) &= \inf_x \{\max(1 - \mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall x \in X \\ \mu_{\overline{C_c}}(F_j) &= \sup_x \{\min(\mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall x \in X \end{aligned} \tag{2.5}$$

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 sample x in the fuzzy equivalence class F_j and output class C_c respectively.

2.1.2 Feature Selection and Classification Using Fuzzy-Rough Sets

The concept of fuzzy discretization of feature space for a rough set theoretic classifier is provided in [10]. The merit of fuzzy discretization over crisp discretization in terms of classification accuracy is demonstrated when overlapping datasets are used. A novel fuzzification technique namely Modified Minimization Entropy Principle Algorithm (MMEPA) and an entropy-based fuzzy-rough approach for extracting classification rules are reported in [11].

The fuzzy-rough uncertainty is exploited to improve the classification efficiency of a conventional K-nearest neighbor (K-NN) classifier in [9]. 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 [12] for hand gesture recognition. Fuzzy-rough concept removes the training samples in the class boundary and overlapping regions, improving classification accuracy. The algorithm is applied to only one type of problem, the hand gesture recognition, whereas [9] applied the algorithm to different problems; speech classification, image based letter

classification, and medical diagnosis (head injury classification). [13] presented a fuzzy-rough nearest neighbour (FRNN) classification algorithm, as an alternative to the fuzzy-rough ownership function (FRNN-O) approach reported in [9]. In contrast to [9], the algorithm proposed in [13] utilizes nearest neighbors to construct lower and upper approximations of decision classes. The algorithm classifies test instances based on memberships in the lower and upper approximations. FRNN outperformed both FRNN-O and traditional fuzzy nearest neighbour (FNN) algorithms.

A novel concept namely *consistence degree* is proposed in [14] to use as a critical value to reduce redundant attributes in database. A rule based classifier using a generalized fuzzy-rough set model is reported. The classifier is effective on noisy data. A comparison between fuzzy-rough classifier and neural network classifier is provided in [15]. The fuzzy-rough classifier is reported as a better choice with lesser training time, lesser dependence on training data, and for being transparent.

A feature selection method with fuzzy-rough approach and ant colony optimization is provided in [16]. Shen et al. [17] 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 [18] 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 feature dimensionality are reduced).

A robust feature evaluation and selection algorithm, using a different model of fuzzy-rough sets namely soft fuzzy-rough sets, is provided in [19]. This method is more effective in dealing with noisy data. [20] proposed a fuzzy-rough feature selection algorithm, with application to microarray based cancer classification. These works utilized standard classifiers (KNN, C 5.0) for classification.

2.1.3 Genetic Algorithm

Genetic Algorithms (GAs) are random search algorithms inspired by natural genetics to find optimum solutions to problems [21, 22]. The basic concept is to maintain a population of chromosomes representing candidate solutions. The population evolves over time through a process of competition and controlled variation. Each chromosome in the population has an associated fitness. Based on the fitness values, new chromosomes are generated using genetic operators such as *crossover* and *mutation*. The outline of a basic GA is given below:

1. Create an initial population;
2. Evaluate fitness of each chromosome in the population;
3. Based on fitness, select chromosomes for reproduction;
4. Apply genetic operations, *crossover* and *mutation* on selected chromosomes, to create new chromosomes;
5. Replace a part of the current population with newly generated chromosomes; and,
6. Terminate GA, if stopping criterion is satisfied, else return to step 2.

GAs have long been utilized in fuzzy inference systems for generating fuzzy rules and training membership functions [23–25]. There are there different approaches to utilize GA in learning, the Michigan, Pittsburgh and the Iterative Rule Learning (IRL) approaches. In Michigan approach, each chromosome corresponds to a classifier rule whereas in the Pittsburgh approach each chromosome encodes a complete set of classifier rules. In the IRL approach, each chromosome represents only one rule, but contrary to Michigan, only the best individual is taken as the solution while discarding the rest of the chromosomes (rules). Since a single rule provides only partial solution, GA is placed in an iterative scheme for generating a set of rules. In the iterative scheme, either the selected rules or the classified samples are penalized, by way of reduced weights, ensuring the search for new rules is focused on unclassified samples. A major issue in Michigan approach is the possible conflict between the objectives of collective rule set (the classifier) and individual rules. In Pittsburgh approach competition occurs among complete rule sets rather than among individual rules, avoiding the conflict. Pittsburgh approach is computationally more intensive due to maintenance and evaluation of complete rule sets. IRL approach is advantageous for being computationally less intensive (seeking only a single rule in each iteration), generating a cohesive classifier through a penalizing mechanism and avoiding conflict between the classifier rule set and individual rules [26].

2.2 Computational Model of Visual Cortex

Extracting features for visual pattern recognition is an ongoing research topic in computer vision. Over the past decades, there are several successful attempts in vision based pattern analysis. The mainstream computer vision has always been challenged by human vision, and the mechanism of human visual system is yet to understand, which is a challenge in both neuroscience and computer vision. The human visual system rapidly and effortlessly recognizes large number of diverse objects in cluttered natural scenes and identifies specific patterns which has 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 gaps among 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 [27]. The cortical simple cell receptive fields are modeled [28] using a Gabor function (also known as Gabor wavelet or Gabor filter) described by 2.6.

$$F(x, y) = \exp\left(-\frac{({x_0}^2 + \gamma^2 y_0^2)}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda} x_0\right), \tag{2.6}$$

$$x_0 = x \cos\theta + y \sin\theta \text{ and } y_0 = -x \sin\theta + y \cos\theta. \tag{2.7}$$

where,

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

Gabor wavelet based features have good discriminative power among different textures and shapes in images. Gabor filters resemble the receptive fields of neurons in the primary visual cortex of mammals [28]. Use of 2D Gabor wavelet representation in computer vision was pioneered by Daugman [29]. Riesenhuber and Poggio extended the approach and proposed a hierarchical model of ventral visual object-processing stream in the visual cortex [30]. Serre et al. implemented a computational model of the system and used it for robust object recognition [31, 32]. The features extracted by the model are known as the C_1 and C_2 standard model features (SMFs). The C_2 SMFs are used for hand writing recognition [33] and face recognition [34]. These features are scale and position invariant, and the feature extraction algorithm does not require image segmentation. The number of extracted features is independent of the input image size. Serre et al. used C_2 features for robust object recognition [31, 32]. The algorithms reported in this book utilize the C_2 features for multi-class recognition of hand postures and human faces [35–40].

2.2.1 Biologically Inspired Feature Extraction System

The computational model proposed by Serre et al. consists of four layers (Table 2.1). 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 sizes, and, to position and rotation variations in a 2-D plane of visual field. In layer 3 (S_2), radial basis functions (RBFs) are used to imitate the quaternary visual area (V4) and posterior inferotemporal (PIT) cortex. This aids shape recognition by comparing the complex features at the output of C_1 stage (corresponding to retinal image) with patches of previously seen visual images and shape features.[1] 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 expressing the best comparison with previously seen images. The output of layer 4 are the C_2 SMFs, which are used for image classification.

Simple cells in the third layer implement an RBF combining bars and edges in image to more complex shapes. RBFs are a major class of neural network model, comparing the distance between input and a prototype [41]. Each S_2 unit response

[1] In humans these patterns are stored in synaptic weights of neural cells.

Table 2.1 Different layers in a 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

depends on the Euclidean distance between a new input and a stored prototype in a Gaussian-like manner. The prototype patches of different sizes (centers of the RBF units) are drawn randomly (random image and position) from the training images at the second layer (C_1). Each patch contains four orientations. The third layer compares these patches by calculating the summed Euclidean distance between the patch and every possible crops (combining all orientations) from the image of similar size. This comparison is done separately with each scale-band representation in second layer.

The final set of shift and scale invariant C_2 response 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 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. 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 Chaps. 4 and 5 extract the image features using the above feature extraction system. The focus of the chapters is on feature selection and classification aspects. The algorithms presented in Chaps. 7 and 8 modify the feature extraction system for hand posture recognition application.

References

1. A. Lotfi, Zadeh. Fuzzy Sets Inf. Control **8**(3), 338–353 (1965)
2. H.J. Zimmermann, *Fuzzy Set Theory and Its Applications* (Kluwer Academic Publishers, Boston, 1991)
3. Z. Pawlak, Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
4. Z. Pawlak, Rough classification. Int. J. Man Mach. Stud. **20**, 469–483 (1984)
5. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data* (Kluwer Academic Publishers, Dordrecht, 1991)
6. Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets. Commun. ACM **38**(11), 89–94 (1995)
7. D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets. Int. J. Gen. Syst. **17**, 191–209 (1990)
8. D. Dubois, H. Prade, in *Putting Rough Sets and Fuzzy Sets Together*, Theory ed. by R. Slowinski Intelligent Decision Support: Handbook of Applications and Advances in Rough Sets , Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 11 (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992), pp. 203–232

9. M. Sarkar, Fuzzy-rough nearest neighbor algorithms in classification. Fuzzy Sets Syst. **158**, 2134–2152 (2007)
10. A. Roy and, K. P. Sankar, Fuzzy discretization of feature space for a rough set classifier. Pattern Recogn. Lett. **24**, 895–902 (2003)
11. Y.C. Tsai, C.H. Cheng, J.R. Chang, Entropy-based fuzzy rough classification approach for extracting classification rules. Expert Syst. Appl. **31**(2), 436–443 (2006)
12. X. Wang, J. Yang, X. Teng, N. Peng, Fuzzy-rough set based nearest neighbor clustering classification algorithm. Lect. Notes Comput. Sci. **3613**(2005), 370–373 (2005)
13. 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
14. S. Zhao, E.C.C. Tsang, D. Chen, X. Wang, Building a rule-based classifier-a fuzzy-rough set approach. IEEE Trans. Knowl. Data Eng. **22**(5), 624–638 (2010)
15. 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
16. R. Jensen, Q. Shen, Fuzzy-rough data reduction with ant colony optimization. Fuzzy Sets Syst. **149**(1), 5–20 (2005)
17. Qiang Shen, Alexios Chouchoulas, A rough-fuzzy approach for generating classification rules. Pattern Recogn. **35**, 2425–2438 (2002)
18. E.C.C. Tsang, S. Zhao, Decision table reduction in kdd: fuzzy rough based approach. Trans. Rough Sets Lect. Notes Comput. Sci. **5946**, 177–188 (2010)
19. H. Qinghua, A. Shuang, Y. Daren, Soft fuzzy rough sets for robust feature evaluation and selection. Inf. Sci. **180**(22), 4384–4400 (2010)
20. F.F. Xu, D.Q. Miao, and L. Wei, Fuzzy-rough attribute reduction via mutual information with an application to cancer classification. Comput. & Math. Appl. **57**(6), 1010–1017 (2009)
21. D.E. Goldberg, *Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, MA, 1989)
22. J.H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975)
23. H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans. Fuzzy Syst. **3**(2), 260–270 (1995)
24. T. Nakashima, G. Nakai, H. Ishibuchi, Improving the performance of fuzzy classifier systems by membership function learning and feature selection, in *Proceedings of IEEE International Conference on Fuzzy Systems* (Honolulu, USA, 2002), pp. 488–493
25. Y. Shi, R. Eberhart, and Y. Chen, implementation of evolutionary fuzzy systems. IEEE Trans. Fuzzy Syst. **7**(2), 109–119 (1999)
26. P.K. Pisharady, P. Vadakkepat, S. Ganesan, A.P. Loh, Boosting based fuzzy-rough pattern classifier, Trends in Intelligent Robotics, in *Proceedings of the 15th Robot World Cup and Congress, FIRA 2010*, Bangalore, 15–19 Sept 2010, vol. 103, pp. 306–313 (2010)
27. T.N. Wiesel, D.H. Hubel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**, 106–154 (1962)
28. J.P. Jones, L.A. Palmer, An evaluation of the twodimensional gabor filter model of simple receptive fields in cat striate cortex. J. Neurophysiol. **58**(6), 1233–1258 (1987)
29. 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**(7), 1160–1169 (1985)
30. M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex. Nat. Neurosci. **2**(11), 1019–1025 (1999)
31. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
32. T. Serre, L. Wolf, and T. Poggio, in *Object recognition with features inspired by visual cortex*, eds. by C. Schmid, S. Soatto, C. Tomasi Conference on Computer Vision and Pattern Recognition (San Diego, CA, 2005), pp. 994–1000

33. T. van der Zant, L. Schomaker, K. Haak, Handwritten-word spotting using biologically inspired features. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1945–1957 (2008)
34. J. Lai and W.X. Wang, in *Face recognition using cortex mechanism and svm*, eds. by C. Xiong, H. Liu, Y. Huang, Y. Xiong. 1st International Conference Intelligent Robotics and Applications (Wuhan, China, 2008), pp. 625–632
35. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition. Ph.D. thesis, National University of Singapore (August, 2011).
36. P.K. Pisharady, Q.S.H. Stephanie, P. Vadakkepat, A.P.Loh, Hand posture recognition using neuro-biologically inspired features, in *Trends in Intelligent Robotics: Proceedings of the 15th Robot World Cup and Congress, FIRA 2010, Bangalore, India*, 15-19 Sep 2010, **103**, 290–297 (2010)
37. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Graph matching based hand posture recognition using neuro-biologically inspired features, in *International Conference on Control, Automation, Robotics and Vision (ICARCV) 2010* (Singapore), 2010
38. P.K. Pisharady, P. Vadakkepat, and A.P. Loh, Attention based detection and recognition of hand postures against complex backgrounds. Int. J. Comput. Vision **101**(3), 403–419 (2013)
39. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft Comput. **11**(4), 3429–3440 (2011)
40. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **7**(3), 331–356 (2010)
41. C Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, New York, 1995)

Chapter 3
Multi-Feature Pattern Recognition

The way is long if one follows precepts, but short... if one follows patterns

Lucius Annaeus Seneca

Abstract This chapter focuses on feature selection and classification of multi-feature patterns. Micro array based cancer classification and image based face recognition are discussed. A detailed review of hand gesture recognition algorithms and techniques is included. The hand gesture recognition algorithms are surveyed by classifying them into three categories (a) hidden Markov model based methods, (b) neural network and learning based methods, and (c) the other methods. A list of available hand gesture databases is provided.

Keywords Feature selection · Classification · Multiple features · Hand gesture recognition review · Cancer classification · Face recognition

3.1 Feature Selection and Classification of Multi-Feature Patterns

The computational expense in any classification process is sensitive to the number of features used to construct a classifier. The algorithms reported in this book consider the category of decision problems which are characterized with multidimensional feature space. Searching for an optimal feature subset in a high dimensional feature space is a complex process and is an NP complete problem [3].

The available features in a multi-feature dataset can be categorized into four: (a) Predictive/relevant: features which are good in interclass discrimination, (b) Misleading: features which affect classification task negatively, (c) Irrelevant: features which provide a neutral response to classifier algorithm, and, (d) Redundant: features of a class which has other relevant features for discrimination. The presence

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_3,
© Springer Science+Business Media Singapore 2014

of misleading features reduces classification accuracy and, the presence of irrelevant and redundant features increases the computational burden. A process known as *attribute reduction* [34] removes such features reducing the classifier rule-base size by preserving relevant and predictive features. In the context of machine learning, it is known as *feature selection* [46].

3.2 Cancer Classification

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 [15]. The number of genes in gene expression data is quite large (each gene profile is a feature considered for classification) and the availability of tissue samples/records is limited [35]. With large number of features compared to the number of available samples, common statistical procedures such as global feature selection can lead to false discoveries [35]. There is need for simple and robust classification algorithms for such multiple-feature classification problems.

According to Piatetsky–Shapiro et al. [35], 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 predictive features, and, (c) Clustering: finding new biological classes or refining the existing ones. The first two types of data analysis are pattern recognition/data mining problems whereas the third requires domain knowledge in biomedical field. The algorithms reported in this book (Part II) address the first two types of data-analysis.

3.3 Face Recognition

In general, images contain abundance of information and data. Large number of extracted image features makes image pattern recognition a complex process. Face recognition involves extraction of multiple facial features from images and classification of patterns utilizing the features.

The research efforts in face processing include face detection, face recognition, face tracking, pose estimation, and expression recognition [60]. The major recognition methods applied to face images are eigenface, neural network, dynamic link architecture, hidden Markov model, geometrical feature matching, and template matching [13]. The surveys on face recognition [6] and face detection [60] detail these methods.

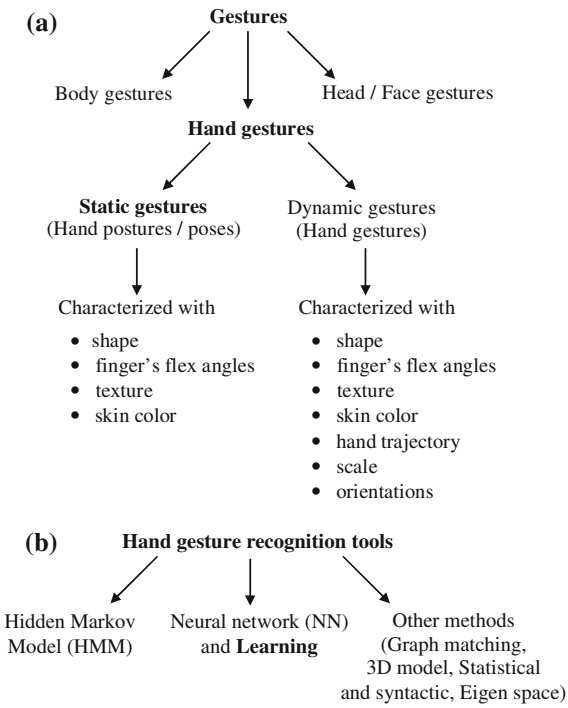


Fig. 3.1 Classification of **a** gestures and **b** hand gesture recognition tools. The algorithms reported in this book focus on static hand gestures, using a learning based approach

3.4 Hand Gesture Recognition

Hand gesture detection and recognition are the main focus of application in this book. Recognition of hand gestures is a multi-feature classification problem. This section provides a categorized analysis of different hand gesture recognition tools and a list of the available hand gesture databases. The developments in hand gesture recognition in the last decade is highlighted.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body [29]. Gestures are classified based on the moving body part (Fig. 3.1a). 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 finger flexure angles. In dynamic hand gestures (hand gestures), the hand position is temporal and it changes continuously with respect to time. Dynamic gestures rely on 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.

There exist several reviews on hand modeling, pose estimation, and gesture recognition [12, 29, 31, 33, 55, 64]. Mitra and Acharya [29] provided a survey of different gesture recognition methods, which considered hand and arm gestures, head

and face gestures, and, body gestures. Hand modeling and three dimensional (3D) motion based pose estimation methods are reviewed in [12]. An analysis of sign languages, grammatical processes in sign gestures, and issues relevant to automatic recognition of sign languages are discussed in [31]. Classification schemes in glove based and vision based sign language recognition are also discussed. Pavlovic et al. [33] surveyed the developments in gesture modeling, interpretation, and recognition techniques. Wu and Huang [55] reviewed vision based gesture recognition. An elaborate and categorized analysis of hand gesture recognition techniques is provided in this book. 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 included.

3.4.1 Hand Gesture Recognition Techniques

The early attempts in hand gesture recognition utilized mechanical devices that directly measure hand and/or arm joint angles and spatial positions, using glove-based devices. Later, vision based non-contact methods developed. Vision-based hand gesture recognition techniques are broadly divided into two categories, appearance-based and 3D hand model-based approaches. Appearance-based approaches utilize image features to model the visual appearance of the hand. Three-dimensional hand model-based approaches rely on a 3D kinematic hand model , estimating angular and linear parameters of the kinematic model.

The tools used for vision based hand gesture recognition are classified into three categories (Fig. 3.1b): (a) Hidden Markov Model (HMM) based methods [7, 20, 24, 42, 57, 62], (b) Neural network (NN) and learning based methods [4, 11, 14, 16, 26, 36–41, 44, 45, 56, 58, 59, 63], and (c) Other methods (graph algorithm based methods [38, 49–51], 3D model based methods [5, 23, 52, 61], statistical and syntactic methods [8, 53], and eigen space based methods [10, 32]).

3.4.1.1 Hidden Markov Model Based Methods

Hidden Markov Model (HMM) is the most widely used hand gesture recognition technique [7, 20, 21, 24, 27, 42, 57, 62]. It is a useful tool for modeling the spatiotemporal variability of gestures in a natural way [17]. HMM is a statistical model in which the system is modeled as a Markov process with unknown parameters. Markov process is a mathematical model of a system where the likelihood of a given future state, at any given moment, depends only on its present state, and not on any past states. HMM represents the statistical behavior of an observable sequence of process states with transition and emission probabilities. An observable event is a probabilistic function process states. The process states can be hidden and the observed data is utilized to identify the corresponding model parameters.

HMM based dynamic hand gesture recognition mainly utilize temporal and spatial features of input images. Fourier descriptor and optical flow based motion analysis are respectively useful in characterizing the spatial and temporal features. In real-world application of gesture recognition, it is required to track the hand in real-time and segment the hand shape from complex natural backgrounds. Chen et al. implemented a dynamic gesture recognition algorithm [7] for 20 different hand gestures in Taiwanese Sign Language (TSL) against complex backgrounds, with 90 % average recognition accuracy.

HMM based recognizer identifies the best likelihood gesture model for a given pattern. The variations in gesture from a reference pattern lead to reduced likelihood with the model. Lee et al. [24] introduced an HMM based *threshold model* concept to filter out patterns with less likelihood. Ten dynamic hand gestures, corresponding to 10 most frequently used browsing commands in Power Point presentation, are considered and the method extracted trained gestures from continuous hand motion with 93.14 % accuracy. Hand movement direction is used in representing spatio-temporal sequences of hand gestures. 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. Large number of states in threshold model increases computational cost and slows down recognition speed. Improvement in performance is achieved by reducing number of states. Merging of pairs of states with similar probability distributions reduces the number of states. Relative entropy (Kullback–Leibler divergence) is used as the similarity measure of two probability distributions. Kwolek [21] utilized a feature vector created from the hand movement direction to model dynamic hand gestures using HMM. The system designed is used for controlling a mobile robot.

Marcel et al. [28] proposed an extension of HMM, namely Input/Output Hidden Markov Model (IOHMM), for hand gesture recognition. IOHMM is based on a non-homogeneous Markov chain where emission and transition probabilities depend on the input. In contrast, HMM is based on homogeneous Markov chains since system dynamics is determined by time independent transition probabilities. IOHMM is a discriminative approach modeling posterior probabilities directly. The study in [28] was limited to two classes. Just et al. extended the study for the recognition of single and double hand gestures. Experiments conducted on larger databases, ranging from 7 to 16 dynamic gesture classes, concluded that HMM performs better than IOHMM for large number of classes [19, 20].

The location, angle and velocity features of the hand are combined in [62] to implement an HMM for hand gesture recognition. Hand is localized using skin-color analysis and tracked by connecting the centroid of the hand regions across frames. K-means vector quantization algorithm is applied to extracted features to create discrete symbols which are input to HMM. The k-means clustering algorithm classifies gesture tokens to different clusters in the feature space. The reported algorithm provided an accuracy of 93.25 % for set of 4,800 hand gestures which are classified to 48 classes (36 alphanumeric characters and 12 graphic elements). The variation in classification accuracy with number of features is analyzed to identify the

best features. The work concludes that angle features are most effective in providing good accuracy among the three features; location, angle and velocity. A similar HMM implementation utilizing the angle of motion of hand centroid is used to recognize 26 alphabets (A–Z) hand gestures with an average recognition rate of 90 % [27].

Gesture based human robot interaction is useful in several applications. Reliable recognition of dynamic gestures with variations in hand motion and poses is challenging especially in real world implementations. Tracking of hand contour helps in extracting the temporal characteristics of gestures. Kalman filter based estimators are suitable in hand contour tracking. Classification of static shapes with contour discriminant analysis provides symbolic descriptors corresponding to gestures which in turn are useful in training HMM. Detection of the starting and ending of gesture sequences in an automated fashion is challenging. HMM based temporal characterization combined with static shape recognition is one of the ways to recognize dynamic gestures [42]. The algorithm finds the gesturing phase by identifying the gesture starting and ending points. The technique combines shape-motion parameters and optimizes real-time gesture recognition process. The system is tested using five dynamic gestures associated with five different functions of robot motion.

Yang et al. [57] proposes an application of HMM for whole-body gesture recognition in Human-Robot Interaction (HRI). A set of features, encoding the angular relationships among a dozen body parts in 3-D is used to describe the gestures and the feature vector is used in HMM. A model reduction is carried out using relative entropy, similar to that in [24]. The whole-body gesture recognition is outside the scope of the present study.

3.4.1.2 Neural Network and Learning Based Methods

Inductive learning is a powerful approach in knowledge acquisition by inducing rules from sets of examples or sets of feature vectors. Zhao et al. [63] proposed recursive induction learning based on extended variable-valued logic in hand gesture recognition. The work modified and extended the old concept of Variable-Valued Logic into Extended Variable-valued Logic (EVL) which provides a more powerful representation capability. Rule Induction by Extended Variable-valued Logic (RIEVL) is a heuristic algorithm which can learn rules both from examples and rule sets leading to compact rules. The ability of RIEVL to abstract reduced rule sets is critical in efficient gesture recognition. The algorithm is capable of automatically determining the most effective features. RIEVL is well suited for gesture pose recognition as recursive learning allows refining gesture coding and variable-valued logic permits multi-valued feature representation of gesture poses.

A time delay neural network is used in [58, 59] to learn 2-D motion trajectories, for classifying dynamic hand gestures. 2-D motion trajectories are extracted by computing pixel matches between consecutive image pairs, after finding the corresponding affine transformations. 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. Motion information in areas with

Table 3.1 Hidden markov model based methods for hand gesture recognition

Algorithm	Features	Classification method	No. of classes	Accuracy (%)	Application
Chen et al. [7]	Fourier descriptors/ optical flow	HMM	20	above 90	Taiwanese sign language
Lee et al. [24]	Direction of hand movement	HMM	10	93.14	Browsing commands in power point presentation
Just et al. [20]	3D trajectory, hand displacement, color and shape of hand blob	HMM and IOHMM	16 and 7 (two datasets)	75 and 63, 98 and 43	Interact-play, manipulation
Yoon et al. [62]	Hand location, angle and velocity	HMM	48	93.25	HCI- recognizing alphanumeric characters and graphic elements
Ramamoorthy et al. [42]	Hand shape and hand motion	HMM	5	79	Remote robot control

HMM hidden Markov model, *IOHMM* input output hidden Markov model, *HCI* human computer interaction

little intensity variation is contained in contours of associated regions. The motion segmentation algorithm computes the best affine transformations for changes in contour shapes. The affine transformation parameters for regions at different scales are used to derive a single motion field, which is then segmented to identify differently moving regions between two frames. The 2-D motion trajectories are then learned using a time-delay neural network (TDNN). TDNN is a multilayer feed-forward network that uses shift windows among all layers representing temporal relationships. 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 into the network. TDNN has two good properties: Ability to recognize patterns from poorly aligned training examples and small number of network weights. The algorithm is tested using 40 hand gestures of American Sign Language. Best accuracy achieved for test set is 99.02 %.

A novel approach to user independent static hand gesture recognition system is proposed in [25, 26]. The system is made adaptive to the user by on-line supervised learning. Any non-trained user be able to use the system instantly. When recognition accuracy decreases, only the falsely detected gestures are retrained leading to fast adaptation. A supervised training method corrects the unrecognized gesture classes and an unsupervised method continuously runs to track the slight changes in gesture styles. These training methods embedded into the recognition phase can modify reference classed during system operation. The system is implemented as a camera-projector system where users can directly interact with the projected image using hand gestures, realizing an augmented reality tool in a multi-user environment. The novelty of the approach is in the dynamic and quick follow-up training, without the need for off-line learning using a large database.

Ng and Ranganath [30] proposed a combination of hidden Markov model (HMM) and recurrent neural networks (RNN) improving the classification accuracy that is achievable with either HMM or RNN. The features used are based on fourier descriptors and both static and dynamic gestures are considered. The system is configured to interpret the user gestures in real-time, to manipulate windows and objects within a graphical user interface. The processing is completed 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 second stage using two independent classifiers, HMM and RNN. Such a modular approach leads to robust gesture recognition. Input to the classifiers in the second stage is the hand pose likelihood vector from each frame and the hand motion vector. Outputs from the two classifiers are combined at the linear output layer.

Locally linear embedding (LLE) [43] is an unsupervised learning algorithm that attempts to map high-dimensional data to low-dimensional space while preserving the neighborhood relationships. Distributed locally linear embedding (DLLE) is proposed in [14], for static gesture recognition and dynamic gesture tracking, by extracting the intrinsic structure of data. A probabilistic neural network (PNN) is employed for static gesture classification. The inherent properties of the input data are discovered considering distributed information. The images which are similar are projected with a short distance while the images that differ greatly are projected

Table 3.2 Neural network and learning based methods for hand gesture recognition

Algorithm	Static/dynamic	Features	Classification method	No. of classes	Accuracy (%)	Application
Zhao et al. [63]	Static	Multi-Valued features	RIL	20	94.4	Gesture commands
Yang et al. [59]	Dynamic	2D motion field/trajectory	NN	40	99.02	American sign language
Licsar et al. [26]	Static	Discrete fourier transform based distance matric	Nearest neighbour/maximum likelihood	9	above 98	Gesture commands
Chan et al. [30]	Static and Dynamic	Fourier descriptors (represents shape of hand blob)	RBF, HMM and RNN	14	91.9	Manipulation of objects in windows/graphical user interface
Ge et al. [14]	Static	Geometric distance	DLLE / PNN	14	93.2	Manipulation of objects in windows/graphical user interface
Teng et al. [45]	Static	Intrinsic geometry of hand	SLLE	30	90	Chinese sign language

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

with a longer distance. The distances in low dimensional space are used to classify different static gesture images with a probabilistic neural network (PNN). A skeleton model of hand is used for recognition and tracking. PNN is based on Bayes theorem, having higher training speed, good classification accuracy, and negligible retraining time. An accuracy of 93.2 % is achieved for classification of 14 static gesture classes, which outperformed HMM and NN based methods [30].

Supervised learning in LLE algorithm is introduced in [45], for recognizing static gestures in Chinese sign language (CSL). LLE is used for feature extraction and is suitable for real-time applications. Supervised LLE (SLLE) makes use of 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 is robust against different lighting conditions and backgrounds.

3.4.1.3 Graph Algorithm Based Methods

Starting from late seventies, Graph-based techniques are used 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 research community recently. Computational cost of graph-based algorithms is now compatible with the computational power of new generation computers [9].

Elastic graph matching (EGM) is a neurally inspired pattern recognition architecture [22]. EGM has inherent ability to handle geometric distortions, does not require a perfectly segmented input image, and can elegantly represent variances in object appearance [50]. These advantages make EGM a powerful tool in gesture recognition applications. However, there are unresolved challenges like flexibility in matching process and real-time performance, which necessitates further research. A brief on graph based hand gesture recognition algorithms is provided in this section.

In a typical graph representation, regions of the image are represented by *vertices*. The vertices are connected by *edges* expressing the structural relationships among objects. Vertices and edges are usually attributed. Triesch et al. [47, 49–51] employed elastic graph matching (EGM) technique to develop person independent hand posture recognition 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 descriptor (image feature) based on Gabor filters. The algorithm 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 graph* (in database) and *data graph* (corresponding to real-time image). The nodes are compared using a similarity function, and data graph pattern is recognized by calculating average node similarities.

Bunch graphs [54] are used to model variability in object appearance. Natural variability in 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. The bunch graph method is useful in modeling complex backgrounds [51]. For the matching process, each attribute value in the bunch is compared with the local image information in data graph to generate a similarity vector. Maximum of the similarities is considered as the similarity of the bunch graph. The algorithm uses multiple features for gesture recognition. The graphs are matched using three features, (a) conventional Gabor jets, (b) color average, and (c) color Gabor jets (convolutions of Gabor filter with images expressing each pixel's similarity to skin color). Nodes are compared using node features. The weighted average of node similarities forms the net node similarity. The algorithm is tested using 12 class hand postures from 19 subjects, against simple and complex backgrounds. Accuracies of 92.9 and 85.8 % are achieved for simple and complex background images, respectively.

3.4.1.4 Topology/3D Model Based Methods

Three dimensional model fitting is used in [52] for hand pose estimation. The method estimates joint angles and hand pose is reconstructed as a voxel model in 3-D. The voxel model is fit with the hand model for recognition. Geometric information of the hand and voxel models are utilized for model fitting, without using any heuristic or priori information. Feasibility of the method is tested using simulated and real hand poses. The recognition speed of the algorithm requires improvement for real-time applications. A computer vision model is introduced in [61] which avoids the complexity in estimation of the angular and linear parameters of the kinematic model. Yin and Xie [61] suggests topological features of the hand for 3-D hand gesture recognition. The hand is segmented from complex background using a restricted coulomb energy (RCE) neural network based on color segmentation. The finger edge points are extracted and matching is done with the topological features. A novel method was suggested to estimate the epipolar geometry (fundamental matrix) between two uncalibrated cameras from stereo hand images. The fundamental matrix contains all geometric information necessary for establishing correspondence between two perspectives, from which the 3-D structure of an object is inferred. The algorithm is tested with real calibrated and uncalibrated images.

3.4.1.5 Statistical and Syntactic Analysis

Chen et al. [8] proposed a two stage approach with statistical and syntactic analyses respectively for recognizing static and dynamic hand gestures. The first stage (statistical analysis) is based on Haar-like features and AdaBoost learning algorithm. The second stage (syntactic analysis) is based on stochastic context-free grammar (SCFG). The Haar-like features effectively describe the hand posture pattern. The AdaBoost algorithm speeds up performance and constructs a strong classifier by combining a sequence of weak classifiers. The posture detected at the first stage are

converted into a sequence of terminal strings according to the grammar. The SCFG algorithm analyzes the syntactic structure based on detected postures. The gesture corresponding to an input string is recognized using the probabilities associated with production rules.

3.4.1.6 Eigen Space Method

A hand gesture recognition method without any explicit feature detection step is proposed in [32]. The method suggests a novel eigenspace based framework to model dynamic hand gestures with shape and trajectory information. Any feature-based method involves a separate time-consuming and noise-prone feature extraction step which is avoided in [32]. The approach is useful in representing gestures that cannot be recognized by shape or trajectory information alone. The algorithm is immune to common hand shape deformations: rotation, translation, scale and shear. The authors modeled an upper bound on success rate of a particular set of gestures by maximizing the distance between gesture classes in eigenspace. The algorithm is tested using an eight class gesture set and provided full classification of all test samples.

3.4.2 Hand Gesture Databases

The number of hand gesture databases available to research community is limited. Table 3.4 lists the available hand gesture databases and sources. Two hand posture databases namely NUS hand posture dataset-I (simple background) and NUS hand posture dataset-II (complex background) are contributed by the authors [39, 41].

3.4.3 Comparison of Methods

HMM based methods are effective and widely used for hand gesture recognition. HMM approaches require elaborate training to be effective. TDNN is compact in structure, economical in weights and capable in detecting general features. Hierarchy of delays in TDNN optimizes feature detectors in each layer. Temporal integration of feature detectors at output layer makes TDNN shift invariant (insensitive to hand position).

Graph based algorithms have high computational complexity making them unsuitable for real-time applications. However, these methods are effective in recognizing hand gestures against complex backgrounds [50].

Appearance-based approaches have better real-time performance as extraction of 2-D image features is fast [8]. Appearance-based methods are *purposive* and computationally efficient capable of working well under constrained situations. However, these algorithms lack generalization capabilities. Three-dimensional hand models

Table 3.3 Other methods for hand gesture recognition

Algorithm	Static/Dynamic	Features	Classification method	No. of classes	Accuracy (%)	Application
Triesch et al. [50]	Static	Gabor jets	EGM	12	92.9 (SB) and 85.8 (CB)	HRI
Ueda et al. [52]	Static	Joint angles	3D model fitting	4	NR	NR
Chen et al. [8]	Static and dynamic	Haar-Like features	Adaboost/syntactic analysis based on SCFG	4	NR	NR
Patwardhan et al. [32]	Dynamic	Hand shape/trajectory (but no explicit feature detection step)	Predictive eigen tracker	8	100	Audio player control

EGM elastic graph matching, *SB* simple background, *CB* complex background, *HRI* human robot interaction, *SCFG* stochastic context-free grammar, *NR* not reported

Table 3.4 Hand gesture databases

Source	Description	Work
[18]	Two dynamic hand gesture datasets with 16 and 7 classes	[20]
Available on e-mail request to yoonhs@etri.re.kr	48 class dynamic hand gestures representing alphanumeric characters and graphic elements	[62]
[48]	Two static hand gesture datasets with 10 (gray scale) and 12 (color) classes. Single dynamic hand gesture dataset with 4 classes	[49–51]
[1]	10 class simple background hand posture dataset	[41]
[2]	10 class complex background hand posture dataset	[39]

are rich feature descriptors capable of recognizing a wide class of hand gestures. 3-D models capture characteristic shapes in different perspectives making the algorithm invariant to size and shape. However, matching test images with every model in the database is time consuming and computationally expensive which limits the usage of 3-D models in real-time applications.

References

1. Nus hand posture dataset-i (2010), http://www.vadakkepat.com/NUS-HandSet/
2. Nus hand posture dataset-ii (2011), http://www.vadakkepat.com/NUS-HandSet/
3. A.A. Albrecht, Stochastic local search for the feature set problem, with applications to microarray data. Appl. Math. Comput. **183**, 1148–1164 (2006)
4. J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(9), 1685–1699 (2009)
5. V. Athitsos, S. Sclaroff, Estimating 3d hand pose from a cluttered image. IEEE Conf. Comput. Vis. Pattern Recognit. **2**, 9–432 (2003)
6. R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces-a survey. Proc. IEEE **83**(5), 705–740 (1995)
7. F.S. Chen, C.M. Fu, C.L. Huang, Hand gesture recognition using a real-time tracking method and hidden markov models. Image Vis. Comput. **21**, 745–758 (2003)
8. Q. Chen, N.D. Georganas, E.M. Petriu, Hand gesture recognition using haar-like features and a stochastic context-free grammar. IEEE Trans. Instrum. Meas. **57**(8), 1562–1571 (2008)
9. D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition. Int. J. Pattern Recognit Artif Intell. **18**(3), 265–298 (2004)
10. K. Daniel, M. John, M. Charles, A person independent system for recognition of hand postures used in sign language. Pattern Recogn. Lett. **31**, 1359–1368 (2010)
11. O. Eng-Jon, R. Bowden, A Boosted Classifier Tree for Hand Shape Detection, in *IEEE Conference on Automatic Face and Gesture Recognition*, pp. 889–894 (2004)
12. A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, X. Twombly, Vision-based hand pose estimation: a review. Comput. Vis. Image Underst. **108**, 52–73 (2007)
13. Y.S. Gao, M.K.H. Leung, Face recognition using line edge map. IEEE Trans. Pattern Anal. Mach. Intell. **24**(6), 764–779 (2002)
14. S.S. Ge, Y. Yang, T.H. Lee, Hand gesture recognition and tracking based on distributed locally linear embedding. Image Vis. Comput. **26**, 1607–1620 (2008)

15. 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, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by geneexpression monitoring. Science **286**, 531–537 (1999)
16. M. Hasanuzzamana, T. Zhanga, V. Ampornaramveth, H. Gotoda, Y. Shirai, H. Ueno, Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform. Rob. Auton. Syst. **55**(8), 643–657 (2007)
17. X.D. Huang, Y. Ariki, M.A. Jack, *Hidden Markov Models for Speech Recognition* (Edinburgh University Press, Edinburgh, 1990)
18. A. Just, S. Marcel, Interactplay dataset, two-handed datasets (2004), http://www.idiap.ch/resources.php
19. A. Just, S. Marcel, Sébastien marcel-interactplay database (2004), http://www.idiap.ch/resource/interactplay/
20. A. Just, S. Marcel, A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. Comput. Vis. Image Underst. **113**(4), 532–543 (2009)
21. B. Kwolek, in *The Usage of Hidden Markov Models in a Vision System of a Mobile Robot*, ed. by K. Kozlowski, M. Galicki, and K. Tchon. 2nd International Workshop on Robot Motion and Control (Bukowy Dworek, Poland, 2001), pp. 257–262
22. M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R.P. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture. IEEE Trans. Comput. **42**(3), 300–311 (1993)
23. J. Lee, T. Kunii, Model-based analysis of hand posture. IEEE Comput. Graph. Appl. **15**(5), 77–86 (1995)
24. K.H. Lee, J.H. Kim, An hmm based threshold model approach for gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **21**(10), 961–973 (1999)
25. A. Licsar, T. Sziranyi, in *Dynamic Training of Hand Gesture Recognition System*, ed. by J. Kittler, M. Petrou, M. Nixonin. 17th International Conference on Pattern Recognition (ICPR) (Cambridge, England, 2004), pp. 971–974
26. A. Licsar, T. Sziranyi, User-adaptive hand gesture recognition system with interactive training, Image Vis. Comput. **23**, 1102–1114 (2005)
27. N. Liu, B.C. Lovell, P.J. Kootsookos, in *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
28. S. Marcel, O. Bernier, J.E. Viallet, D. Collobert, Hand gesture recognition using input/output hidden markov models, in *Proceedings of the Conference on Automatic Face and Gesture Recognition*, (2000), pp. 456–461
29. S. Mitra, T. Acharya, Gesture recognition: a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **37**(3), 311–324 (2007)
30. C.W. Ng, S. Ranganath, Real-time gesture recognition system and application. Image Vis. Comput. **20**, 993–1007 (2002)
31. S.C.W. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 873–891 (2005)
32. K.S. Patwardhan, S.D. Roy, Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker. Pattern Recogn. Lett. **28**, 329–334 (2007)
33. V.I. Pavlovic, R. Sharma, T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 677–694 (1997)
34. Z. Pawlak, Rough sets and fuzzy sets, in *Proceedings of ACM*, Computer Science Conference (Nashville, Tennessee, 1995), pp. 262–264
35. G. Piatetsky-Shapiro, P. Tamayo, Microarray data mining: facing the challenges. SIGKDD Explor. **5**(2), 1–5 (2003)
36. P.K. Pisharady, Computational Intelligence Techniques in Visual Pattern Recognition, Ph.D. thesis, National University of Singapore, 2011
37. P.K. Pisharady, Q.S.H. Stephanie, P.Vadakkepat, A.P. Loh, Hand posture recognition using neuro-biologically inspired features, in *Proceedings of the Trends in Intelligent Robotics: 15th*

Robot World Cup and Congress, FIRA 2010, Bangalore, India, vol. 103, pp. 290–297, 15–19 Sept 2010
38. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Graph matching based hand posture recognition using neuro-biologically inspired features, in *Proceedings of the International Conference on Control, Automation, Robotics and Vision (ICARCV)* (Singapore, 2010)
39. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Attention based detection and recognition of hand postures against complex backgrounds. Int. J. Comput. Vision **101**(3), 403–419 (2013)
40. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft. Comput. **11**(4), 3429–3440 (2011)
41. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **07**(3), 331–356 (2010)
42. A. Ramamoorthy, N. Vaswani, S. Chaudhury, S. Banerjee, Recognition of dynamic hand gestures. Pattern Recognit **36**, 2069–2081 (2003)
43. S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
44. M.C. Su, A fuzzy rule-based approach to spatio-temporal hand gesture recognition. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **30**(2), 276–281 (2000)
45. X. Teng, B. Wu, W. Yu, C. Liu, A hand gesture recognition system based on local linear embedding. J. Vis. Lang. Comput. **16**, 442–454 (2005)
46. D. Tian, J. Keane, X. Zeng, Evaluating the effect of rough sets feature selection on the performance of decision trees. Granular Computing. IEEE Int. Conf. **2006**, 57–62 (2006)
47. J. Triesch, C. Eckes, in *Proceedings of the ICANN'98: Object Recognition with Multiple Feature Types*, 8th International Conference on Artificial Neural Networks (Skovde, Swedan, 1998)
48. J. Triesch, C. Malsburg, Sebastien marcel hand posture and gesture datasets : Jochen triesch static hand posture database (1996), http://www.idiap.ch/resource/gestures/
49. J. Triesch, C. Malsburg, in *Proceedings of the A gesture Interface for Human-Robot-Interaction*, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, (Nara, Japan, 1998), pp. 546–551
50. J. Triesch, C. Malsburg, A system for person-independent hand posture recognition against complex backgrounds. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1449–1453 (2001)
51. J. Triesch, C. Malsburg, in *Proceedings of the Robust Classification of Hand Postures against Complex Backgrounds*, 2nd International Conference on Automatic Face and Gesture Recognition, (Killington, VT, USA, 1996), pp. 170–175
52. E. Ueda, Y. Matsumoto, M. Imai, T. Ogasawara, A hand-pose estimation for vision-based human interfaces. IEEE Trans. Industr. Electron. **50**(4), 676–684 (2003)
53. W.H.A. Wang, C.L. Tung, in *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
54. L. Wiskott, J.M. Fellous, N. Kruger, C. Malsburg, Face recognition by elastic bunch graph matching. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 775–779 (1997)
55. Y. Wu, T. S. Huang, *Vision-Based Gesture Recognition: A Review*, ed. by A. Braffort, R. Gherbi, S. Gibet, J. Richardson, D. Teil. International Gesture Workshop on Gesture-Based Communication in Human Computer Interaction (Gif Sur Yvette, France), (Springer, Berlin, 1999), pp. 103–115
56. Y. Wu, T.S. Huang, View-independent recognition of hand postures. IEEE Conf. Comput. Vis. Pattern Recognit. **2**, 88–94 (2000)
57. H.D. Yang, A.Y. Park, S.W. Lee, Gesture spotting and recognition for humanrobot interaction. IEEE Trans. Rob. **23**(2), 256–270 (2007)
58. M.H. Yang, N. Ahuja, in *Proceedings of the Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (Santa Barbara, CA, USA, 1998), pp. 892–897
59. M.H. Yang, N. Ahuja, M. Tabb, Extraction of 2d motion trajectories and its application to hand gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1061–1074 (2002)

60. M.H. Yang, D.J. Kriegman, N. Ahuja, Detecting faces in images: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 34–58 (2002)
61. X. Yin, M. Xie, Estimation of the fundamental matrix from uncalibrated stereo hand images for 3d hand gesture recognition. Pattern Recognit. **36**, 567–584 (2003)
62. H.S. Yoon, J. Soh, Y.J. Bae, H.S. Yang, Hand gesture recognition using combined features of location, angle and velocity. Pattern Recognit. **34**, 1491–1501 (2001)
63. M. Zhao, F.K.H. Quek, X. Wu, Rievl: recursive induction learning in hand gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1174–1185 (1998)
64. H. Zhou, T.S. Huang, in *Proceedings of the Tracking Articulated Hand Motion with Eigen Dynamics Analysis*. International Conference on Computer Vision, vol. 2, (2003), pp. 1102–1109

Part II
Feature Selection and Classification

Chapter 4
Fuzzy-Rough Discriminative Feature Selection and Classification

Research is what I'm doing when I don't know what I'm doing

Wernher von Braun

Abstract Classification of datasets with multiple features is computationally intensive. Fuzzy-rough set based feature selection and classification requires reduced computational efforts. Lower and upper approximations of fuzzy equivalence classes are useful in finding discriminative features and classification boundaries in a dataset. This chapter discusses a *fuzzy-rough single cluster* (FRSC) classifier which is a discriminative feature selection and classification algorithm. The FRSC classifier translates each quantitative value of a feature into fuzzy sets of linguistic terms using membership functions and, identifies discriminative features. The membership functions are formed by partitioning the feature space into fuzzy equivalence classes, using feature cluster centers identified through subtractive clustering. Classification rules are generated using fuzzy membership values partitioning the lower and upper approximations. The patterns are classified through a voting process. Both the feature selection and classification algorithms have polynomial time complexity. The algorithm is tested in two types of classification problems, namely, cancer and image-pattern classification. The large number of gene expression profiles and relatively small number of available samples make the feature selection a key step in microarray based cancer classification. The algorithm identified relevant features (predictive genes in the case of cancer data) and provided good classification accuracy, at a less computational cost, with good *margin of classification*. A comparison of the performance of the FRSC classifier with other relevant classification methods shows the classifier's better discriminative power.

Keywords Fuzzy-rough sets · Feature selection · Margin classifier · Cancer classification · Fuzzy equivalence class · Discriminative features · FRSC algorithm

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_4,
© Springer Science+Business Media Singapore 2014

4.1 Introduction

The fuzzy-rough classifiers surveyed in Chap. 2 (Sect. 2.1.2) are mostly based on predefined fuzzy membership functions and are focused either on classification or feature selection. The fuzzy-rough approach is adapted as a preprocessing step. Direct construction of classifiers as an application of fuzzy-rough sets has been less studied [26]. The fuzzy-rough single cluster (FRSC) classifier discussed in this chapter is computationally efficient in feature selection and classification, especially for datasets with multidimensional feature space. The FRSC classifier automatically generates fuzzy membership functions by identifying the main cluster for each feature in a dataset. The lower and upper approximations of the fuzzy-rough sets are utilized in selecting the discriminative features. Generating the classification rules from a selected subset of features improves the computational efficiency of the classifier.

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 [13, 14] is useful for decision making in situations where indiscernibility is present, and, fuzzy set theory [25] is suitable when vague decision boundaries exist. The feature space is partitioned into fuzzy equivalence classes through fuzzy-discretization in the FRSC algorithm. The predictive features in a dataset are identified and if-then classification rules are generated using these features. The decision is made through a voting process. The FRSC feature selection and classification algorithm has a polynomial time complexity. The algorithm selects relevant features, and avoids the misleading, irrelevant and redundant ones.

The FRSC algorithm is applied to five cancer/tumor datasets. The algorithm identified predictive genes and provided good classification accuracy for all datasets considered. The 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 the visual cortex model [19].

4.2 Fuzzy-Rough Single Cluster Feature Selection and Classification Algorithm

The FRSC feature selection and classification algorithm is discussed in this section. The FRSC algorithm identifies discriminative features in a dataset and classifies data with less computational expense [15, 16].

Figure 4.1 shows the overview of the FRSC algorithm development process. In the training phase, the discriminative features in the data are selected and the classification rules are generated using a fuzzy-rough approach.

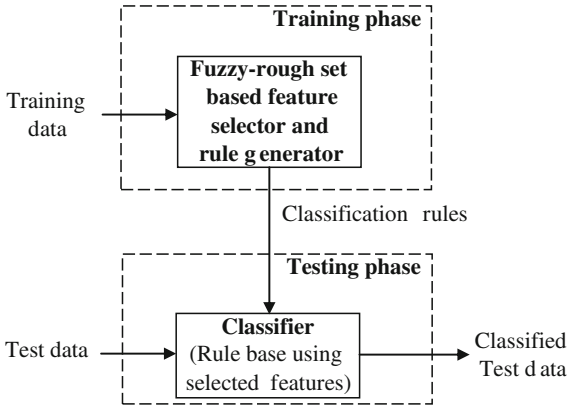


Fig. 4.1 Overview of the FRSC classifier algorithm development

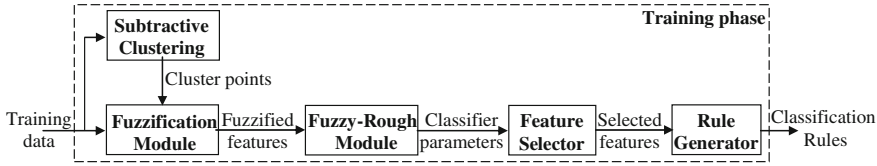


Fig. 4.2 Training phase of the FRSC classifier

4.2.1 Training Phase: Discriminative Feature Selection and Classifier Rules Generation

The training phase (Fig. 4.2) involves fuzzy discretization of feature space and, formation of fuzzy membership functions using cluster centers identified by the subtractive clustering technique [2]. The discriminative features in dataset are selected and classification rules are generated using fuzzy lower and upper approximations of the fuzzified training data.

4.2.1.1 Fuzzy Equivalence Classes, and Lower and Upper Approximations

Fuzzy membership functions are formed using feature cluster centers identified by the subtractive clustering technique. 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. Subtractive clustering algorithm selects data point with highest potential as the first cluster center and then removes all 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. Second data

[1] The subtractive clustering radius represents the width of the data considered in each step of clustering. A radius within [0.2 0.5] leads to a diameter within [0.4 1.0] and it covers 40–100 %

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 FRSC 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. 4.3a 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.

Fuzzy membership function in the example 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. The same fuzzy equivalence class contains samples from different output classes which leads to fuzzy-rough uncertainty. The presented algorithm identifies the membership values μ_{A_L} and μ_{A_H} (4.3) that partition the definite and possible members of the output class (class 2 in the example, Fig. 4.3a) and identifies the relevant features for discriminating the output class.

Samples 1–8 in Fig. 4.3a are from different partitions of the feature space. The lower and upper approximations of samples from class 2 (set X) are shown in Fig. 4.3b. The lower approximation consists of definite members and the upper approximation consists of definite and possible members of class 2.

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

The distribution of data samples in feature space may have varying overlap among different classes along different feature axes. The FRSC algorithm identifies discriminative features, features with minimum interclass overlap, as predictive/relevant features in the dataset, and generates classification rules using them. The datasets considered here have large number of features. The presence of large number of features increases the possibility of identifying predictive features with less interclass overlap.

4.2.1.2 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. 4.3a). μ_{A_L} and μ_{A_H} are calculated as follows:

(Footnote 1 continued)
of data width. This is the usual range reported in literature [2]. The number of rules and accuracy decrease with the radius.

[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 $\{A_L, A_H\}$.

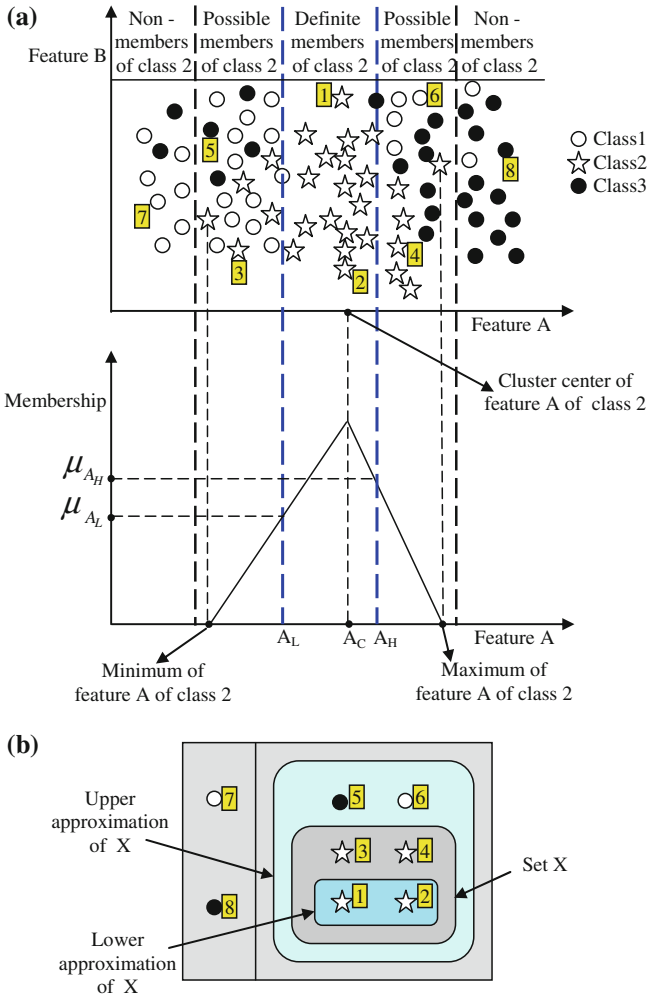


Fig. 4.3 **a** Feature partitioning and formation of membership function from cluster center points for a 3 class dataset. The output class considered is class 2. **b** *Lower* and *upper* approximations of set X

For a particular feature A, in class C_i, the maximum membership value is,

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

where,

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

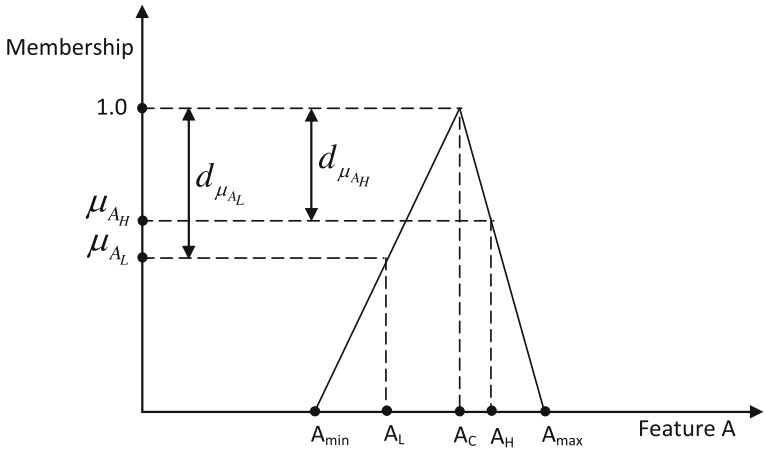


Fig. 4.4 Calculation of d_μ

and,

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

then,

$$\begin{aligned} \mu_{A_L} &= \max_{C_i \neq C^{max},\ A_{min} \leq A \leq A_C} \{\mu^{max}(C_i)\}, \\ \mu_{A_H} &= \max_{C_i \neq C^{max},\ A_C \leq A \leq A_{max}} \{\mu^{max}(C_i)\}. \end{aligned} \tag{4.3}$$

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 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. 4.4).

$$d_\mu = (d_{\mu_{A_L}} + d_{\mu_{A_H}})/2, \tag{4.4}$$

where,

$$d_{\mu_{A_L}} = 1 - \mu_{A_L} \text{ and } d_{\mu_{A_H}} = 1 - \mu_{A_H} \tag{4.5}$$

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 the 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 algorithm is tested

[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 $d_{\mu_{A_H}}$.

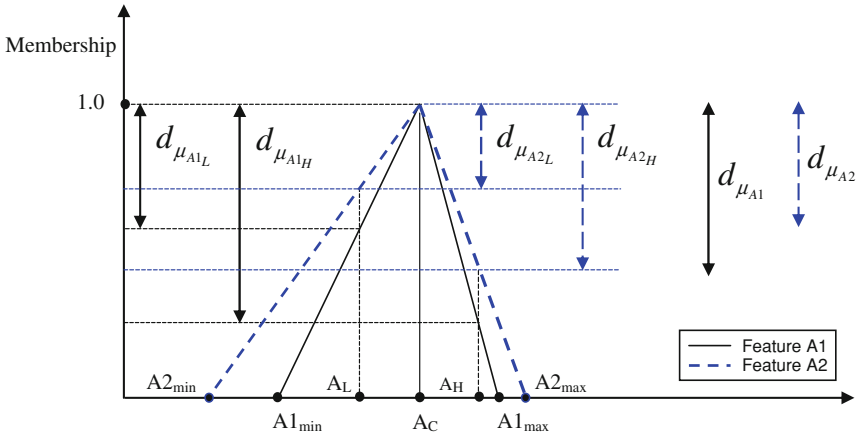


Fig. 4.5 d_μ for two features A1 and A2 with different feature ranges

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.

The capability of d_μ to identify discriminative features is explained as follows. The calculation of d_μ for two features A1 and A2 is depicted in Fig. 4.5. 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 FRSC algorithm provides preference to the denser feature range $[A_C, A_H]$ and assigns a high value to the corresponding d_μ ($d_{\mu_{A1_H}}$), compared to the d_μ of the sparse feature range $[A_L, A_C]$ ($d_{\mu_{A1_L}}$).

4.2.1.3 Classification

The feature values A_L and A_H that partition the lower and upper approximations of fuzzy equivalence class entails the rule: if the value of a feature A is within (A_L, A_H) then the sample belongs to class C^{max} (Fig. 4.4, 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 classesare

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 & *Rule* 2).

Let $\{A_{L_{ij}}, A_{H_{ij}}\}$ be the set of feature values obtained as per (4.3) where,

$i = 1, \ldots, p$; p—the number of classes,
$j = 1, \ldots, q'$; q'—the 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:

$$IF\ [A_{L_{ij}} < A_j < A_{H_{ij}}]\ THEN\ [N_{C_i} = N_{C_i} + 1] \tag{4.6}$$

where,

A_j is the jth feature of the sample to be classified,
N_{C_i} is the number of votes for a particular class C_i.

Rule 2:

$$C = argmax_{C_i}\{N_{C_i}\} \tag{4.7}$$

where,

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

A detailed flowchart of the training phase of the classifier is provided in Fig. 4.6.

The FRSC 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 classifier is defined as,

$$MC = N_p - N_{nMAX} \tag{4.8}$$

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

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

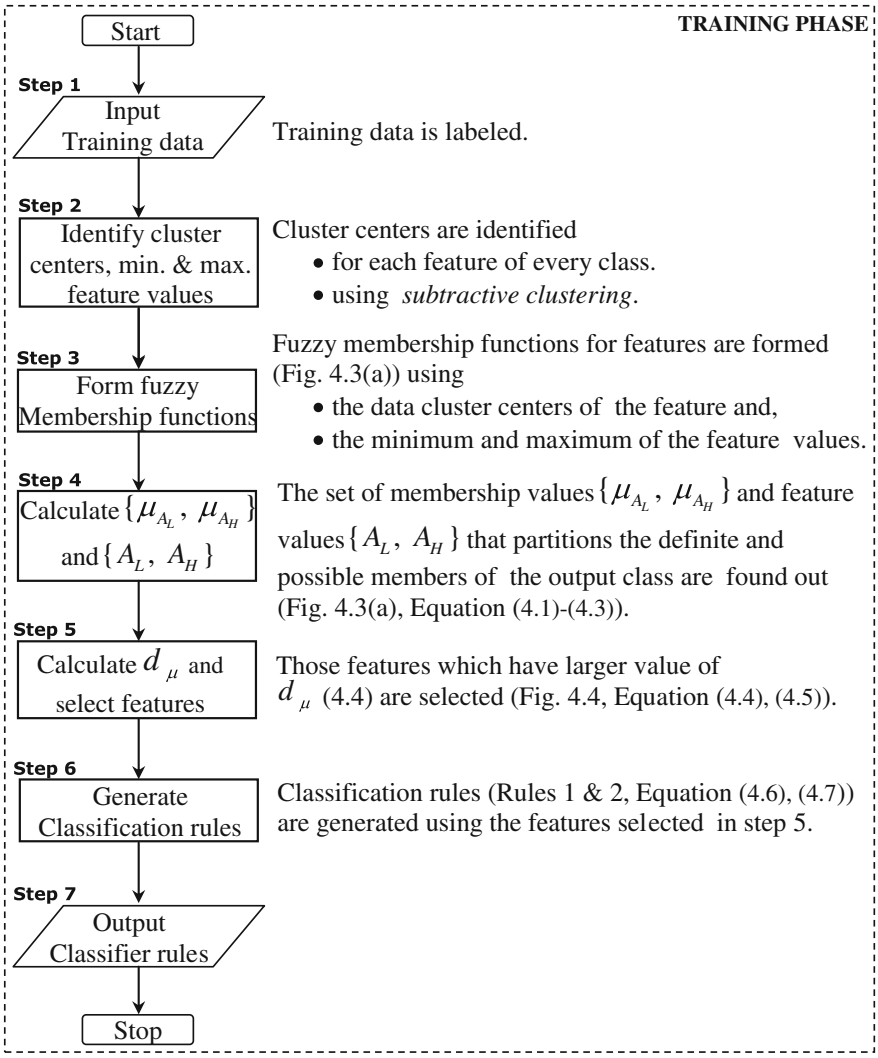


Fig. 4.6 Flowchart of the training phase

misclassification. The average margin of classification for a dataset indicates the discriminative power of the classifier. The experimental results (Sect. 4.3) evidence good discriminative power of the FRSC classifier.

4.2.2 The Testing Phase: The Classifier

The FRSC classifier is formed using the classification rules generated in the training phase. Figure 4.7 shows the flowchart of the testing phase (the classifier). The selected

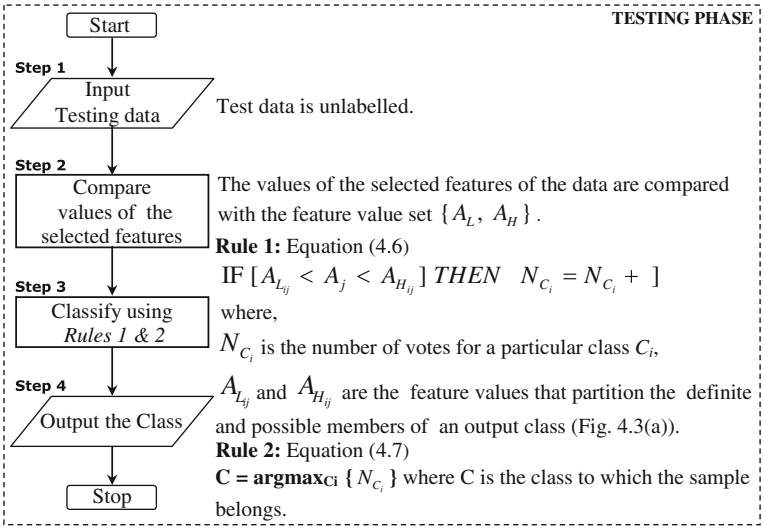


Fig. 4.7 Flowchart of the testing phase

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 Sect. 4.3.

4.2.3 Computational Complexity Analysis

This section provides the computational complexity analysis of the training and testing algorithms. Both the training and testing algorithms have polynomial time complexity.

4.2.3.1 Computational Complexity of the Classifier Training Algorithm

Figure 4.8 shows the pseudo code of the classifier training algorithm. The different parameters at the input of the training algorithm are the number of classes, 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 values of features, $O(p^2qr)$ for calculating the fuzzy memberships,

```
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 (4.1)-(4.3)
    calculate {A_L , A_H }; % finding features values that partitions definite
                            % and possible members of an output class
    calculate {d_μ}; % Equation (4.4), (4.5)
  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
```

Fig. 4.8 Pseudo code of the classifier training algorithm

$O(p^2qr)$ 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 in polynomial time.

4.2.3.2 Computational Complexity of the Classifier

The pseudo code of the classifier is shown in Fig. 4.9. 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 receives maximum votes. The overall complexity of the classifier algorithm is in polynomial time, $O(pq')$.

```
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 (4.6)
    end
  end
end
class = index {max (vote)} ; % Rule 2, Equation (4.7), find the class index
                             % which received maximum votes
print {class} ;
```

Fig. 4.9 Pseudo code of the classifier

4.3 Performance Evaluation and Discussion

The proposed FRSC algorithm provides good classification accuracy for datasets, with well separated feature cluster centers. The algorithm identifies features with well separated cluster centers as predictive features.

The FRSC algorithm is tested using 5 cancer (Table 4.1) and 3 image datasets (Table 4.2). 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 the number of features used per class and the number of classes), and the average margin of classification (MC) (Tables 4.3, 4.4, 4.5). The variation in classification accuracy with respect to number of selected features is reported (Fig. 4.10). A comparison of the classification accuracy of the reported algorithm with relevant classification methods is also provided.

4.3.1 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,

Table 4.1 Details of cancer datasets

Dataset	# Classes	# Samples
Leukemia [5]	3	72
Tumor versus normal samples [5]	2	75
Lung cancer [7]	2	181
Small round blue cell tumor [10]	4	83
Central nervous system embryonal tumor [5]	5	42

Table 4.2 Details of hand posture, face and object datasets

Dataset	# Classes	# Samples
Jochen triesch hand posture dataset [22]	10	240
A subset of yale face dataset [4]	10	640
Caltech object database [3]	4	3479

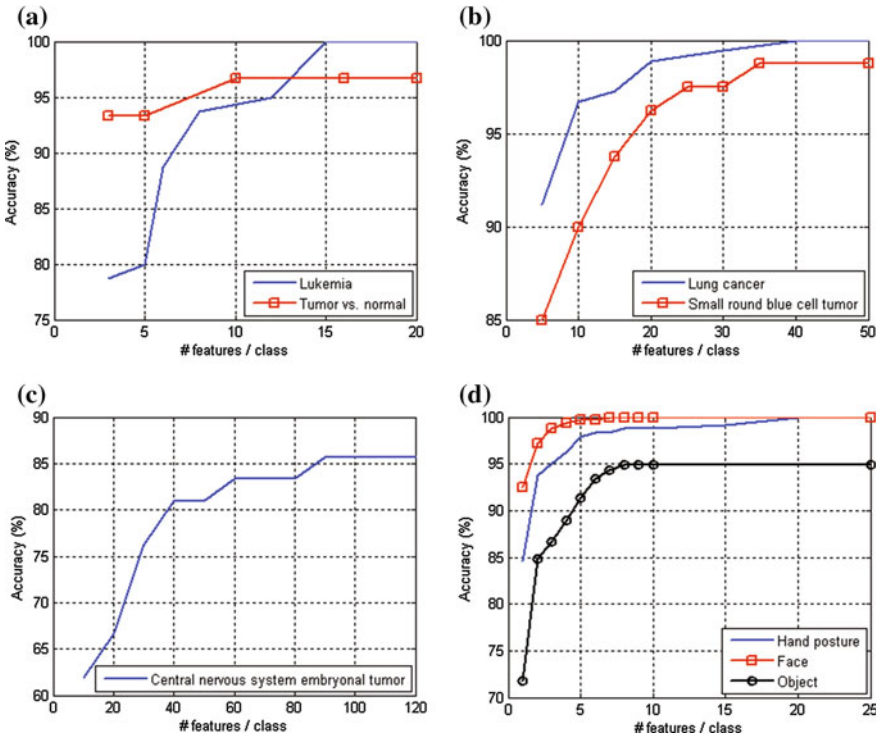


Fig. 4.10 Variation in classification accuracy with the number of selected features. **a** Lukemia and tumor versus normal classification, **b** Lung cancer and small round *blu* cell tumor classification, **c** Central nervous system embryonal tumor classification, **d** Hand posture, face and object classification

Table 4.3 Summary and comparison of cross validation test results—Cancer datasets (Training and testing are done by cross validation)

Dataset	Accuracy (%) presented method	# Features/class	Total # features	Average MC	Accuracy (%) SVM
Leukemia	100	15	35	7.03	94.44
Tumor versus 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 4.4 Comparison of classification accuracy (%) with reported results in literature—Cancer datasets (Training and testing are carried out using the sample divisions in the compared works)

Dataset	Presented method	Benchmark
Leukemia	100	98.24 [9]
Tumor versus normal samples data	100	100 [12]
Lung cancer	100	99 [8]
Small round blue cell tumor	100	100 [11]
Central nervous system embryonal tumor	85.71	83.33 [18]

10 fold cross-validation is done for all cancer datasets, except for the central nervous system embryonal tumor dataset (5th dataset), for which *leave one out cross validation* is done. The classification results are compared with that of Support Vector Machines (SVM) implemented using LIBSVM [1] (Table 4.3). The classification results are also compared with those reported in the literature (for this comparison, the training and testing of the algorithm are carried out using the same sample divisions as that in the compared work) (Table 4.4).

4.3.1.1 Leukemia Classification

The leukemia dataset [5] 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) [6]. Jirapech-Umpai et al. [9] 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). The three class classification is carried out and the 55 top ranked genes[5] by the RankGene method [9] are utilized for classification. For the 10 fold cross validation, 4 samples of ALL B-cell, 1 sample of ALL T-cell and

[5] The list of 55 top ranked genes is available in [9].

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 4.3. The FRSC algorithm provided maximum classification accuracy of 100 %, when 35 features are used. The variation in classification accuracy with respect to the number of selected features per class is shown in Fig. 4.10a. The SVM classifier provided 94.44 % accuracy for the dataset (even though all available features are used in SVM).

The classification of the same dataset is carried out using evolutionary algorithm and GA-KNN classifier in [9] reporting 98.24 % accuracy. In the work, 38 samples are used for training and 34 samples are used for testing. The FRSC algorithm is tested using the same sample divisions achieving 100 % classification accuracy (Table 4.4). Fifty gene profiles (features) are used in [9], whereas 35 gene profiles (15 selected features per class) are used in the FRSC algorithm.

4.3.1.2 Tumor Versus Normal Sample Classification

In [12], tumor detection is done using MicroRNA expression profiles.The dataset [5] 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 human tumor and normal samples. The classifier is utilized for the prediction of tumor in mouse lung samples [12]. 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 or normal samples (43 tumor and 32 normal samples) and the testing is done using mouse lung tumor and normal samples (12 samples). The kNN classifier provided 100 % correct detection of tumor (Table 4.4), when 131 markers are used [12].

Being a detection problem (rather than classification), 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 % 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 carried out using the human tumor and 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 fuzzy-rough algorithm provided 96.67 % classification accuracy when a total of 35 selected features are used (Table 4.3). Figure 4.10a 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 %.

4.3.1.3 Lung Cancer Classification

Lung cancer dataset [7] is used by Gordon et al. [8] for gene expression ratio based diagnosis of mesothelioma (MPM) and adenocarcinoma (ADCA). 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 classification. Three MPM and 15 ADCA samples are used in one subset for the 10 fold cross validation (Each subset contains 18 samples. The execution is repeated until all the 181 samples are tested). The FRSC algorithm provided 100 % classification accuracy (Table 4.3) 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. 4.10b. The SVM classifier also provided 100 % accuracy for the dataset.

In [8], 99 % classification accuracy is achieved for the lung cancer dataset, when 16 MPM and 16 ADCA samples are used for training, whereas the FRSC algorithm provided 100 % accuracy, when trained in a similar manner (Table 4.4). Six genes are used in [8] as the method is based on gene expression ratios. The FRSC algorithm without gene expression ratios needs 24 genes to provide 99 % accuracy and 44 genes for 100 % accuracy.

4.3.1.4 Small Round Blue Cell Tumor Classification

The dataset used is NCI's dataset [10] of small round blue cell tumors (SRBCTs) of childhood [11]. 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) in the database. The dataset consists of 2,308 genes. The FRSC algorithm is tested using the 200 individually discriminatory genes (IDGs)[6] identified by Xuan et al. [24]. For the 10 fold cross test, 1 BL, 3 EWS, 2 NB and 2 RMS are considered in one subset (Each subset contains 8 samples. The execution is repeated until all the 83 samples are tested). The classification results are provided in Table 4.3. The FRSC algorithm provided an accuracy of 98.75 %, when a total of 103 features are used. Figure 4.10b shows the variation in classification accuracy with respect to the number of selected features per class. For the SRBCT dataset, the SVM provided similar performance (98.80 % accuracy).

The classification of the SRBCT dataset is carried out in [11], using neural network (NN) classifier with 96 features, and in [24] using multilayer perceptron (MLP) classifier with 9 features. These works used 63 samples for training and 20 samples for testing, and achieved 100 and 96.9 % classification accuracies respectively. The FRSC algorithm provided an accuracy of 100 % when the same training and testing samples are used, with 103 features (35 selected features per class). The FRSC

[6] These genes are listed in the additional material Table S1 of [24].

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) [24] are used.

4.3.1.5 Central Nervous System Embryonal Tumor Classification

The gene expression based prediction of central nervous system embryonal tumor is reported in [18]. The multiple tumor classes are predicted using the k Nearest Neighbor (kNN) algorithm. There are five classes namely 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 7,129 gene profiles. The evaluation method used is *leave one out cross validation*, similar to [18]. The accuracy achieved by the FRSC classifier is 85.71 %, whereas that achieved by the kNN classifier is 83.33 % (Table 4.3). The FRSC algorithm used 90 selected genes (markers) per class whereas [18] used 10 markers per class (the FRSC algorithm provided lesser accuracy when 10 markers per class are used—Fig. 4.10(c)). The accuracy provided by the SVM classifier for this dataset is 80.95 %.

4.3.2 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) [19], which are extracted using a computational model of the ventral stream of visual cortex [19, 20]. A total of 1000 C_2 SMFs are extracted. Two fold cross validation is done for hand posture and face datasets. 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 [23]), both implemented using the C_2 SMFs (Table 4.5).

4.3.2.1 Hand Posture Recognition

The hand posture dataset considered is Jochen-Triesch hand posture dataset [21]. Ten classes of hand postures, performed with 24 different persons, against light background, are considered for classification. The hand postures vary in size and shape. The dataset has a total of 240 samples with 24 images in each class. The FRSC algorithm is tested by dividing the dataset equally into training and testing samples (12 images in each class). The algorithm is repeated by interchanging the training and testing sets, and the average results are reported (two-fold cross test).

Table 4.5 Summary and comparison of cross validation test results—hand posture, face and object recognition

Dataset	Accuracy (%) presented method	# features/class	Total # features	Average MC	Accuracy (%) PCA	Acc uracy (%) 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

The FRSC algorithm provided 100 % accuracy, whereas PCA provided an accuracy of 98.75 % (Table 4.5). The SVM classifier also provided 100 % accuracy. However, all the 1000 features are used for classification using PCA and SVM, whereas the FRSC 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. 4.10d. 100 % accuracy is achieved with 20 or more selected features per class.

4.3.2.2 Face Recognition

The face dataset considered is a subset of the Yale face database B [4], which contains 10 classes of face images, taken with different lighting directions. It consists of 640 frontal face images, 64 in each class. The algorithm is tested in a similar manner as done for the hand posture dataset. The FRSC and the SVM classifiers provided 100 % accuracy, whereas the PCA classifier provided 99.7 % accuracy (Table 4.5). The whole set of features (1000) is used in PCA and SVM. The FRSC algorithm is tested by varying the number of selected features (Fig. 4.10d). 100 % accuracy is achieved by the FRSC algorithm with 7 selected features per class (total of 56 features).

4.3.2.3 Object Recognition

The classes considered in object recognition are human frontal face, motorcycle, rear car, and airplane [3]. The dataset contains a total of 3,479 images (450 faces, 800 motor cycles, 1,155 cars, and 1,074 air-planes). The training set is formed with 100 randomly selected images from each class, and the testing is done using the remaining 3,079 images. The reported results are the averages over ten runs. The FRSC algorithm provided an average classification accuracy of 94.96 %, whereas PCA provided 88.37 % accuracy (Table 4.5). The SVM classifier provided an accuracy of 94.84 %, equivalent to that of the FRSC algorithm. All the 1000 features are used in PCA and SVM, whereas the FRSC algorithm needed only 30 features (8 selected features per class). The recognition accuracy saturated at 94.96 % when 8 or more selected features per class are used (Fig. 4.10d).

4.4 Summary

A feature selection and classification algorithm based on the concept of fuzzy-rough sets is presented in this chapter. 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. A measure of the quality of classification, the *margin of classification*, is defined for the FRSC classifier.

The performance of the FRSC algorithm is evaluated with two types of multiple feature datasets namely cancer and image datasets, and compared with relevant classification techniques. The FRSC algorithm provided classification accuracy which is equivalent to or better than those 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 initially 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 FRSC classifier is effective in cancer and tumor classification. 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 the FRSC algorithm, the initial (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, the initial cluster center is denser than the second, which in turn is denser than the third). Modification to the FRSC algorithm is presented in Chap. 5 leading to a fuzzy-rough multi-cluster (FRMC) classifier [17].

Acknowledgments Figures and tables in this chapter are adapted from Applied Soft Computing, Vol.11, Issue No.4, Pramod Kumar Pisharady, Prahlad Vadakkepat, and Loh Ai Poh, 'Fuzzy-Rough discriminative feature selection and classification algorithm, with application to microarray and image datasets', Page Nos. 3429–3440, Copyright (2011), with permission from Elsevier.

References

1. C.C. Chang C.J. Lin, Libsvm: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/cjlin/libsvm/
2. S. Chiu, Fuzzy model identification based on cluster estimation, J. Int. Fuzzy Syst. **2**(3), 18–28 (1994)
3. R. Fergus, P. Perona, 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
4. A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
5. T.R. Golub, D.K. Slonim, C. Huard, M. Gaasenbeek, P. Tamayo, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Cancer program data sets, (1999), http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi
6. 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, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by geneexpression monitoring. Science **286**, 531–537 (1999)
7. G.J. Gordon, R.V. Jensen, L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, Supplemental information of gordon et al. paper, (2002), http://www.chestsurg.org/publications/2002-microarray.aspx
8. G.J. Gordon, R.V. Jensen, L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Res. **62**, 4963–4967 (2002)
9. T. Jirapech-Umpai, S. Aitken, Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes, BMC Bioinform. **6**, 148 (2005)
10. J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Microarray project (2001), http://research.nhgri.nih.gov/microarray/Supplement
11. J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat. Med. 7 **6**, 673–679 (2001)
12. 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, T.R. Golub, Micro rna expression profiles classify human cancers, Nature **435**, 834–838 (2005)
13. Z. Pawlak, Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
14. Z. Pawlak, Rough classification, Int. J. Man-Mach. Stud. **20**, 469–483 (1984)
15. P.K Pisharady, Computational intelligence techniques in visual pattern recognition, Ph.D. Thesis, National University of Singapore (August, 2011)
16. P.K. Pisharady, P. Vadakkepat, A.L. Poh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft Comput. **11**(04), 3429–3440 (2011)
17. P.K. Pisharady, P. Vadakkepat, A.L. Poh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **07**(03), 331–356 (2010)
18. 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. Lett. Nat. **415**, 436–442 (2002)
19. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Int. **29**(3), 411–426 (2007)

20. T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex. *Conference on Computer Vision and Pattern Recognition*, ed by C. Schmid, S. Soatto, C. Tomasi, (San Diego, CA, 2005) pp. 994–1000
21. J. Triesch, C. Malsburg, Sebastien marcel hand posture and gesture datasets: Jochen triesch static hand posture database (1996), http://www.idiap.ch/resource/gestures/
22. J. Triesch, C. Malsburg, Robust classification of hand postures against complex backgrounds, in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, (Killington, VT, USA, 1996) pp. 170–175
23. M. Turk, A. Pentland, Eigenfaces for recognition. J. Cog. Neurosci. **3**, 71–86 (1991)
24. J. Xuan, Y. Wang, Y. Dong, Y. Feng, B. Wang, J. Khan, M. Bakay, Z. Wang, L. Pachman, S. Winokur, Y. Chen, R. Clarke, E. Hoffman, Gene selection for multiclass prediction by weighted fisher criterion, EURASIP J. Bioinform. Syst. Biol. **2007**, (2007)
25. A.L. Zadeh, Fuzzy sets, Inform. Control **8**(3), 338–353 (1965)
26. S. Zhao, E.C.C. Tsang, D. Chen, X. Wang, Building a rule-based classifier-a fuzzy-rough set approach. IEEE Trans. Knowl. Data Eng. **22**(5), 624–638 (2010)

Chapter 5
Hand Posture and Face Recognition Using Fuzzy-Rough Approach

The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom

Isaac Asimov

Abstract A *fuzzy-rough multi cluster* (FRMC) classifier for the recognition of hand postures and face is presented in this chapter. Features of the image are extracted using the computational model of the ventral stream of visual cortex. The recognition algorithm translates each quantitative value of the feature into fuzzy sets of linguistic terms using membership functions. The membership functions are generated by the fuzzy partitioning of the feature space into fuzzy equivalence classes, using the feature cluster centers generated by the subtractive clustering technique. A rule base generated from the lower and upper approximations of the fuzzy equivalence classes classifies the images through a voting process. Using Genetic Algorithm (GA), the number of features required for classification is reduced by identifying the predictive image features. The *margin of classification*, which is a measure of the discriminative power of the classifier, is used to ensure the quality of classification process. The fitness function suggested assists in the feature selection process without compromising on the classification accuracy and margin. The algorithm is tested using two hand posture and three face datasets. The algorithm provides good classification accuracy, at a less computational effort. The selection of relevant features further reduced the computational costs of both feature extraction and classification algorithms, which makes it suitable for real-time applications. The performance of the algorithm is compared with that of Support Vector Machines.

Keywords Hand posture recognition · Face recognition · Human robot interaction · Feature selection · Fuzzy-rough sets · Biologically inspired vision · Genetic algorithm · FRMC algorithm

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_5,
© Springer Science+Business Media Singapore 2014

5.1 Introduction

The multi-class recognition of hand postures and human faces is considered in this chapter. The FRMC classification algorithm classifies the images at less computational cost. The features of the images are extracted using a cortex-like mechanism [1]. The algorithm 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 [2–4]. These rules are utilized for classifying images, through a voting process. The FRMC 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 or features used to construct the classifier. Genetic Algorithm (GA) is useful in reducing the number of features required for classification, by identifying and removing irrelevant and redundant features. The FRMC classifier is a margin classifier with good *margin of classification*. Margin of classification (MC) is a measure of the distance from the classification boundary and is used in the fitness function. The 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 reduces the number of features to be extracted. The increased speed of feature extraction and classification process makes the algorithm suitable for real-time applications.

The FRMC algorithm is applied to three face datasets (subsets of the Yale face database B [5], the color FERET database [6] and the CMU face dataset [7]) and two hand posture datasets (the NUS hand posture dataset-I [4] and Jochen Triesch hand posture dataset [8]). The FRMC algorithm is capable of recognizing hand postures in real-time, and the performance is comparable with that of Support Vector Machines (SVM).

5.2 The Fuzzy-Rough Multi Cluster Classifier

The Fuzzy-Rough Multi Cluster (FRMC) classifier algorithm is discussed in this section. The recognition of a pattern in an image consists of two processes, feature extraction and classification. The FRMC algorithm is simple and computationally less intensive. The algorithm is capable of generating a classifier automatically from the training set using a fuzzy-rough set approach.

Figure 5.1 shows an overview of the various components of the FRMC algorithm. Classification rules are generated in the training phase, after extracting the image features. Discriminative image features are identified and selected in the feature selection phase. Images are classified using the generated classification rules.

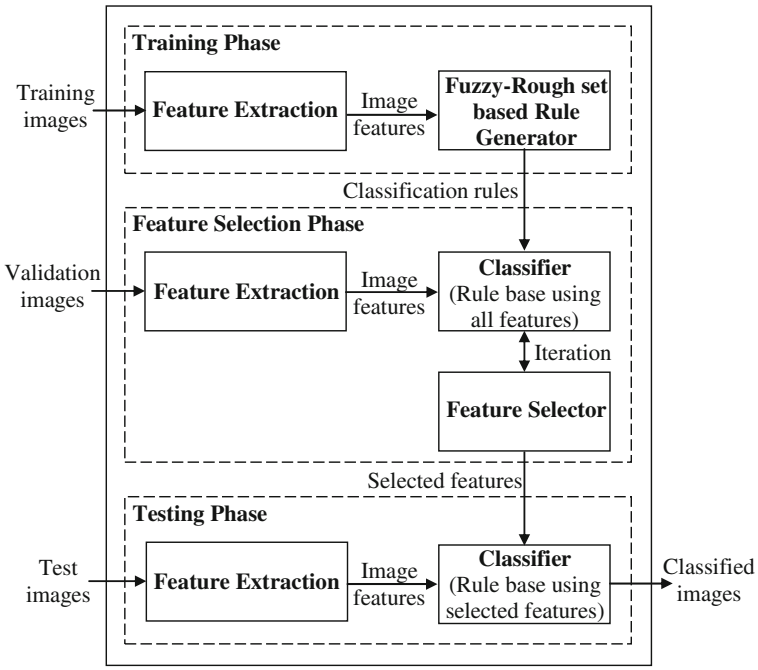


Fig. 5.1 Overview of the FRMC classifier algorithm

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

This section explains the rule generation process in the FRMC classifier. Figure 5.2 shows the training phase. Image features are extracted using the C_2 feature extraction system (Sect. 2.2.1). Fuzzy discretization of feature space is done using the cluster centers identified by subtractive clustering technique [9]. The lower and upper approximations of the feature fuzzy sets are found out and the classification rules are generated. Following subsections elaborate the various phases associated with the algorithm development.

5.2.1.1 Generation of Fuzzy Membership Functions

The fuzzy membership functions are created using the feature cluster centers. These membership functions are utilized to partition the data into fuzzy equivalence classes. The subtractive clustering technique [9], which is an extension of Yager's mountain clustering method [10], is utilized to identify the feature cluster center points.

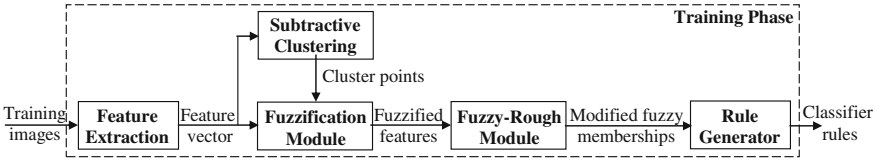


Fig. 5.2 Training phase of the FRMC algorithm

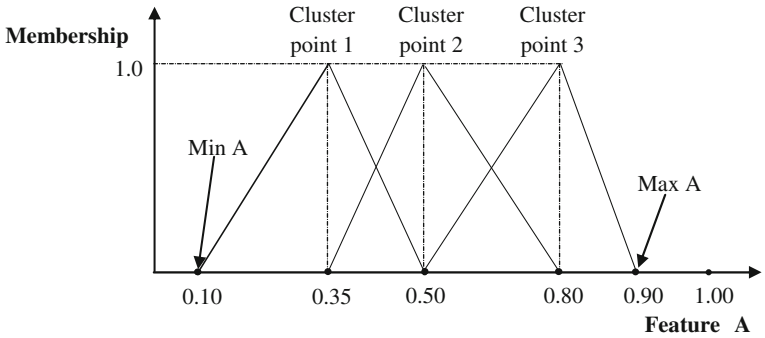


Fig. 5.3 Formation of membership functions from cluster center points

The cluster centers are identified for all the features in every class. In subtractive clustering, the number of clusters increases with smaller radius leading to increased computational requirements. The clustering radius used in the FRMC algorithm is 0.5. The number of cluster centers generated varies from 1 to 3.

The peaks of the triangular membership functions are placed at the identified data cluster centers. The left (right) sides of the membership functions span to the adjacent left (right) data cluster centers. The minimum (maximum) feature value positions the left (right) side of the leftmost (rightmost) membership function. For example, triangular membership functions in Fig. 5.3 are for a set of cluster points $cp_1 = \{0.35, 0.5, 0.8\}$ for feature A which is within a range of $[0.1, 0.9]$.

A set of membership functions is generated 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 derived.

5.2.1.2 Modified Fuzzy Membership Functions

The same fuzzy equivalence class contains samples from different output classes which results in fuzzy-rough uncertainty. The FRMC classifier algorithm finds the lower and upper approximations of fuzzy equivalence classes. The limiting values of memberships which partition definite and possible members of the output class are utilized to generate the classification rules.

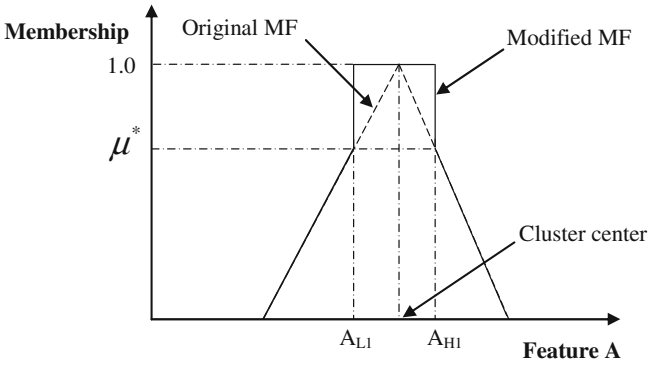


Fig. 5.4 Modified fuzzy membership function

The limiting values of the memberships (Fig. 5.4) are calculated as follows. Let MF be the fuzzy set associated with a particular cluster center of a feature in a class and,

$$\mu^{max}(C_i) = \max\{\mu_{MF}[x_{C_i}(l)]\}, \tag{5.1}$$

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

$$C_{max} = \text{argmax}_{C_i}\{\mu^{max}(C_i)\}, \tag{5.2}$$

then,

$$\mu^* = \max_{C_i \neq C_{max}} \{\mu^{max}(C_i)\}. \tag{5.3}$$

μ^* is the maximum of the membership values associated with data samples belonging to classes other than C_{max}. The limiting value of a 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 a 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 output class (C_{max}). The rule always holds true for 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 classifier reliability, only rules those classify two or more training samples are stored in the rule base. In order to classify a new sample, the rule base requires generalization whicn is elaborated in the following Sect. 5.2.1.3.

5.2.1.3 Generation of Classification Rules

The generalized classification rules are discussed in this section. Let $\{\mu^*_{ijk}\}$ be the set of membership values obtained (Eq. 5.3) where,

i $1, \ldots, p : p$ number of classes,
j $1, \ldots, q : q$ number of features,
k $1, \ldots, r : r$ number of clusters corresponding to the jth feature of the ith 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:

$$IF\ [\mu_{MF_{ijk}}(x) > \mu^*_{ijk}]\ THEN\ [N_{C_i} = N_{C_i} + 1] \tag{5.4}$$

where,

x the sample to be classified.
N_{C_i} the number of votes for class C_i,

Rule 2:

$$C = argmax_{C_i}\{N_{C_i}\} \tag{5.5}$$

where,

C is the class to which the sample belongs to.

For example, the number of votes N_{C_1}, N_{C_2} and N_{C_3} are calculated for a sample from three classs dataset (*Rule 1*). *Rule 2* identifies the class with maximum votes. *Rules 1* and *2* form the classifier rule base, keeping the algorithm computationally simple.

The FRMC classifier is a *margin classifier* providing a measure, namely *margin of classification (MC)*, for the distance between a sample and corresponding classification boundary. The MC for a sample is defined as,

$$MC = N_p - N_{nMAX} \tag{5.6}$$

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 $N_{C_3} = 10$ (for a three class dataset). The MC for the sample is $90 - 10 = 80$. The sample receives 90 positive votes.[1] The number of negative votes are 5 and 10 for the second and third classes respectively. A positive margin indicates correct classification whereas negative margin indicates misclassification. In Sect. 5.2.2, the MC is used to identify the relevant features of an image.

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

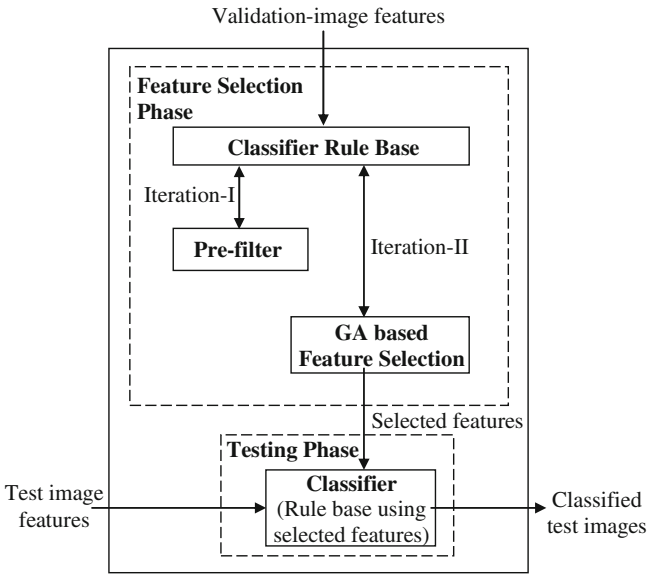


Fig. 5.5 Feature selection and testing phases

5.2.2 Genetic Algorithm Based Feature Selection

The number of features used to describe a pattern determines the size of the search space to explore [11]. An abundance of features increases the size of the search space, thereby increasing the time needed for classification. The cost of classification is sensitive to the number of features.

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

In the FRMC algorithm, feature selection is carried out in two stages. The first stage is a pre-filter which filters out misleading features. The second stage is a GA based feature selection algorithm which removes irrelevant and redundant features. The feature selection process is depicted in Fig. 5.5.

Each classification rule in the FRMC classifier corresponds to an image feature. Classification rules provide positive or negative votes to samples. The voting helps to identify the relevant features (features providing more positive votes) of an image. The margin of classification (5.6) is utilized for feature selection. Selection of relevant features improves the performance of the recognition system and reduces the computational burden of the FRMC algorithm.

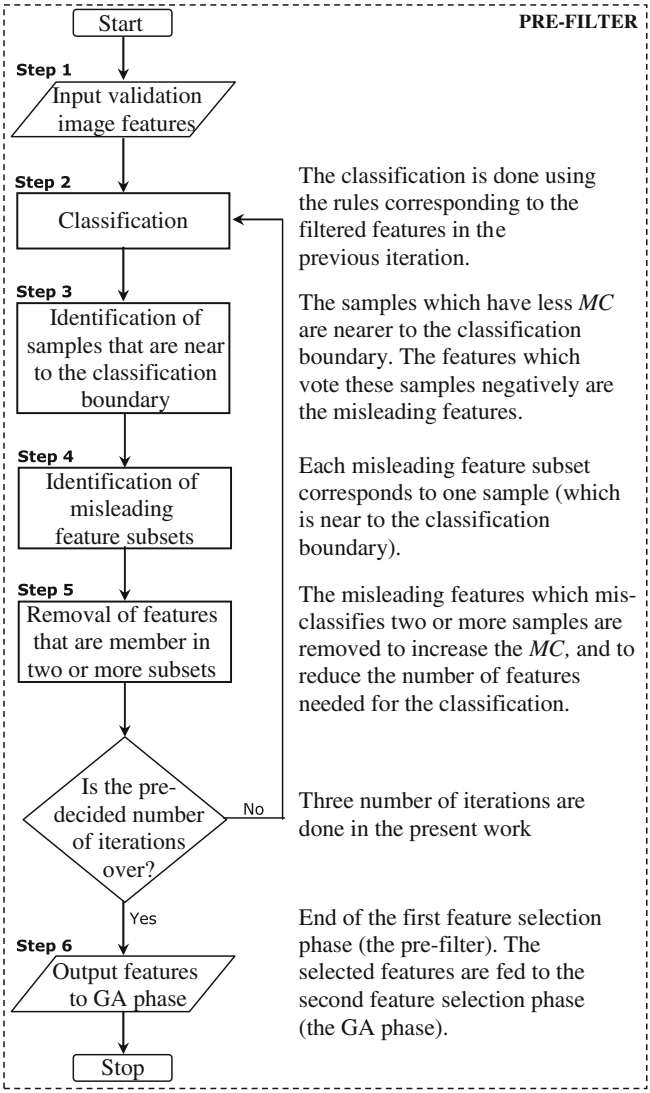


Fig. 5.6 Flowchart of the pre-filter

5.2.2.1 Pre-Filter: Weeding-Out Misleading Features

Pre-filter (Fig. 5.6) identifies and removes features giving negative votes. The features negatively vote for two or more samples as per (5.4) are weeded out. The validation data is sorted in ascending order of the MC and those samples with less *MC* are considered (samples with less *MC* have more negative votes and are nearer to the classification boundary). The features voting negatively for these samples are identified and removed.

The process is iterated a few times by repeating the validation procedure. The number of iterations needed depends on the nature of dataset. More number of iterations are required for datasets with large number of misleading features. For the face and hand gesture datasets, the pre-filter is iterated three times. The pre-filtering process is simple, having significant effect in increasing the MC, leading to reduction in computational load at the GA based feature selection stage.

5.2.2.2 Feature Selection Using Genetic Algorithm

Genetic Algorithms (GAs) are widely used for feature selection and extraction in machine learning [12–16]. In the present work, GA is utilized to remove the irrelevant and redundant features. A fitness function is defined for the FRMC classification algorithm in Sect. 5.2.1. The fitness function considers the accuracy, cost and MC.

Each chromosome in the GA population represents a set of features used for classification. The chromosome consists of N_p bits and each bit represents a specific feature. When a particular bit is unity, it indicates the presence of a feature and zero indicates its absence. In the FRMC algorithm development, the value of N_p is 200. Elitism technique is utilized to create new population and the fitness function Eq. (5.7) is minimized.

$$F = w_1 * N_1 + w_2 * N_2 - w_3 * N_3, \tag{5.7}$$

where,

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

In 5.7, N_1 represents the accuracy of classification and N_2 represents the computational expense (presence of more features leads to more number of classification rules and so is the computational cost). N_3 gives preference to those chromosomes whcih provide better MC. 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 tuned depending on the dataset. An empirical guideline is to select the weights (w_1, w_2, w_3) such that $w1 > w2 > w3$. The weighing factors (w_1, w_2, w_3) are associated with the accuracy of classification, number of features and MC, respectively. The accuracy of classification is given preference over the number of features which in turn is preferred over the MC. The weighing factors can have values in a range of 0–1. In the FRMC algorithm, $w_1 = 1.0$, $w_2 = 0.6$, and $w_3 = 0.4$.

The crossover and mutation rates used are 0.9 and 0.01, respectively. The GA population size is kept at 100. Full classification is possible by considering all features. GA is used to identify the minimum number of relevant features for full classification. With the GA process, the relevant features are identified by removing the irrelevant, mis-leading and redundant features. Iterations are carried on

till a non-zero classification error occures or until a pre-defined maximum number of iterations (which is 50 in the FRMC algorithm). When the iteration is stopped after the occurance of a non-zero classification error, the features used in the previous iteration are selected. The number of features needed for classification is decided by taking the average over ten GA runs.

A detailed flowchart of the development of the FRMC classifier algorithm is provided in Fig. 5.7.

5.2.3 FRMC Classifier Testing Phase

Figure 5.8 shows the flowchart of the testing phase of the FRMC classifier. The selected features of the unlabeled test images are extracted using the C_2 feature extraction system. The membership values of features in different classes are calculated with the membership functions generated in the training phase. The membership values of features are compared with limiting values of different class memberships (5.3) and classification is done using *Rules 1* and *2*. Each of the execution steps of the classifier consists of a number of comparisons (equal to the number of feature cluster centers), using classification rules, which makes the algorithm computationally simple.

The classification results are discussed in Sect. 5.3.

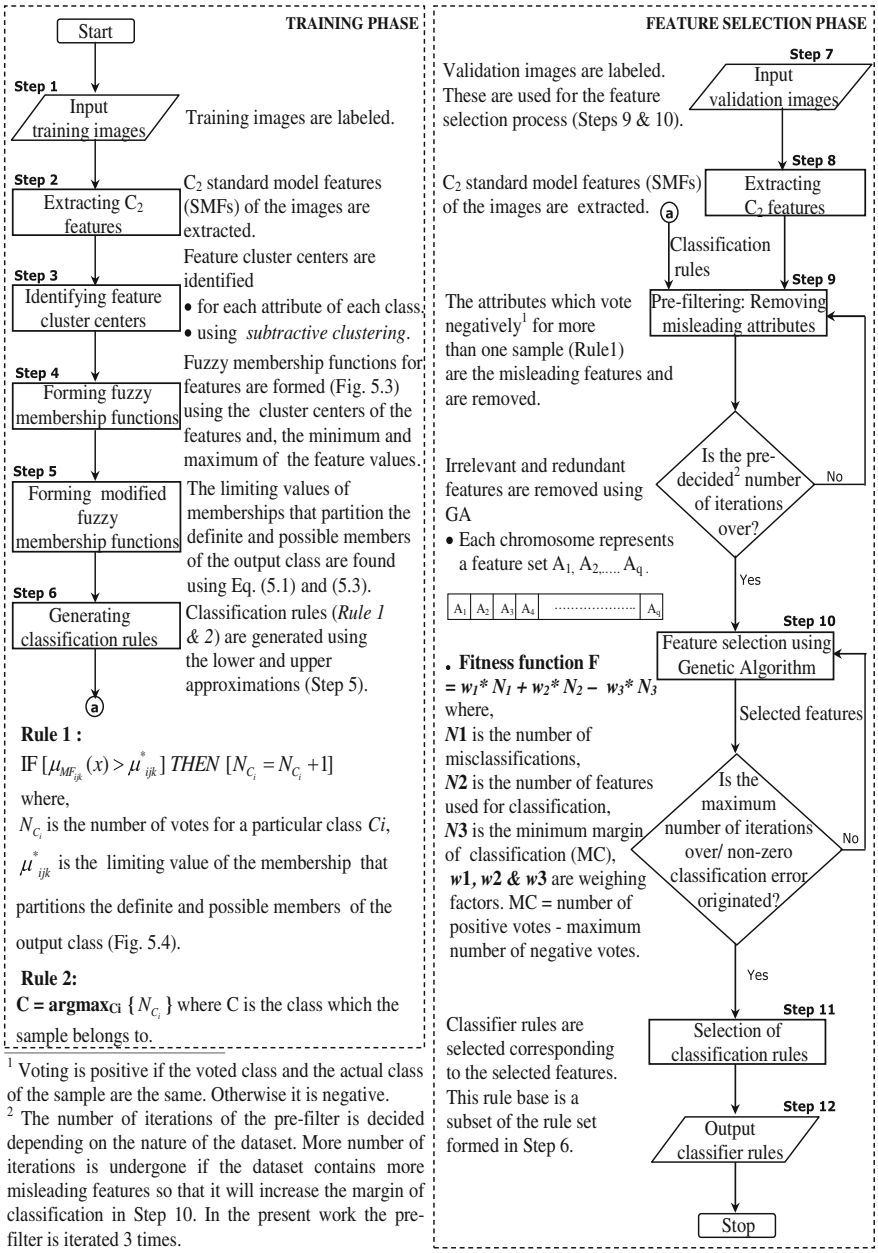
5.2.4 Computational Complexity Analysis

The input parameters of the recognition system are the number of classes and image size. The complexity of the recognition algorithm is due to feature extraction and classification. This section discusses the computational complexity of the FRMC classification algorithm.

The pseudo code of the FRMC classifier is shown in Fig. 5.9. Let p be the number of classes and q be the number of selected features. 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 receives maximum number of votes. The overall complexity of the algorithm is in polynomial time, $O(pq)$. As the feature extraction algorithm provides a fixed number of features which is independent of the input image size, the effective complexity of the classification algorithm reduces to $O(p)$.

5.3 Experimental Evaluation of FRMC Classifier

The FRMC algorithm is tested using three face datasets and two hand posture datasets (Table 5.1). The datasets contain ten classes with equal number of images. Each dataset is divided into N subsets. The images in the first subset are used for classifier development and the remaining $N-1$ subsets for testing. The classifier is developed



[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.

Fig. 5.7 Flowchart of the development of the FRMC classifier algorithm

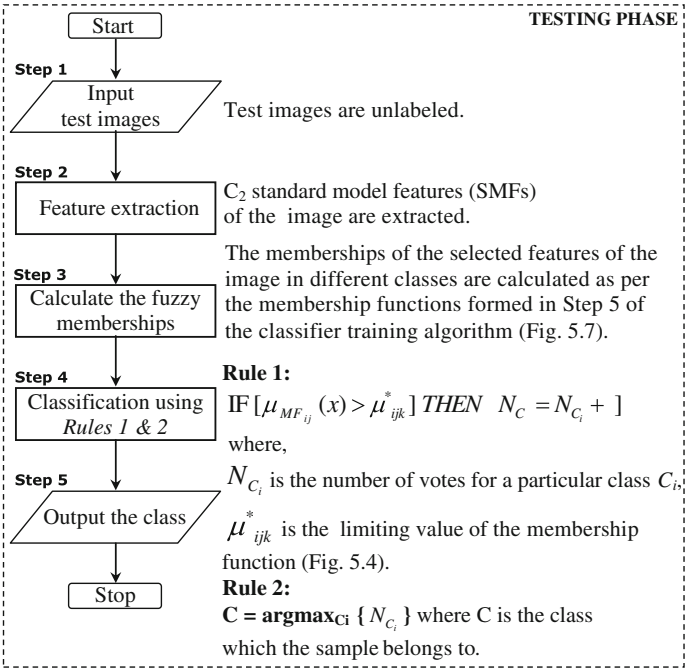


Fig. 5.8 Flowchart of the testing phase

with the second subset and the algorithm is tested with remaining $N - 1$ subsets. Experiments are repeated in a similar fashion, N times, until each of the subset is used for the development. The average accuracy achieved over N runs is reported. The values of N are two and four ($N = 2, 4$ respectively corresponds to 50, 25 % images in each subset). The data subset used for classifier development is equally divided into training and validation sets. Classification results are compared with that of Support Vector Machines (SVM) with polynomial kernel [17].

5.3.1 Face Recognition

The FRMC algorithm is tested using three different face datasets with variations in lighting direction (Yale face database B [5]), in pose (color FERET database [6]), and in illumination (CMU face dataset [7]).

The Yale face database B [5] contains ten classes of face images captured with different lighting directions (Fig. 5.10a). The subset used has 640 frontal face images. A comparison of the achieved results with that of SVM using C_2 features is provided in Table 5.2. The FRMC classifier provided full classification of the dataset whereas

```
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} ;
```

Fig. 5.9 Pseudo code of the classifier

Table 5.1 Face and hand posture dataset details

Dataset	Source	# images
Face dataset-1	Subset of Yale face database B [5]	640
Face dataset-2	Subset of color FERET database [6]	240
Face dataset-3	Subset of CMU face dataset [7]	240
Hand posture dataset-1	NUS hand posture dataset-I	240
Hand posture dataset-2	Jochen Triesch hand posture dataset [8]	480

SVM provided 99.38 % accuracy, when $N = 2$. The FRMC classifier utilized 86 features, whereas SVM utilized all the 1000 features.

The color FERET database [6] subset contains 240 face images in ten classes with pose variations (Fig. 5.10b). A comparison of results with SVM is provided in Table 5.2. The FRMC classifier needed 102 features (when N = 2) for classification and provided better accuracy compared to the SVM classifier.

The third face dataset considered is a subset of the CMU face dataset [7] with good amount of illumination variation (Fig. 5.10c). The dataset consists of 240 frontal face images in 10 classes. The recognition results are provided in Table 5.2. Both the FRMC and SVM classifiers classified the dataset fully. However, the FRMC classifier required lesser number of features.

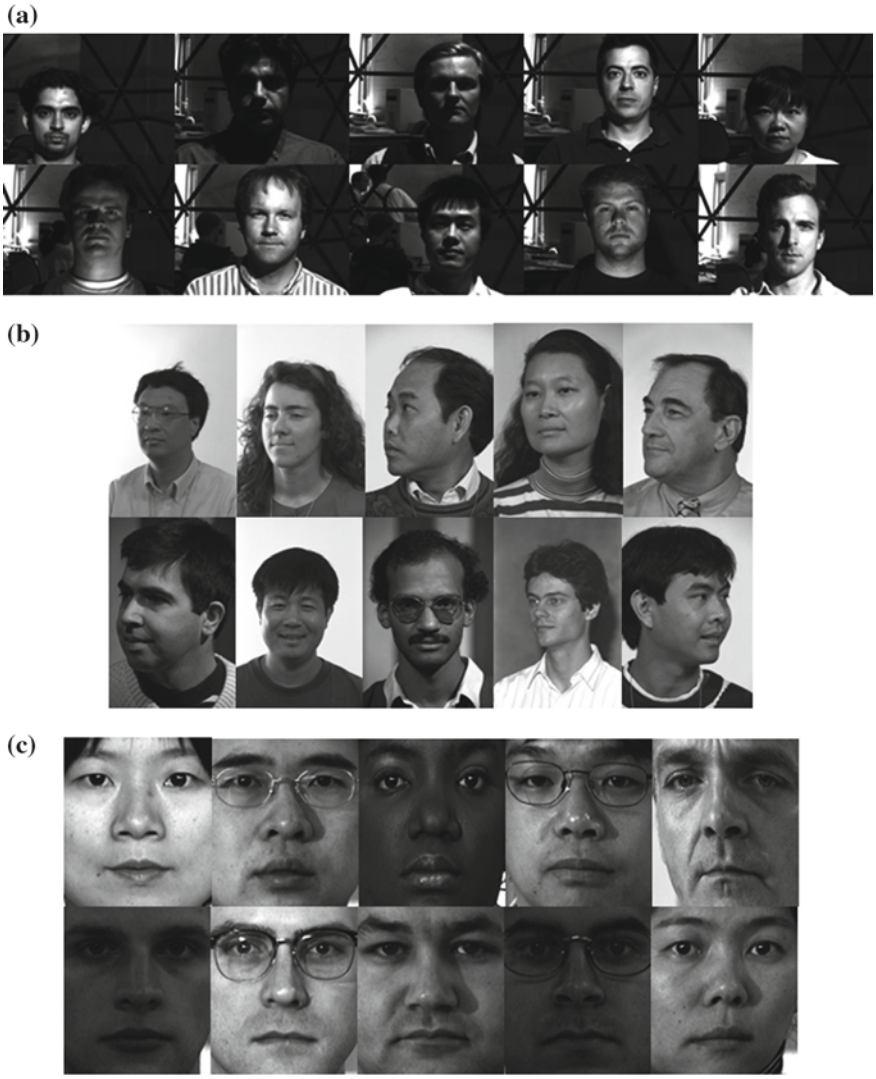


Fig. 5.10 Sample images from **a** Yale face dataset [5], **b** FERET face dataset [6], and **c** CMU face dataset [7]

5.3.2 Hand Posture Recognition

A new dataset namely the NUS hand posture dataset-I,[2] is created with ten classes of hand postures. The dataset consists of 24 gray-scale images (160×120 pixels) in each posture, with variations in position and size of hand within the image frame.

[2] The dataset is available for free download: http://www.vadakkepat.com/NUS-HandSet/.

Table 5.2 Recognition results—face datasets

Dataset	No. of training samples	FRMC		SVM	
		Accuracy (%)	No. of features	Accuracy (%)	No. of features
A subset of Yale face dataset [5]	160 (N = 4)	99.16	95	98.33	
	320 (N = 2)	100	86	99.38	
A subset of color FERET database [6]	60 (N = 4)	88.89	111	88.33	
	120 (N = 2)	91.66	102	90.83	1000
A subset of CMU face dataset [7]	60 (N = 4)	99.44	62	98.88	
	120 (N = 2)	100	55	100	

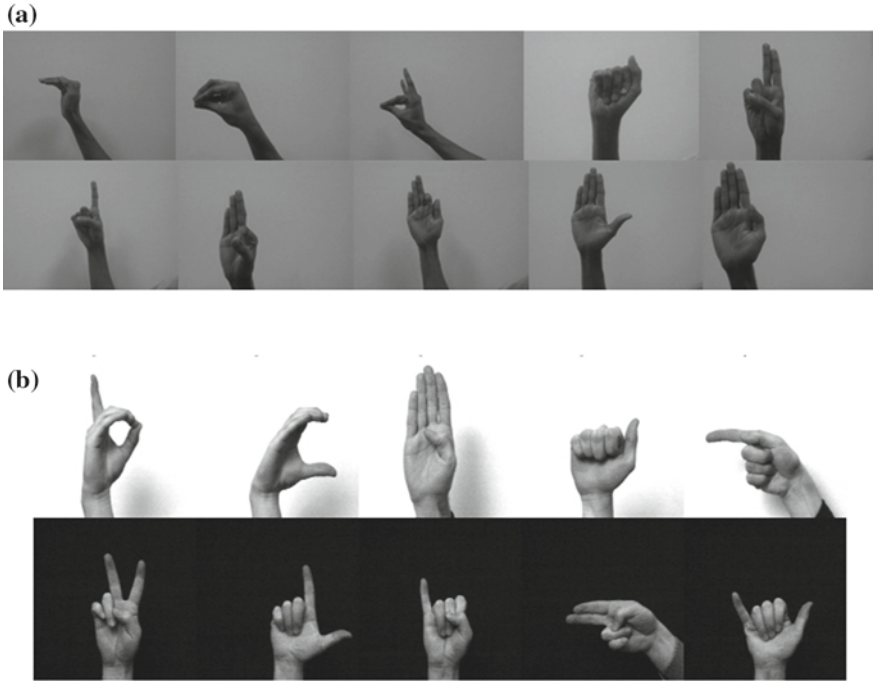


Fig. 5.11 Sample hand posture images from **a** NUS hand posture dataset-I [4], and **b** Jochen Triesch dataset [8]

The hand postures are selected in such a way that the inter class variation of postures is less, making the recognition task challenging. Sample images from the dataset are shown in Fig. 5.11a. The superior performance of the FRMC classifier is evident from the recognition results (Table 5.3). The algorithm utilized only 92 number of features (when N = 2) for classification. Misclassifications occured are mostly among similar classes.

Table 5.3 Recognition results—hand posture datasets

Dataset	No. of training samples	FRMC		SVM	
		Accuracy (%)	No. of features	Accuracy (%)	No. of features
NUS hand posture dataset-I	60 (N = 4)	91.66	98	91.80	
	120 (N = 2)	93.33	92	92.50	
Jochen Triesch hand posture dataset [18]	120 (N = 4)	95.83	72	94.44	1000
	240 (N = 2)	98.75	63	97.91	

Table 5.4 Comparison of computational time

	FRMC	SVM
Feature extraction	1.92 s	7.43 s
Classification	0.96 ms	3.24 ms

The second hand posture dataset considered is Jochen Triesch hand posture dataset [8]. The dataset has ten classes of hand postures against light and dark backgrounds (Fig. 5.11b) from 24 different subjects (total of 480 images). The images vary in hand size and posture shape. The FRMC classifier provided better classification accuracy in comparison with SVM (Table 5.3). The FRMC classifier required only 63 selected features (when N = 2), whereas the SVM classifier utilized all the 1000 features.

5.3.3 FRMC Classifier Online Implementation

The FRMC recognition system is suitable for online implementation due to the classifier's good realtime performace. For evaluating the online performance, the FRMC algorithm is trained with the NUS hand posture dataset-I. The testing images are accessed using a web-camera. The images are resized and converted to gray scale.

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

Table 5.4 compares the average computational time of the FRMC algorithm with that of SVM. The total time for recognition includes the time taken for feature extraction and classification. In the SVM classifier all the 1,000 features are utilized for classification. The FRMC algorithm utilized a subset of the 1,000 features leading to reduced time for feature extraction and classification. The FRMC algorithm required lesser time for classification even when all the 1,000 features are utilized.

Each C_2 SMF corresponds to a particular prototype patch with a specific patch size (Sect. 2.2.1). The selection of relevant features lead to identifying discriminative prototype patches, enhancing shape selectivity of the algorithm. The feature

selection process reduces the feature extraction time. Longer feature extraction time is a limitation of the algorithm proposed by Serre et al. [1]. The significant reduction in feature extraction and classification time makes the FRMC algorithm suitable for real-time applications.

5.4 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 presented in this chapter. The fuzzy membership functions and 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 fitness function utilized reduces the number of features required for classification, without compromising the classification accuracy. The performance of the algorithm is evaluated with some well-known datasets and is compared with that of an SVM classifier. The FRMC algorithm provided good recognition accuracy for all the datasets considered, at a less computational cost, making it suitable for real-time applications.

Acknowledgments Figures and tables in this chapter are adapted from 'Hand posture and face recognition using a Fuzzy-Rough Approach', Pramod Kumar Pisharady, Prahlad Vadakkepat, and Loh Ai Poh, International Journal of Humanoid Robotics, Vol.7, Issue No.3, Copyright @ 2010, World Scientific Publishing Company.

References

1. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
2. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition, Ph.D. thesis, National University of Singapore (August, 2011)
3. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft Comput. **11**(04), 3429–3440, (2011)
4. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **07**(03), 331–356 (2010)
5. A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
6. P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, The feret database and evaluation procedure for face recognition algorithms. Image Vis. Comput. **16**(5), 295–306 (1998)
7. T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression database. IEEE Trans. Pattern Anal Mach. Intell. **25**(12), 1615–1618 (2003)
8. 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/

9. S. Chiu, Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst **2**(3), 18–28 (1994)
10. R. Yager, D. Filev, Generation of fuzzy rules by mountain clustering. J. Intell. Fuzzy Syst. **2**(3), 209–219 (1994)
11. M.H. Yang, N. Ahuja, Extraction and classification of visual motion patterns for hand gesture recognition, in, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Santa Barbara, CA, USA, 1998), pp. 892–897
12. F. Brill, D. Brown, W. Martin, Fast genetic selection of features for neural network classifiers. IEEE Trans. Neural Netw. **3**(2), 324–328 (1992)
13. H.D. Kim, C.H. Park, H.C. Yang, K.B. Sim, Genetic Algorithm Based Feature Selection Method Development for Pattern Recognition, in *Proceedings of SICE-ICASE International Joint Conference 2006*. (Bexco, Busan, Korea, 2006), pp. 1020–1025
14. J. Lu, T. Zhao, Y. Zhang, Feature selection based-on genetic algorithm for image annotation. Knowl. -Based Syst. **21**(8), (2008)
15. L. Rokach, Genetic algorithm-based feature set partitioning for classification problems. Pattern Recognit **41**, 1676–1700 (2008)
16. W. Siedlecki, J. Sklansky, A note on genetic algorithms for large scale feature selection. IEEE Transa. Comput. **10**, 335–347 (1989)
17. C. C. Chang and C. J. Lin, Libsvm: a library for support vector machines, (2001). http://www.csie.ntu.edu.tw/-cjlin/libsvm/
18. J. Triesch, C. Malsburg, Robust classification of hand postures against complex backgrounds, in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, (Killington, VT, USA, 1996), Oct 1996, pp. 170–175

Chapter 6
Boosting Based Fuzzy-Rough Pattern Classifier

Take up one idea. Make that one idea your life—think of it, dream of it, live on that idea. Let the brain, muscles, nerves, every part of your body, be full of that idea, and just leave every other idea alone. This is the way to success

Swami Vivekananda

Abstract The boosting-based fuzzy-rough multi-cluster (BFRMC) classification algorithm utilizing the rough set concepts of fuzzy lower and upper approximations is discussed in this chapter. The BFRMC algorithm transforms each quantitative value of a feature into fuzzy sets of linguistic terms using membership functions and calculates the fuzzy lower and upper approximations. The membership functions are generated from cluster points generated by the subtractive clustering technique. A *certain rule set* based on fuzzy lower approximation and a *possible rule set* based on fuzzy upper approximation are generated. Iterative rule learning based genetic algorithm is employed in combination with a boosting technique, for generating the possible rule set, incrementally optimizing each fuzzy classifier rule. The BFRMC classifier's performance on the datasets from the UCI machine learning repository is discussed.

Keywords Boosting · Fuzzy lower and upper approximations · Subtractive clustering · Iterative rule learning · BFRMC algorithm

6.1 Introduction

Inductive knowledge acquisition is a prime area of research in pattern recognition. Computational intelligence techniques are widely used in inductive machine learning. Fuzzy and rough sets are two computational intelligence tools used for decision making in uncertain situations. The BFRMC algorithm utilizes a fuzzy-rough set approach for decision making.

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_6,
© Springer Science+Business Media Singapore 2014

Fuzzy sets [1], a generalization of the classical sets, proposed by Zadeh in 1963, is a mathematical tool to model vagueness. Due to its similarity with human reasoning, fuzzy inferencing is effective in handling vague information. Fuzzy inference rules are traditionally generated from expert's knowledge. In case of non-availability of expert's knowledge, the data clusters are utilized to train the fuzzy membership functions. Subtractive clustering technique [2] is an attractive tool for determining data clusters. Subtractive clustering does not require prior knowledge of the number of clusters.

Pawlak introduced the rough set theory in early eighties, as a tool to handle inconsistencies among data [3, 4]. Rough set is a formal approximation of a set by a pair of sets, the lower and upper approximations. The definition of the approximations follows from an indiscernibility relation. No additional knowledge about 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 rough set approach has fundamental importance in artificial intelligence and cognitive science, especially in areas of machine learning, knowledge acquisition, decision analysis, inductive reasoning and pattern recognition.

Fuzzy and rough set theories are considered complementary in that they both deal with uncertainty: vagueness for fuzzy sets and indiscernibility for rough sets [5]. These two theories can be combined to form rough-fuzzy sets or fuzzy-rough sets [5, 6]. 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 [7–9].

In the boosting-based fuzzy-rough multi-cluster (BFRMC) classifier algorithm [10], fuzzy lower and upper approximations of samples are used for generating classifications rules. A *certain rule set* based on the fuzzy lower approximation and a *possible rule set* based on the fuzzy upper approximation are generated. A genetic algorithm based on iterative rule learning is employed in combination with a boosting technique, for generating the possible rule set, incrementally by optimizing one fuzzy classifier rule at a time. The performance of the BFRMC classifier for three datasets viz. Iris, Wine, and Glass from UCI machine learning repository is discussed. The algorithm provided good classification accuracy for all the three datasets.

6.2 Fuzzy-Rough Sets for Classification

Characteristics of various features in a dataset are useful in classifying patterns. However, presence of similar and overlapped features makes the classification difficult. Interclass feature overlaps and similarities lead to indiscernibility and vagueness. Rough set theory [3, 4] is useful for decision making in situations where indiscernibility is present, and, fuzzy set theory [1] is suitable when vague decision boundaries exist.

The concept of equivalence classes forms the basis for rough set theory which can be extended to fuzzy set theory to form fuzzy equivalence classes [5]. The concept of crisp equivalence class can be extended with the fuzzy similarity relation S within a universe. The relation S determines the similarity of two elements in the universe. Using the fuzzy similarity relation, the fuzzy equivalence class $[x]_S$ for objects close to x can be defined as $\mu_{[x]_S}(y) = \mu_S(x, y)$. When S is non-fuzzy, this definition shrinks to the normal definition of equivalence classes. A family of normal fuzzy sets generated through fuzzy partitioning, can form fuzzy equivalence classes [5]. The fuzzy *P-lower* and *P-upper approximations* are defined as [5]:

$$\begin{aligned} \mu_{\underline{P}X}(F_i) &= \inf_x \max\{1 - \mu_{F_i}(x), \mu_X(x)\}\ \forall i, \\ \mu_{\overline{P}X}(F_i) &= \sup_x \min\{\mu_{F_i}(x), \mu_X(x)\}\ \forall i, \end{aligned} \tag{6.1}$$

where F_i denotes a fuzzy equivalence class in U/P. The tuple $\langle \underline{P}X, \overline{P}X \rangle$ is a fuzzy-rough set.

The crisp positive region in rough set theory is defined as the union of the lower approximations. By the extension principle, the membership of an object $x \in U$, belonging to the fuzzy positive region is defined by (6.2).

$$\mu_{POS_P(Q)}(x) = \sup_{X \in U/Q} \mu_{\underline{P}X}(x). \tag{6.2}$$

Fuzzy equivalence classes and, fuzzy lower and upper approximations have been successfully applied in classification problems [11–14]. Most of these techniques are based on pre-defined fuzzy membership functions. The BFRMC classification algorithm automatically generates low dimensional fuzzy rules and corresponding membership functions directly from the training dataset, based on the fuzzy lower and upper approximations.

6.3 Boosting Based Fuzzy-Rough Multi-cluster Classifier

In this section, the BFRMC classifier algorithm based on fuzzy lower and upper approximations is described. The fuzzy discretization of the feature space is done using the cluster centers identified by the subtractive clustering technique [2]. The lower and upper approximations of the fuzzy sets are identified and the classification rules are generated. A set of *certain rules* is generated from the fuzzy lower approximations. Another set of *possible rules* is generated from the fuzzy upper approximations by a boosting enhanced genetic algorithm, for those training samples which are not classified by the certain rule set. The various stages of the algorithm are explained in following subsections.

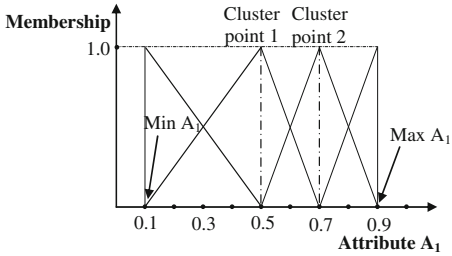


Fig. 6.1 Formation of membership functions from cluster center points

6.3.1 Stage 1: Membership Functions from Cluster Points

The fuzzy membership functions are created using feature cluster centers, and are utilized for the fuzzy partitioning of the data. Subtractive clustering technique [2] is utilized to identify the feature cluster center points. The cluster centers are determined for each feature of a class. The peak of a triangular membership function is placed at the cluster center with its left and right sides spanning to the adjacent left and right cluster centers. The minimum (maximum) feature value positions the left (right) side of the leftmost (rightmost) membership function. For example, if a set of cluster points $cp_1 = \{0.5, 0.7\}$ are obtained for an attribute A_1, which is within the range [0.1, 0.9], then the triangular membership functions for A_1 are as shown in Fig. 6.1.

A set of membership functions $\{MF_{A_i}\}$ is obtained for each attribute of a class. The set of membership functions acts as a base for the crisp sets generated in Stage 2 and for the descriptive fuzzy if-then rules generated in Stage 3.

6.3.2 Stage 2: Generation of Certain Rules from Membership Functions

The set of membership functions $\{MF_{A_i}\}$ obtained in Stage 1, serves as the starting point for defining a crisp set of *certain* regions $\{MF_L\}$. The crisp sets are formed at data clusters to maximize the fuzzy lower limit approximations as shown in Fig. 6.2.

Figure 6.2b plots a sample distribution of data for a two attribute classification case. Figure 6.2a maps the data distribution on attribute A_1 to one of the triangular membership functions $\{MF_{A_1}\}$, obtained in Stage 1. The consequent class c_{lj} of a rule is determined by identifying the dominating class, C_{max}, among training samples for the attribute A_1 (6.3).

$$C_{max}(x^k_{A_i}) = \text{argmax}_{C_m} \sum_{R_{lj}/c_{lj}=C_m} \mu_{R_{lj}}(x^k_{A_i}) \tag{6.3}$$

The maximum membership value μ_u of a sample belonging to other classes (other than C_{max}) is evaluated as per (6.4).

$$\mu_u = \{\max_{k|c^k \neq c_{lj}} (\mu_{A_i}(x^k_{A_i}))\} \tag{6.4}$$

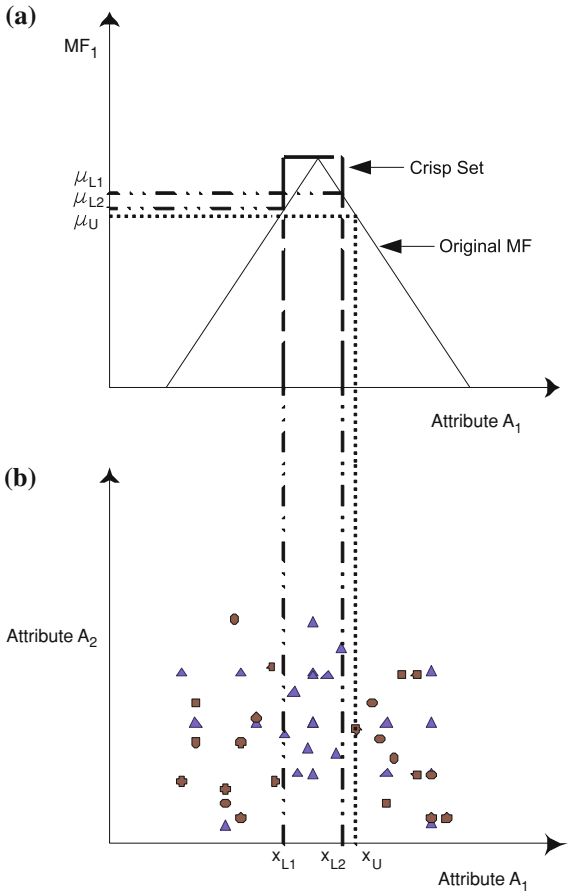


Fig. 6.2 Obtaining crisp sets from cluster center points

All the samples which have memberships greater than μ_u can be conclusively identified as belonging to class c_{lj}. The attribute values for memberships larger than μ_u are identified and are sorted. The smallest A_1 value, x_{L1}, and the largest A_1 value, x_{L2}, are identified. These values serve as the limits of a crisp set as in Fig. 6.2. Only samples belonging to class c_{lj} attain a membership of 1, while samples from other classes attain a membership of 0. Each crisp set serves as a singleton rule. The set of all such *certain* regions forms the certain rule base. Samples attaining unity membership form the fuzzy lower approximation for the partition induced by class c_{lj}.

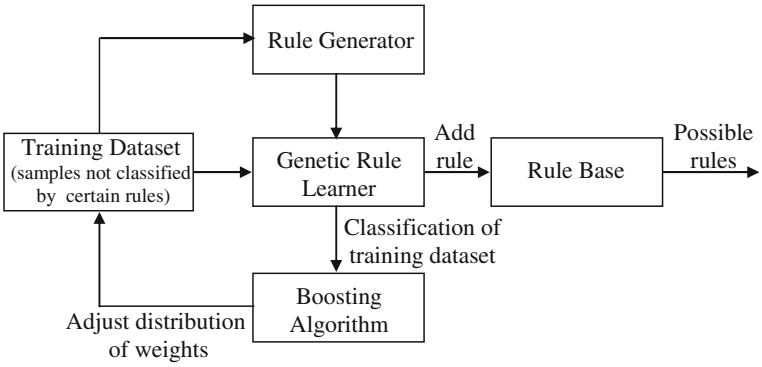


Fig. 6.3 Generation of possible rules

6.3.3 Stage 3: Generation of Possible Rules from Membership Functions

The training samples which are not classified by the certain rule set in Stage 2, serve as the training dataset for Stage 3. In this stage, possible rules are generated based on fuzzy upper approximations derived from $\{MF_{A_i}\}$ through a boosting enhanced GA.

6.3.3.1 Boosting Enhanced Genetic Algorithm for Learning Fuzzy if-then Rules

The architecture of the Iterative Rule Learning (IRL) [15, 16] based genetic algorithm is shown in Fig. 6.3. The fuzzy rule chromosome population is created by a rule generator, from the set of membership functions $\{MF_{A_i}\}$ obtained in Stage 1. The genetic rule learner selects the best rule from the population based on fitness, after employing the *crossover* and *mutation* genetic operators. The generated rule is added to the rule base and the weights of the training samples are adjusted by the boosting algorithm for the current set of rules. These steps are iterated till the error rate converges, or the maximum number of iterations is reached. Each of these steps are explained in the following subsections.

6.3.3.2 Rule Generation and Encoding of Fuzzy if-then Rules

Every chromosome in the population encodes a single fuzzy rule. Each attribute, $[A_i, i = 1, \ldots, n]^T$, is represented by a membership function. The membership function for each attribute, A_i, is chosen at random from the set $\{MF_{A_i}\}$ obtained in Stage 1. Triangular membership functions in the set $\{MF_{A_i}\}$ are assigned with integer labels. The triangular functions for attributes are encoded in chromosomes by

the integer labels. The rule is represented by an integer valued vector $a_1, a_2, \ldots, a_n$ in the chromosome.

A specific fuzzy rule may involve most of the attributes. General fuzzy rules involve only a few attributes. For the generation of general rules, an additional bit string $S = \{s_1, \ldots, s_n\}$ indicating the presence or absence of attributes $(A_1, \ldots, A_n)$ is included in a chromosome.

When a fuzzy rule is activated (attributes attaining a membership value greater than zero), the degree of support for the rule is calculated by applying the fuzzy operator *min* on membership functions of the antecedent attributes (6.5).

$$\mu_{R_j}(x^k) = \mu_{R_j}(\{x_1^k, \ldots, x_n^k\}) = \min_{n=1}^{N} \mu_{A_{jn}}(x_n^k) \tag{6.5}$$

where,

R_j	Label of the jth fuzzy if-then rule
$A_{j1}, \ldots, A_{jn}$	Antecedent fuzzy sets corresponding to attributes

Through the aggregation of antecedent attribute memberships, the degree of activation of a fuzzy rule R_j is determined. Sample x^k is classified according to the class label $C_{max}(x^k)$ by (6.6).

$$C_{max}(x^k) = \text{argmax}_{C_m} \sum_{R_j/c_j=C_m} \mu_{R_j}(x^k) \tag{6.6}$$

The GA iterations are fast due to two reasons. The first reason is that the training set has become comparatively smaller after Stage 2. The second reason is that the chromosome encodes only the integer labels of the membership functions, rather than real valued attributes of the triangular membership functions. This reduces the chromosome length to one-third of a real valued chromosome. Reduction in chromosome length leads to considerable increase in iteration speed especially for datasets with higher dimensionality.

6.3.3.3 Fitness Criteria

The genetic rule learner evaluates each rule based on a fitness criteria. The fitness criteria is designed for optimizing four objectives. Each objective is evaluated individually. All the objectives are aggregated into a single scalar fitness value in the range [0, 1]. The weights w_k is assigned to individual samples by the boosting algorithm, based on the relative difficulty in classifying a sample. The weight w_k is a part of the fitness criteria.

The first objective ensures that new rules are learned from positive samples with larger weights as compared to negative samples. This is achieved by a weighted plausibility factor f_{j_1} defined as,

$$f_{j_1} = \frac{\sum\limits_{k|c^k=c_j} w_k \mu_{R_j}(x_k)}{\sum\limits_{k} w_k \mu_{R_j}(x_k)} \tag{6.7}$$

At later stages of iteration, the weights of unclassified samples tend to be high. To ensure that each rule is generic rather than relying only on large weighted samples at later stages, a plausibility factor (6.8) is utilized as the second objective f_{j_2}.

$$f_{j_2} = \frac{\sum\limits_{k|c^k=c_j} \mu_{R_j}(x_k)}{\sum\limits_{k} \mu_{R_j}(x_k)} \tag{6.8}$$

The first two objectives indicate the relative aggregate membership attained by the samples in class c_j. If there are large number of samples from class c_j as compared to other classes in the training set, the objectives f_{j_1} and f_{j_2} do not ensure that the memberships attained by samples in class c_j are large. The third objective f_{j_3} maximizes the memberships attained by samples in class c_j.

$$f_{j_3} = \frac{\sum\limits_{k|c^k=c_j} \mu_{R_j}(x_k)}{l = \left|\mu_{R_j}(x_k) > 0 \forall k|c^k = c_j\right|} \tag{6.9}$$

The final objective f_{j_4} is to maximize the number of positive samples l_1 utilized for learning, as compared to the number of negative samples l_2. Fitness values are normalized by the number of samples l_3 from class c_j.

$$\begin{aligned} l_1 &= \left|\mu_{R_j}(x_k) > 0 \;\forall\, k|c^k = c_j\right| \\ l_2 &= \left|\mu_{R_j}(x_k) > 0 \;\forall\, k|c^k \neq c_j\right| \\ l_3 &= \left|k|c^k = c_j\right| \\ f_{j_4} &= \begin{cases} 0 : l1 < l2 \\ \dfrac{l1 - l2}{l3} \end{cases} \end{aligned} \tag{6.10}$$

The aggregate fitness f_j of the rule is computed as,

$$f_j = \prod_{1 \le i \le 4} f_{j_i} \tag{6.11}$$

Table 6.1 Test datasets for BFRMC classifier

Dataset	# Classes	# Features	# Samples
Iris	3	4	150
Wine	3	13	178
Glass	6	9	214

6.3.3.4 Iterative Boosting and Assignment of Weights

In Iterative Rule Learning (IRL), the classified samples are penalized. In the BFRMC algorithm, the training samples are assigned with weights, which are reduced upon classification, to penalize the classified samples. All the training samples are assigned with uniform weights $w^k = 1$ at the beginning. After each iteration, the weights of all samples successfully classified by the selected rule are reduced by a fixed quantity $d = 0.2$. The weights of all the misclassified samples are increased by the same fixed quantity $d = 0.2$ (6.12). The weights of unclassified samples remain unchanged. Such a weight adaptation ensures that the classified samples have lesser weights than the misclassified and unclassified samples. The focus of the subsequent generation of rules is on the misclassified and unclassified samples.

$$w^k(t+1) = \begin{cases} 0 & \text{if classified and } w^k(t) - d < 0, \\ w^k(t) - d & \text{if classified by selected rule,} \\ w^k(t) + d & \text{if misclassified by selected rule} \end{cases} \tag{6.12}$$

Though the algorithm is designed to focus on misclassified samples, some of the generated best rules classify samples already classified by existing rules. New rules generated are not added to the rule base, if they do not classify at least a few misclassified or unclassified samples. The existing sample weights are not changed, if a new rule is not added to the rule base.

Once the final rule set is generated, classification of a new sample is carried out using a single winner rule. If more than one rule is activated, winner is decided as the one which has the largest aggregate membership (6.5). This allows for the intuitive interpretation of the fuzzy rule base.

6.4 Performance Evaluation and Discussion

The performance of the BFRMC classifier is evaluated on three datasets, Iris, Wine, and Glass, from the UCI machine learning repository [17]. The details of the datasets are provided in Table 6.1. Ten fold cross validation is carried out for all the datasets and average accuracies are reported.

A crossover rate of 0.9 and a mutation rate of 0.1 are used for the GA. The GA is allowed to run for a maximum of 50 generations before selecting the best rule. The number of newly created chromosomes in each generation is limited to 90% of the population. In each generation, a new population is formed based on the fitness

Table 6.2 BFRMC classification results for a ten fold cross test

Dataset	Avg. # certain rules	Avg. # possible rules	Avg. Accuracy (%)	Best Accuracy (%)
Iris	10	6	96.7	98.4
Wine	11	10	98.3	99.5
Glass	3	17	61.7	63.4

Table 6.3 BFRMC classifier—Performance comparison

Dataset	Algorithm	Avg. Accuracy %
Iris	BFRMC classifier	**98.7**
	C4.5 [18]	93.7
	Naive Bayes	95.5
	Pittsburgh Approach [19]	98.5
Wine	BFRMC classifier	**100**
	Ishibuchi without *CF* modification [20]	98.5
	Ishibuchi with *CF* modification [20]	100
	Corcoran [21]	99.5
Glass	BFRMC classifier	**63.1**
	Ishibuchi [20]	64.4
	1R [22]	53.8
	C4 [23]	63.2

value of the combined parent and child pool. The rule generation continues until all samples are classified. The error rate started converging after the generation of 20 to 25 rules. The rule generation is then stopped, though there are misclassified samples.

Classification results of the ten fold cross tests are provided in Table 6.2. Table 6.3 provides a comparison of the performance of the BFRMC classifier with relevant classification algorithms reported in literature. Experiments are carried out with the same sample divisions (training and testing sets) as in the compared works. The BFRMC classifier provided equivalent or better accuracies for all the datasets considered.

Majority of samples from different classes are well separated or formed distinctive data clusters in the case of Iris and Wine datasets. In these cases more number of certain rules are generated and the classification is performed mainly by the certain rules. When the data clusters are very narrow as in the case of glass dataset (which lead to lesser classification accuracy), only a few certain rules are generated. In this case, the classification is performed mainly by the possible rule set. The classifier is able to handle both well separated and inter-mixed data, with a trade-off between the number of certain and possible rules.

6.5 Conclusion

The BFRMC classifier based on the concept of fuzzy equivalence relation is presented. A certain rule set is generated based on the fuzzy lower approximation. A possible rule set based on fuzzy upper approximation is generated by a boosting

enhanced genetic algorithm. Both certain and possible rule sets are generated automatically from the training dataset. The performance of the BFRMC classifier is evaluated with some well known datasets and compared with relevant classification techniques. The classification results show the good discriminative power of the BFRMC classifier.

Acknowledgments Figures and tables in this chapter are adapted from 'Boosting based Fuzzy-Rough Pattern Classifier', Prahlad Vadakkepat, Pramod Kumar Pisharady, Sivakumar Ganesan, and Loh Ai Poh, Trends in Intelligent Robotics, Springer, Book Series: Communications in Computer and Information Science, Vol.103, Page Nos. 306-313, Copyright @ 2010, with permission from Springer Science+Business Media.

References

1. L.A. Zadeh, Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
2. S. Chiu, Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Sys. **2**(3), 18–28 (1994)
3. Z. Pawlak, Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
4. Z. Pawlak, Rough classification. Int. J. Man Mach. Stud. **20**, 469–483 (1984)
5. D. Dubois, H. Prade, in *Putting rough sets and fuzzy sets together*, ed. by R. Slowinski, Intelligent Decision, Support: Handbook of Applications and Advances in Rough Sets Theory, Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 11, (Kluwer Academic Publishers, Dordrecht, 1992), pp. 203–232
6. D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets. Int. J. Gen. Syst. **17**, 191–209 (1990)
7. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **07**(03), 331–356 (2010)
8. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets. Appl. Soft. Comput. **11**(04), 3429–3440 (2011)
9. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition, PhD Thesis, National University of Singapore 2011.
10. P.K Pisharady, P. Vadakkepat, S. Ganesan, A.P. Loh, Boosting based fuzzy-rough pattern classifier, in Proceedings of Trends in Intelligent Robotics: 15th Robot World Cup and Congress, FIRA 2010, Bangalore, India, vol. 103, pp. 306–313, 15–19 Sept 2010.
11. T.P. Hong, T.T. Wang, S.L. Wang, Knowledge acquisition from quantitative data using the rough-set theory. Intell. Data Anal. **4**, 289–304 (2000)
12. T.P. Hong, T.T. Wang, S.L. Wang, B.C. Chien, Learning a coverage set of maximally general fuzzy rules by rough sets. Expert Syst. Appl. **19**, 97–103 (2000)
13. Qiang Shen, Alexios Chouchoulas, A rough-fuzzy approach for generating classification rules. Pattern Recogn. **35**, 2425–2438 (2002)
14. A. Roy, K.P. Sankar, Fuzzy discretization of feature space for a rough set classifier. Pattern Recogn. Lett. **24**, 895–902 (2003)
15. A. Gonzalez, F. Herrera, Multi-stage genetic fuzzy systems based on the iterative rule learning approach. Mathware Soft. Comput. **4**, 233–249 (1997)
16. A. Gonzalez, R. Perez, Slave: a genetic learning system based on an iterative approach. IEEE Trans. Fuzzy Syst. **7**(2), 176–191 (1999)
17. C.L. Blake, C.J. Merz, Uci repository of machine learning databases (1998), http://www.ics.uci.edu/ mlearn/MLRepository.html
18. J.R. Quinlan, *Neuro-fuzzy Soft computing* (Morgan Kaufmann, San Mateo, 1993)

19. Y. Shi, R. Eberhart, Y. Chen, Implementation of evolutionary fuzzy systems. IEEE Trans. Fuzzy Syst. **7**(2), 109–119 (1999)
20. H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems. IEEE Trans. Syst. Man Cybern. Part B. Cybern. 29(5), 601–618 (1999).
21. A.L. Corcoran, S. Sen, Using real-valued genetic algorithms to evolve rule sets for classification, in Proceedings of first IEEE International Conference on Evolutionary Computing (Orlando, FL), pp. 120–1241, June, 1994.
22. R.C. Holte, Very simple classification rules perform well on most commonly used dataset. Mach. Learn. **11**(1), 63–90 (1993)
23. J.R. Quinlan, Simplifying decision trees. Int. J. Man Mach. Stud. **27**, 221–234 (1987)

Part III
Biologically Inspired Approaches in Hand Posture Recognition

Chapter 7
Hand Posture Recognition Using Neuro-Biologically Inspired Features

If the human brain were so simple that we could understand it, we would be so simple that we couldn't

Emerson M. Pugh

Abstract This chapter presents two algorithms for hand posture recognition, based on the computational model of the ventral stream of visual cortex. In the first algorithm, the hand postures are recognized by elastic graph matching between a model graph and patterns in an input image. Graph nodes are assigned to geometrically significant positions in a hand image and each node in the graph is labeled using an image feature extracted using the computational model of the ventral stream. A radial basis function is utilized as the similarity function for the matching process. In the second algorithm, shape based features which are good in interclass discrimination are identified. To classify the patterns, similarities of the shapes are compared. The second algorithm needs only single training image per class. The overall algorithm is computationally efficient due to the simplicity of the classification process. The algorithm is implemented in real-time for interaction between a human and a virtual character *Handy*. *Handy* can symbolically express the recognized hand posture and pronounce the relevant class name. The experimental results show that the algorithm is robust to hand posture size and shape variations and varying lighting conditions.

Keywords Hand posture recognition · Elastic graph matching · Human computer interaction · Biologically inspired features

7.1 Introduction

Feature selection and classification algorithms for pattern classification are discussed in Chaps. 4 and 5. Image features are extracted using a standard visual cortex model [9]. Visual object recognition is mediated by a ventral visual object processing stream in visual cortex [2, 8]. The computational model of the ventral stream [7, 9, 10] is

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_7,
© Springer Science+Business Media Singapore 2014

based on the standard model of visual object recognition [8]. The model provides a hierarchical system with four layers imitating the feed-forward object recognition path in the ventral stream of primate visual cortex (Sect. 2.2.1). This chapter discusses hand posture recognition systems with a modified feature extraction algorithm.

7.2 Graph Matching Based Hand Posture Recognition: C1 Features

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

In a typical graph representation, regions of an image are represented by graph nodes. The nodes are related to each other by edges, which express the structural relationships among regions. The nodes of a graph are labeled with a local image description, and 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 [12, 13]. EGM is a neurally inspired object recognition technique [3], 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 generated from training images are matched with the input image which contains the pattern to be recognized. Elastic matching of a model graph with an image data graph searches 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 [12].

The EGM algorithm utilizes a set of image features [4, 6] which are extracted using the computational model of the ventral stream of visual cortex [9, 10]. The image features are utilized in labelling various nodes in the graph. The model graph is created by assigning nodes to geometrically significant positions in a hand image. Each hand posture is classified by identifying the best match between the corresponding input image and model graphs. A radial basis function (RBF) is used as the similarity measure for comparing the node features. The performance of the algorithm is analysed with a ten class hand posture database.

7.2.1 Elastic Graph Matching Based Hand Posture Classifier

The algorithm utilizes graph node geometry to capture hand posture structure. Each node is labeled using Gabor filter based features (C_1 features). Gabor filters are edge detecting filters. Gabor filter responses of images, along various filter orientations,

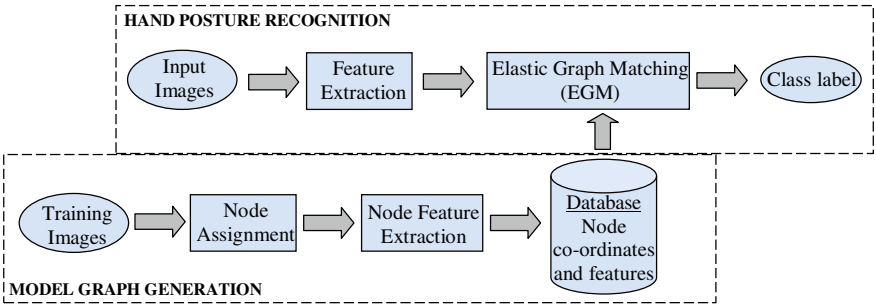


Fig. 7.1 Elastic graph matching based hand posture recognition algorithm

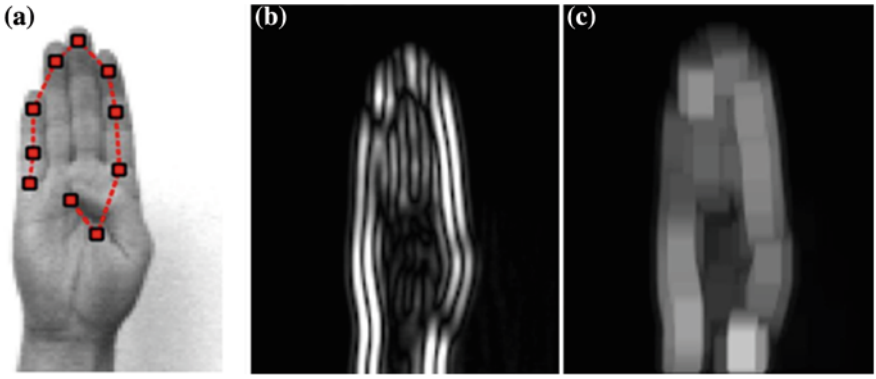


Fig. 7.2 **a** Graph node positions in a sample hand image, **b** S_1 response (orientation 90°), **c** C_1 response (orientation 90°)

capture hand shapes. The node features are expressed using C_1 image patches containing local shape information. A group of such nodes stores complex hand posture shapes.

Figure 7.1 shows a simplified block diagram of the overall algorithm. Model graphs are created from the C_1 responses of the training images. Model graph nodes are assigned to geometrically significant positions in hand images (Fig. 7.2a). C_1 images (Fig. 7.2b) of training images are extracted and are sub-sampled (1 out of 5 pixels) to reduce the computational burden. The sub-sampled C_1 image patches centered at graph nodes are extracted. The patch size used is 15×15 pixels. Each patch contains four orientations (0°, 45°, 90°, −45°). These image patches are used as model graph node features.

In [9] RBFs are used in S_2 stage to compare complex C_1 image (corresponding to an input image) with patches of previously seen visual patterns. RBFs compare the distance between an input and a prototype [1]. The EGM based algorithm utilizes RBF as a similarity function for matching the graphs. Similarity of a node is dependent on the Euclidean distance between node features of data and model graphs, in a Gaussian-like way. The shorter the Euclidean distance, bigger is the similarity.

The algorithm utilizes several images of the same class (four images per class) to create a bunch graph ([14]). Bunch graphs model the variability in object appearance. In a bunch graph, each model graph node is represented using a bunch of node features. Bunch of features is extracted from identical nodes (nodes assigned to identical positions) of several images, which vary in posture sizes and shapes. During the recognition process, all features from a bunch are matched, and the maximum of the similarities is considered for decision making.

To recognize a hand posture, model graphs are sequentially matched to the input image as described below.

1. C_1 image of input image is extracted.
2. C_1 image is sub-sampled (1 out of 5 pixels).
3. Sub-sampled image is scanned in x and y directions using model graphs.
4. Similarity between each graph node i is calculated using the equation

$$s_{ij} = \exp(-||X_i - P_{ij}||^2), \tag{7.1}$$

   where X_i is the feature (C_1 image patch) of the ith input image node and P_{ij} is the feature[1] of the ith node of jth class (model graph). Each node is allowed to shift its position by two pixels in x and y directions (two pixels to the left, right, above and below). This is to take care of small distortions in a graph.
5. Similarity S_j between two graphs is calculated by summing up node similarities (7.2).

$$S_j = \sum_{i=1}^{n} s_{ij}, \tag{7.2}$$

   where n is the number of nodes in the model graph.
6. Class corresponding to the model graph which provides the best match is the output class C (the class to which the posture belongs to) (7.5).

$$C = \text{argmax}_j\{S_j\} \tag{7.3}$$

   Figure 7.3 shows flowcharts of the model graph generation and the hand posture recognition algorithm.

7.2.2 Experimental Results

The EGM algorithm is tested on Jochen Triesch hand posture dataset [11]. The dataset consists of ten classes of hand postures, performed by 24 subjects, against light and dark backgrounds (Fig. 7.4). The images are in 8-bit grey-scale with a resolution of

[1] The feature, among a bunch of node features, which provides maximum similarity.

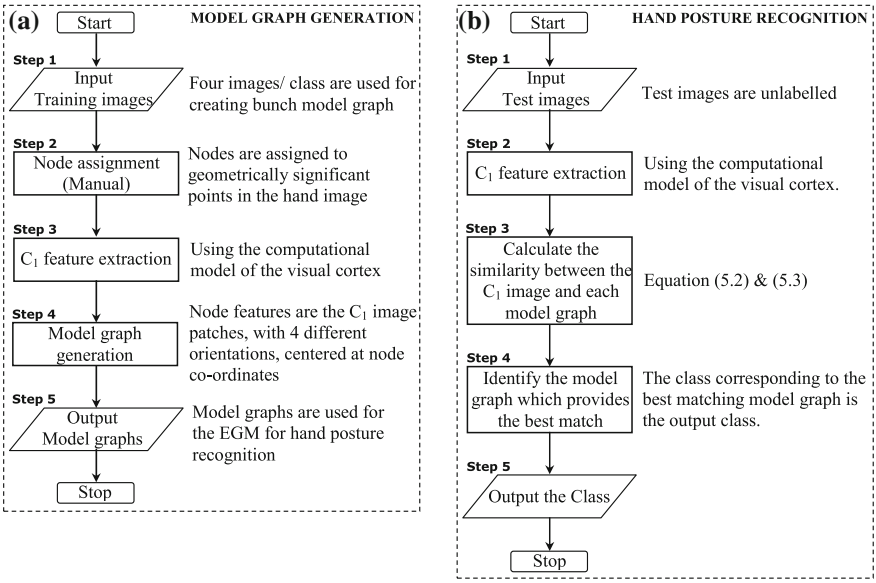


Fig. 7.3 Flowcharts of **a** the model graph generation and **b** the hand posture recognition algorithm

128×128 pixels. The repository has 240 images with light background and 238 with dark background. The images vary in size and shape of the hand postures.

Hand postures of two subjects against light and dark backgrounds (40 images) are utilized in generating the model graphs. The algorithm is tested using the remaining 438 images. Out of the 220 test images with light background (which are not used in model graph generation) 213 were recognized correctly, which corresponds to a recognition accuracy of 96.82 %. Recognition rate for dark background images is 209 out of 218 (95.87 %). Misclassification were mainly due to the close similarity between postures 5 and 9 (Fig. 7.4). The average accuracy provided by the algorithm is 96.35 % (Table 7.1), whereas the earlier reported accuracy [13] for the same dataset is 93.77 %. In [13], 60 images (six images per class), each with 35 nodes, are used for model graph generation, whereas in the presented algorithm model graphs are generated using 40 images (four images per class), each with ten nodes. The reduction in number of nodes reduces the computational burden of the recognition algorithm.

The total computational time needed for the recognition process is divided into two, time for feature extraction and classification. The feature extraction takes 1.9 s (on average) to complete. The classifier, which is based on elastic graph matching, has a high computational complexity. The matching algorithm utilizes only ten nodes per model graph leading to lesser classification time. The time taken for classification is 12 ms (on average), on a computer with 3 GHz Intel Pentium 4 CPU.

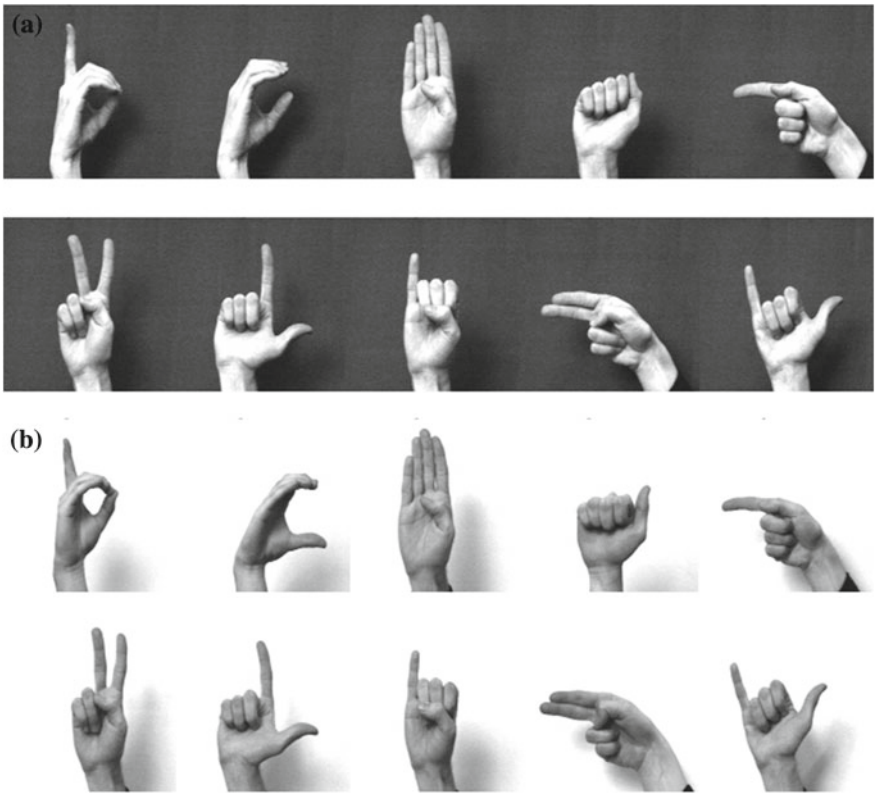


Fig. 7.4 Sample hand posture images with **a** dark and **b** light backgrounds, from Jochen Triesch hand posture dataset [11]

Table 7.1 Comparison of recognition accuracy

Background	# Test images	# Correct classifications	EGM algorithm	Reported in [13]
Dark	218	209	95.87	93.3
Light	220	213	96.82	94.3
Net	438	422	96.35	93.77

7.2.3 Summary: Graph Matching Based Hand Posture Recognition Algorithm

The elastic graph matching based hand posture recognition algorithm is discussed. The nodes of the graph are labeled using the C_1 Standard Model Features (SMFs), which are extracted using the computational model of the ventral stream of visual cortex. Shapes of hand postures are recognized by identifying the structure of distribution of nodes in input images. Graphs are matched using a radial basis function as

similarity function. Variations in size and shape of hand postures are taken care of by local maximization in the C_1 stage and by using the bunch graph approach. The algorithm is tested on a ten class hand posture dataset. The algorithm which is scale tolerant and person independent, provided better recognition accuracy compared to the earlier reported results.

7.3 Feature Extraction and Selection for Hand Posture Recognition: C2 Features

The computational model proposed by Serre et al. extracts scale and position tolerant C_2 Standard Model Features (SMFs). A major limitation of this model is its processing speed in real-world applications. The algorithm needs a separate classifier for classification. A modified computationally efficient algorithm is discussed in this section, which does not require a separate classifier stage [4, 5]. The prototype C_1 patches used in learning stage are selected from specific classes. Patches which have good interclass differences are identified and the discriminative C_2 features are extracted. Classification is done by comparing the extracted C_2 features.

7.3.1 Discriminative Feature Extraction and Selection

Serre et al. have suggested C_2 SMFs for robust object recognition. The C_2 SMFs are learned from a set of randomly selected natural images. The *object specific* features (features learned from images containing the target object) perform better compared to the features learned from random images. The algorithm presented in this section extracts features by considering the class division. The C_2 features are made class specific by modifying the third layer of feature extraction process. Instead of a random selection, the patches are selected from geometrically significant positions of specific hand posture classes. In [9] support vector machines (SVM) and boosting based classifiers are used for the classification, whereas the discussed algorithm classifies hand shapes by a comparison of extracted features. Patches with higher inter-class differences are identified and selected, leading to better classification accuracy and reduced computational cost (due to reduced number of features). Figure 7.5 shows a block diagram description of the discussed algorithm. The different blocks are explained in following subsections.

7.3.1.1 Assigning Prototype Patches Specific to Posture Classes

C_1 features of training images (single image per class) are extracted and prototype patches (center of the RBF units) of different sizes are assigned to class specific

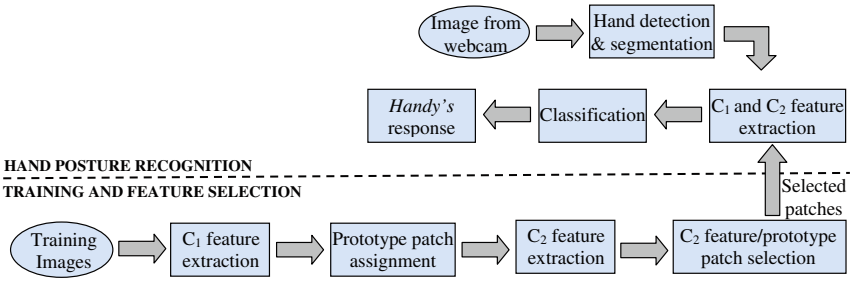


Fig. 7.5 C_2 features based hand posture recognition algorithm

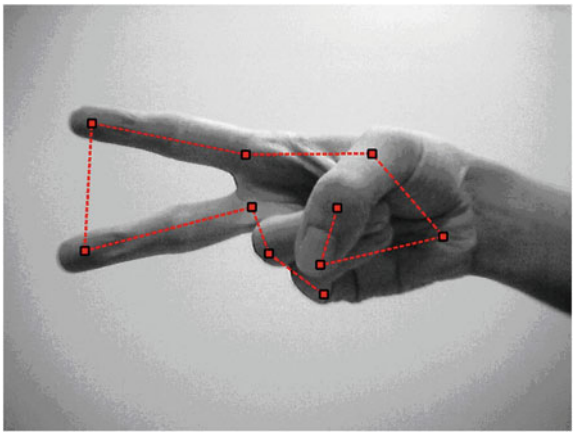


Fig. 7.6 Sample hand image depicting positions of prototype patches

C_1 images. N (=10) patches, each with Q (=4) different patch sizes are extracted from M (=10) classes. The patches are placed at geometrically significant positions in a hand image (Fig. 7.6). Each patch contains C_1 responses with O (=4) different orientations. C_2 features are extracted from C_1 images using prototype patches. An extracted C_2 feature corresponds to a particular patch with specific size. The total number of C_2 features is $Q \times M \times N$.

7.3.1.2 Classification

Once the C_2 features are extracted, patterns are classified as per (7.5).

Let,

$$P_{ij} = \sum_{k=1}^{Q} C_2(ijk), \tag{7.4}$$

where $C_2(ijk)$ is the C_2 feature corresponding to the jth patch from the ith class with kth patch size, then,

$$C = \text{argmax}_{C_i}\{Q_{C_i}\}, \tag{7.5}$$

where,

$$Q_{C_i} = \sum_{j=1}^{N} P_{ij}, \tag{7.6}$$

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

7.3.1.3 Selection of Discriminative C_2 Features

The C_2 features are selected based on interclass differences. Each C_2 feature corresponds to a particular prototype patch. Prototype patches which can distinguish different classes are identified and selected to make the extracted C_2 features discriminative.

Let,

$$P'_{ij} = \max_{C_l \neq C}\{P_{ij}(C_l)\}, \tag{7.7}$$

where, $P_{ij}(C_l)$ is the response P_{ij} (7.4) of an image from class C_l, then $P_{ij} - P'_{ij}$ represents the *margin of classification (MC)*. The prototype patches from each class are sorted in descending order with respect to the corresponding *MCs* and the first N' (=5) patches are selected.

The features of a test image are extracted using the selected discriminative patches to obtain the discriminative C_2 features. The hand posture classification is done using (7.5) and (7.6) ($N = N'$ in 7.6).

7.3.2 Real-time Implementation and Experimental Results

The algorithm discussed is implemented in real-time for the interaction between human and a virtual character *Handy*. The programming is done using C# language. The virtual character is created using Adobe flash. Twelve different animations (ten of them representing different predefined classes and, one each representing a non-class and the idle mode) are developed. A text to speech converter is used to generate voice response. *Handy* can welcome a user and provide a command to connect a camera if it is not connected already. Once the camera is connected and a posture is shown, *Handy* can identify the posture and respond to human by symbolically showing that posture. Handy can also pronounce the class number to which the posture belongs to. If a posture does not belong to any of the trained classes, *Handy* will exclaim and ask *what is that?*

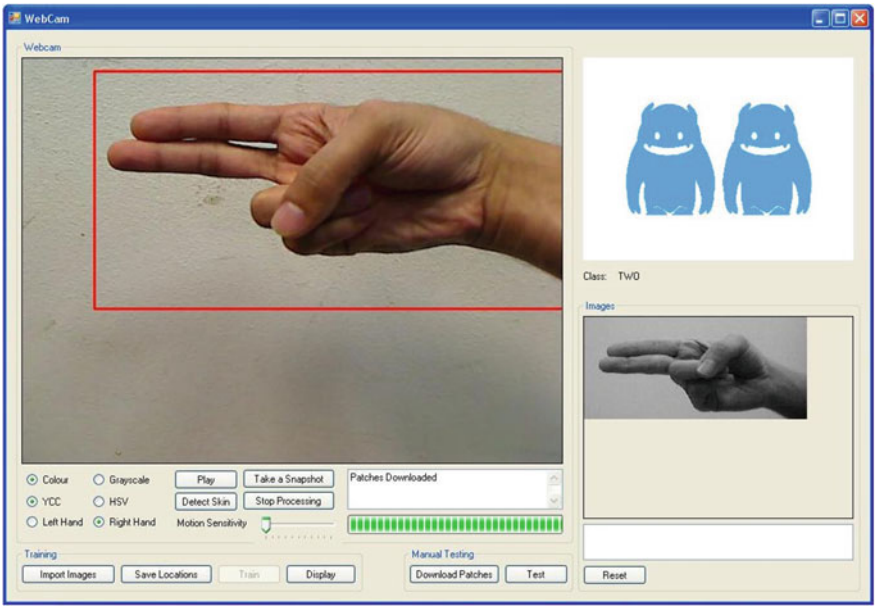


Fig. 7.7 User interface showing an image from webcam, the recognized posture and Handy's response

RGB image of hand posture is captured using a webcam. The image is converted to the YCbCr color space and the hand is detected using skin color segmentation.[2] The segmented image is converted to gray scale and used for further processing. Hand motion is detected by subtracting subsequent video frames. Image of the hand is captured only when the hand is stationary (when the detected pixel difference between two consecutive frames is less than a threshold value). Lower threshold value provides high motion sensitivity. Higher motion sensitivity leads to the hand being detected only when the motion is less.

The graphical user interface (Fig. 7.7) shows a sample hand posture segmentation, the recognized hand posture and *Handy*'s response. The interface has provisions for left/right hand selection, to vary the motion sensitivity, and to control various stages of the entire process.

The algorithm is trained for ten classes of hand postures (Fig. 7.8). Only one image per class is used for training and testing is carried out by showing each posture 20 times. The hand postures are performed by different persons, with variations 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 of the order of milliseconds.

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

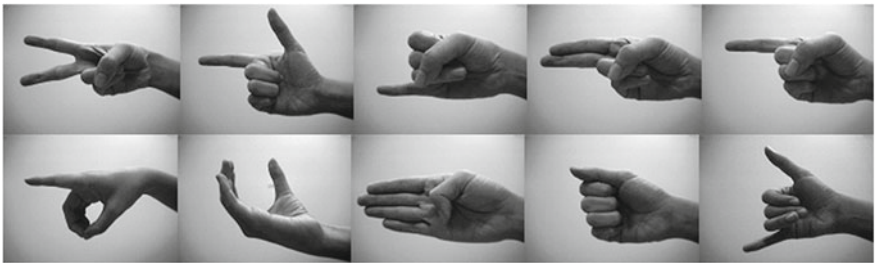


Fig. 7.8 Hand posture classes used in the experimentss

7.3.3 Summary: Feature Extraction and Selection for Hand Posture Recognition

A computationally efficient and reliable hand posture recognition algorithm for human computer interaction is discussed. The algorithm utilizes 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 they provide good discrimination among different classes, and classification is carried out by comparing the class features. The algorithm is implemented in real-time for interaction between a human and the virtual character *Handy*. The algorithm recognized hand postures with reliable accuracy, in a person independent manner.

The discussed algorithm easily recognized hand postures with light and dark backgrounds. The performance of the algorithm for complex background images needs improvement. A person independent hand posture recognition system against complex backgrounds is presented in Chap. 8.

Acknowledgments The figures and tables in Sect. 7.3 of this chapter are adapted from 'Hand posture recognition using neuro-biologically inspired features', Pramod Kumar Pisharady, Stephanie Quek Shu Hui, Prahlad Vadakkepat, and Loh Ai Poh, Trends in Intelligent Robotics, Springer, Book Series: Communications in Computer and Information Science, Vol.103, Page Nos. 290-297, Copyright @ 2010, with permission from Springer Science+Business Media.

References

1. C. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, 1995)
2. M. Goodale, A. Milner, Separate visual pathways for perception and action. Trends Neurosci. **15**, 20–25 (1992)
3. M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R.P. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture. IEEE Trans. Comput. **42**(3), 300–311 (1993)
4. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition, Ph.D. thesis, National University of Singapore, 2011

5. P.K. Pisharady, Q.S.H. Stephanie, P. Vadakkepat, A.P. Loh, Hand posture recognition using neuro-biologically inspired features, in *Trends in Intelligent Robotics: 15th Robot World Cup and Congress, FIRA 2010, Bangalore, India, September 15–19, 2010, Proceedings*, vol. 103, pp. 290–297
6. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Graph matching based hand posture recognition using neuro-biologically inspired features, in *International Conference on Control, Automation, Robotics and Vision (ICARCV) 2010 (Singapore)*, Dec 2010
7. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach, Int. J. Humanoid Rob. **07**(03), 331–356 (2010)
8. M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex. Nat. Neurosci. **2**(11), 1019–1025 (1999)
9. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
10. T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex, in *Conference on Computer Vision and Pattern Recognition*, pp. 994–1000, San Diego, 2005, ed. by C. Schmid, S. Soatto, C. Tomasi
11. J. Triesch, C. Malsburg, Sebastien marcel hand posture and gesture datasets: Jochen triesch static hand posture database (1996), http://www.idiap.ch/resource/gestures/
12. J. Triesch, C. Malsburg, A system for person-independent hand posture recognition against complex backgrounds. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1449–1453 (2001)
13. J. Triesch, C. Malsburg, Robust classification of hand postures against complex backgrounds, in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, (Killington, VT, USA), pp. 170–175, Oct 1996
14. L. Wiskott, J.M. Fellous, N. Kruger, C. Malsburg, Face recognition by elastic bunch graph matching. IEEE Trans. Pattern Anal Mach. Intell. **19**(7), 775–779 (1997)

Chapter 8
Attention Based Segmentation and Recognition Algorithm for Hand Postures Against Complex Backgrounds

Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains
Steve Jobs

Abstract The *Attention based Segmentation and Recognition* (ASR) algorithm for hand postures against complex backgrounds is discussed in this chapter. The ASR algorithm can detect, segment and recognize multi-class hand postures. Visual attention, which is a cognitive process of selectively concentrating on a region of interest in visual field, helps humans to recognize objects in cluttered natural scenes. The ASR algorithm utilizes a Bayesian model of visual attention 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 similarity map, using a computational model of the ventral stream of visual cortex. The skin similarity map, which represents the similarity of each pixel to the human skin color in HSI color space, enhances the edges and shapes within the skin colored regions. The color features used are discretized chrominance components in HSI, YCbCr color spaces, and similarity-to-skin map. The hand postures are classified using shape and texture features, with a support vector machines classifier. The NUS hand posture dataset-II with 10 classes of complex background hand postures is utilized for testing the algorithm. The dataset contains hand postures from 40 subjects of different ethnicities. A total of 2,750 hand postures and 2,000 background images are available in the dataset. The hand postures vary in size and shape. The ASR algorithm is tested for hand detection and hand posture recognition using 10 fold cross-validation. The experimental results show that the algorithm has a person independent performance, and is reliable against variations in hand sizes and complex backgrounds.

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8_8,
© Springer Science+Business Media Singapore 2014

Keywords Hand posture recognition · Complex background · Shape-texture features · Visual attention · Attention model · Saliency map · Biologically inspired features · ASR algorithm

8.1 Introduction

The mainstream computer vision research has always been challenged by human vision, and the mechanism of human visual system is yet to understand well. The human visual system is capable of rapidly and effortlessly recognizing large number of diverse objects in cluttered, natural scenes and identifies specific patterns. The capability of the human vision system has inspired development of computational models of biological vision systems. Intermediate and higher visual processes in primates select a subset of the available sensory information before further processing [28], in order to reduce the complexity in scene analysis. The selection is effected in the form of *focus of attention* [12]. Recent developments in neurobiological models in computer vision bridge the gap between neuroscience, computer vision and pattern recognition [5, 7, 8, 20–24].

Eng-Jon et al. proposed an unsupervised algorithm for hand detection and static hand shape recognition [13]. A tree of boosted (hand) detectors is reported with two layers; the top layer for hand detection and the branches in the bottom layer for hand shape classification. A shape context based distance metric is utilized for clustering similar looking hand shapes in order to construct a tree structure. The algorithm provided good detection and recognition accuracy.

A systematic approach to building a hand appearance detector is presented in [10]. The paper proposes a view specific hand posture detection algorithm based on the object recognition method proposed by Viola and Jones. A frequency analysis based method is utilized for instantaneous estimation of class separability, without the need for training. The algorithm is applied for detecting six hand postures. Wu et al. proposed an algorithm for view independent hand posture recognition [29]. A number of classification methods is investigated to make the algorithm view independent. The work combined supervised and unsupervised learning paradigms proposing a learning approach namely Discriminant-Expectation Maximization (D-EM). The D-EM uses an unlabeled dataset to help in supervised learning to reduce the number of labeled training samples. Image datasets with simple (uniform) and relatively similar backgrounds are utilized to test the algorithm. These works did not address the issues with complex backgrounds containing clutter and other distracting objects.

Recognizing hand postures in complex natural environments is useful for real-world applications of interactive systems. Triesch et al. [26, 27] addressed the complex background problem in hand posture recognition using elastic graph matching (EGM). *Bunch graph method* [27] is utilized to improve the performance in complex environments. In graph algorithms, entire image is scanned to detect

an object, which increases the computational burden.[1] In bunch graphs nodes are represented by bunches of identical node features which further decreases the processing speed. Athitsos et al. proposed an algorithm to estimate hand pose from cluttered images [1]. The algorithm segmented the image using skin color, and needs fairly accurate estimates of the hand size and center. The above algorithms cannot deal with complex backgrounds containing skin colored regions and large variations in hand size.

This chapter focuses on the detection and recognition of hand postures in cluttered natural environments. Attention based Segmentation and Recognition (ASR) algorithm is discussed. The ASR algorithm utilizes a biologically inspired approach, which is based on the computational model of visual cortex [23] and the Bayesian model of visual attention [5]. The Bayesian theory of attention is utilized to detect and identify the hand region in complex background images. The *where* information is extracted using feature based visual attention. The features utilized are based on shape, texture and color.

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 [17, 23]. The color features are extracted by discretizing chrominance color components in HSI and YCbCr color spaces, and the similarity-to-skin map. A saliency map is created by calculating posterior probabilities of pixel locations to be 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 segmenting the input image using the thresholded saliency map. Hand postures are recognized using shape and texture based features (of hand region), with a Support Vector Machines (SVM) classifier.

The ASR algorithm is robust against complex and skin colored backgrounds as hand regions are segmented using the attention mechanism. The attention mechanism utilizes a combination of color, shape and texture features. The experimental results show that the algorithm has a person independent performance. The algorithm is robust against variations in hand size and hand position in an image.

The number of hand posture databases available to the research community is limited [25]. A 10 class complex background hand posture dataset, namely NUS hand posture dataset-II[2] is utilized in this work. The images in the NUS hand posture dataset-II vary in hand size and are from 40 subjects of different ethnicities. The images have both indoor and 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 ASR algorithm. The recognition algorithm is tested with the dataset using a 10 fold cross validation strategy and provided an accuracy of 94.36 %.

[1] Graph matching is considered as one of the most complex algorithms in vision based object recognition [2]. The complexity is due to the combinatorial nature of matching process.

[2] The dataset is available for free download: http://www.vadakkepat.com/NUS-HandSet/.

Table 8.1 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 and 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 visual cortex. The simple cells primarily respond to oriented edges and bars. The complex cells provide spatial invariance

8.2 The Feature Extraction System and the Model of Attention

Biologically inspired shape and texture feature extraction system, and the Bayesian model of visual attention [5] are briefed in this section. The shape and texture based features are extracted using a cortex like mechanism [23]. The visual attention is implemented using the extracted shape and texture features, and a set of color features.

8.2.1 Extraction of Shape and Texture Features

The biologically inspired shape and texture feature extraction system consists of four layers (Table 8.1). 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 simple cells in primary visual cortex (V1[3]), detecting edges and bars. Layer 2 (C_1) models complex cells in V1, by applying a *MAX* operator locally (over different scales and positions) to the first layer's outputs.[4] The *MAX* operation provides tolerance to different object projection sizes, positions, and rotations in the 2-D plane of 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 training (in humans, these patches correspond to learned patterns of previously seen visual images and are stored in synaptic weights of neural cells). Finally, the fourth layer (C_2) applies a *MAX* operator (over all scales, but not over positions) to the outputs of layer S_2, resulting in a representation that expresses the similarities with 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.

Figure 8.1 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

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

[4] Reference [23] for further explanation on S_1 and C_1 stages (layer 1 and 2).

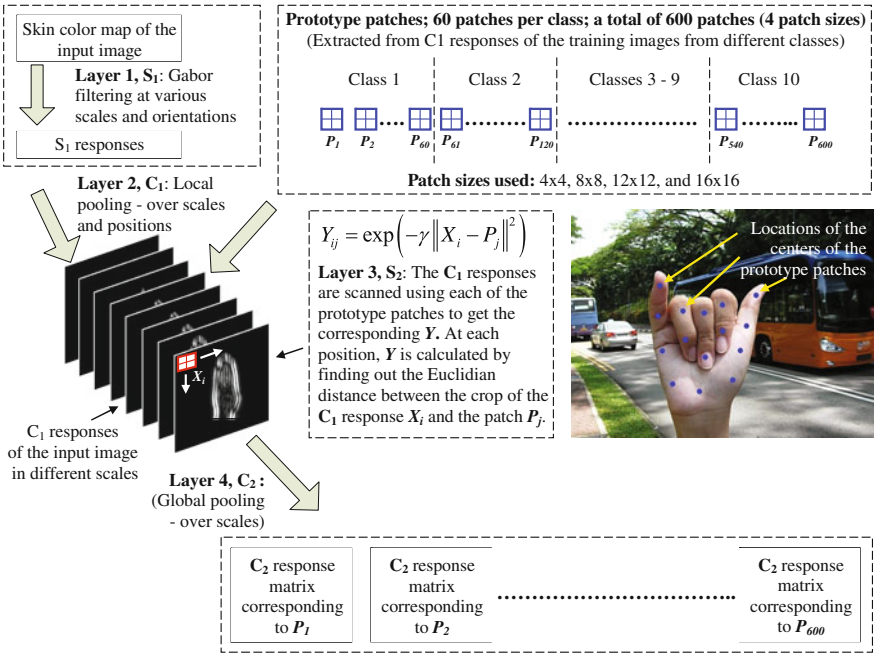


Fig. 8.1 Extraction of shape and texture based features (C_2 response matrices). The S_1 and C_1 responses are generated from a skin color map (Sect. 8.3.1.2) of the input image. The prototype patches of different sizes are extracted from C_1 responses of training images. Fifteen patches, each with four 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 geometrically significant and textured positions of 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 on the Euclidean distance between crops of the C_1 response (of the input image) and corresponding prototype patch, in a Gaussian-like manner

in the image to more complex shapes. RBF networks are a class of neural network model, comparing the distance between input and prototype signals [3]. Each S_2 unit response depends on the Euclidean distance between crops of the C_1 image (X_i) and the stored prototype patch (P_j), in a Gaussian-like manner. The prototype patches (*centers* of the RBF units) of different sizes are extracted from C_1 responses of training images. The patch centers are positioned at geometrically significant and textured positions of hand postures (Fig. 8.1). Each patch contains four orientations. The third layer compares the patches by calculating the summed Euclidean distance between a patch (P_j) and every possible crops (X_i) of a C_1 image. This comparison is done with all the C_1 responses in second layer (C_1 responses at different scales), combining all orientations.

The scale invariant C_2 responses are computed by taking a global maximum over all scales 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 ASR algorithm 15 prototype patches with 4 patch sizes are extracted from each of the 10 classes. The total number of patches is 600, leading to 600 shape and texture based features.

8.2.1.1 Modifications in the Shape and Texture Feature Extraction System

The shape-texture feature extraction in ASR algorithm has the following differences with the algorithm reported in [23].

1. Shape and texture features are extracted from a similarity-to-skin color map (Sect. 8.3.1.2) (not from grey scale image).
2. Prototype patches are extracted from different class images. The patch centers are placed at geometrically significant and textured positions of hand postures.
3. Output feature component is a C_2 response matrix (instead of a real number) which retains the hand location information.

The parameters of the shape-texture feature extraction system are adopted from [23] except for the number (600) and positions of prototype patches.

8.2.2 Bayesian Model of Visual Attention

Visual attention is the part of the inferencing process that addresses the visual recognition problem of *what is where* [5]. Visual attention helps to infer the identity and position of objects in a visual scene. Attention reduces the *size* of search space and *computational complexity* of recognition.

Visual attention is directed selectively to objects in a scene using both image-based saliency cues (bottom-up) and task-dependent cues (top-down). The top-down task based attention is more deliberate and powerful [7], and is depended on object features. The ASR algorithm is task based and utilizes a top-down approach. Attention is focussed on the region of interest using object features.

Visual perception is interpreted as a Bayesian inference process whereby top-down priors help to disambiguate noisy bottom-up sensory input signals [6]. Visual recognition corresponds to estimating posterior probabilities of visual features for specific object categories and their locations in an image. Posterior probabilities of location variables serve as a *saliency map*.

There are two types of visual attention, spatial attention and feature attention (Fig. 8.2). Spatial and feature based attentions are respectively utilized to reduce uncertainty in shape and spatial information. In spatial attention, different spatial priors are assigned by learning the context from training images (information about the position of an object in an image). In feature attention, different feature priors are assigned by counting the frequency of occurrence of each feature among training images.

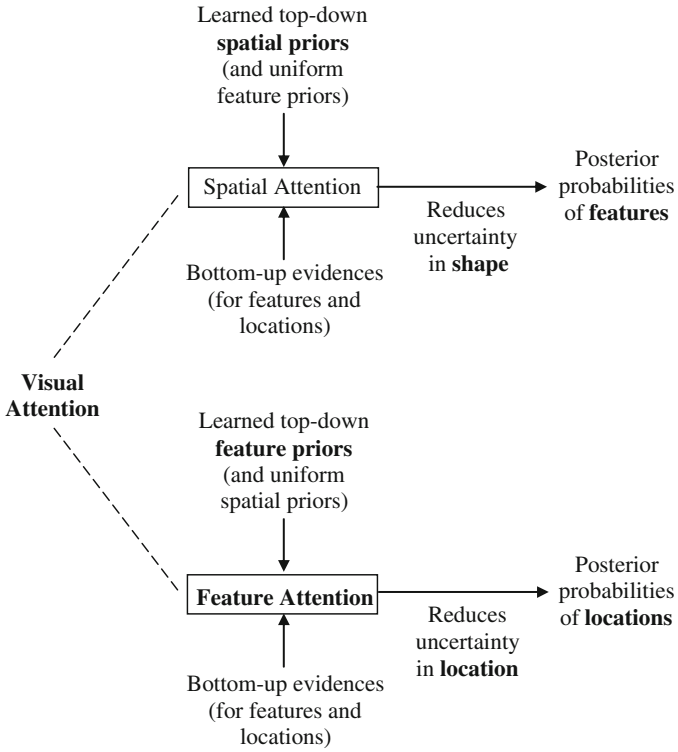


Fig. 8.2 Two types of visual attention as per the Bayesian model [5, 21]. Spatial attention utilizes different priors for locations and helps to focus attention on the location of interest. Spatial attention reduces uncertainty in shape. Feature attention utilizes different priors for features and helps to focus attention on the features of interest. Feature attention reduces uncertainty in location. The output of the feature detector (with location information) serves as *bottom-up* evidence in both spatial and feature attentions. Feature attention with uniform location priors is utilized in ASR hand posture recognition system. Hand position can be random in an image and every pixel has equal probability to be in a hand region

A Bayesian model of spatial attention is proposed in [21]. Chikkerur et al. [5] modified the model to include feature based attention, in addition to spatial attention. The model imitates the interactions between the parietal and ventral streams of visual cortex, using a Bayesian network (Bayes net). Bayes net represents joint probability distributions in a compact manner via conditional independence. It helps in calculating the posterior probability of an event, probability of the event given any evidence, through Bayesian inference. A biologically plausible belief propagation algorithm [14] is utilized in calculating the posterior probabilities.

The ASR algorithm utilizes feature based attention, with different feature priors, to create a saliency map. The location priors are set to be uniform, as the hand can be randomly positioned in the image. Visual attention model is developed utilizing a combination of shape, texture and color features. The saliency map (posterior

probabilities of pixel locations) is generated using the learned feature priors and evidence from test images. The hand region is segmented by thresholding the saliency map. Complex backgrounds in images can contain skin colored pixels and utilizing color based features alone is not effective for hand segmentation. Combining color features with shape and texture features results in better segmentation of hand region compared to that using shape and texture features alone.

8.3 Attention Based Detection, Segmentation and Recognition

The ASR algorithm addresses the complex image background issue by utilizing a combination of different features. The hand region in an image is identified by using the joint posterior probabilities of the combination of features. Bayesian inference is utilized to create a saliency map, which helps in segmenting 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 prior experience. In a complex background image, the shape patterns emerging from background affect the pattern recognition task negatively. To recognize a pattern from a complex background image, features corresponding to the foreground object are given higher weight compared to those corresponding to the background.

In ASR algorithm, the shape and texture features of hand postures, and the color features of 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 to the features corresponding to hand area. The priors are learned from training images by counting the frequency of occurrence of features. Hand postures are classified using the shape and texture features of the hand region with an SVM classifier.

Figure 8.3 shows the block diagram of the ASR hand posture recognition algorithm. The functions of different blocks are elaborated in following subsections.

8.3.1 Image Pre-processing

Image pre-processing includes color space conversions and generation of similarity-to-skin map.

8.3.1.1 Color Space Conversions: RGB to HSI and YCbCr

Input images in RGB space are converted to HSI and YCbCr color spaces. The conversion between RGB and HSI is non-linear whereas that between RGB and YCbCr is linear. The chrominance components in the HSI and YCbCr color

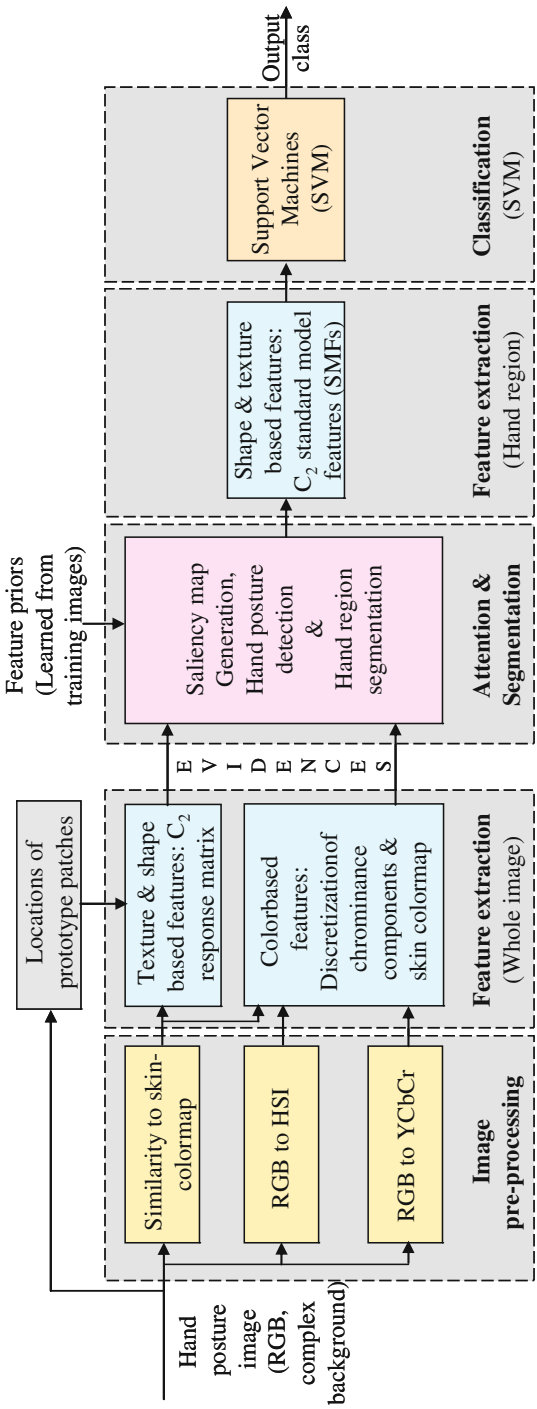


Fig. 8.3 Attention based hand posture segmentation and recognition (ASR) algorithm

Table 8.2 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

spaces (H, S, C_b, and C_r) are utilized[5] to detect 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 respectively represent the blue component ($B - Y$) and red component ($R - Y$) [4] (Y stands for the luminance value). The H and S values are in the range [0 1], and those of C_b and C_r are within [16 240].

8.3.1.2 Similarity-to-Skin Map

A skin similarity map (8.1) is created using the similarity of each pixel in HSI space to the average H and S pixel values of skin color.

$$S_{skin} = 1 - \sqrt{(H - H_{s0})^2 + \left(\frac{H_{smax} - H_{smin}}{S_{smax} - S_{smin}}\right)^2 (S - S_{s0})^2} \tag{8.1}$$

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 skin colors,
H_{smax} & H_{smin}	the maximum and minimum hue values of skin colors, and,
S_{smax} & S_{smin}	the maximum and minimum saturation values of skin colors

The average hue and saturation values are calculated by considering 10 skin colored pixel values of all subjects.[6] Values of different parameters in (8.1) are provided in Table 8.2. The hue value span (0.1770) is smaller than that of saturation (0.5692). The coefficient of saturation term ($(S - S_{s0})^2$) in (8.1) is a scaling factor to compensate for the span difference.

The similarity-to-skin map enhances the edges and shapes within skin colored regions in images, while preserving the textures (Fig. 8.4). The presented system extracts the shape and texture based features of hand postures from the skin similarity map. The feature extraction system detects and learns the edges and bars (at different orientations), and the textures in hand images. Utilization of the skin similarity map enhances the capability of the system to detect the hand region in complex background images.

[5] The luminance color components are not utilized as these components are sensitive to skin color as well as lighting.

[6] The dataset consists of hand postures by 40 subjects, with different ethnic origins.

Fig. 8.4 Sample hand posture images (*column 1—RGB*, *column 2—grayscale*) with corresponding skin similarity map (*column 3*). The skin similarity map enhanced the edges and shapes in hand posture images. The marked regions in *column 3* have better edges of the hand, as compared with that within the corresponding regions in *columns 1* and *2*. The edges and bars of the non-skin colored areas are diminished in skin similarity map (*column 3*). However, the edges corresponding to skin colored non-hand regions are also enhanced (*row 2*, *column 3*). This issue is addressed in ASR algorithm by utilizing shape and texture patterns of hand regions, in addition to color features

8.3.2 Extraction of Color, Shape and Texture Features

The ASR algorithm utilizes a combination of low and high level image features to develop the visual attention model. A saliency map is generated and attention is focussed on the hand region using color features (low level) and shape-texture features (high level). The postures are classified using the high level features.

8.3.2.1 Color Based Features

Jones and Rehg proposed a statistical color model for detecting skin regions in images [9], suggesting skin color as a powerful cue for detecting people in unconstrained imagery. However, the color model is developed only with the *RGB* color space. Analysis and comparison of different color spaces for skin segmentation are provided in [15] and [4]. Chaves-Gonzlez et al. [4] rates HSI space as the best choice for skin segmentation. Cb and Cr components in YCbCr space provided better results (compared to that provided by H–S and R–G components) in the experiments in [15]. A combination of HSI and YCbCr color spaces can improve segmentation of skin colored regions in images. The ASR algorithm utilizes a combination of chrominance color components (H, S, C_b, and C_r) of these color spaces as color features for hand region segmentation.

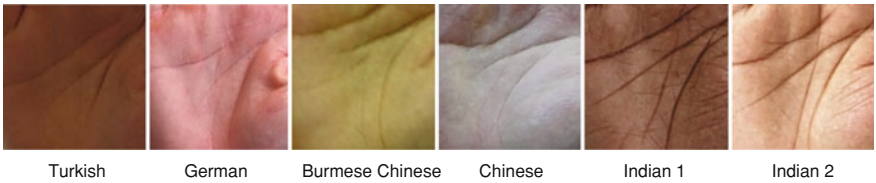


Fig. 8.5 Skin samples showing inter and intra ethnic variations in skin colors. Table 8.3 provides the average H, S, C_b, and C_r values of the six skin samples

Table 8.3 Average H, S, C_b, and C_r values of the six skin samples in Fig. 8.5

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 bold figures represent the maximum and minimum values in each column

The ASR algorithm generates the skin similarity map using average skin color components (H_{s0} and S_{s0} in Table 8.2). The skin colors and the corresponding component values vary about these mean values. Figure 8.5 shows six skin samples which have inter and intra ethnic variations in skin colors. The average chrominance component values of these samples are provided in Table 8.3. 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 variations, the ASR algorithm considers different ranges of color components.

Color features utilized in ASR algorithm are the discretized values of chrominance color components, and the similarity-to-skin map (Table 8.4). The color features (with shape and texture features) are utilized to calculate joint posterior probabilities of pixel locations to be part of a hand region. The values of H, S, C_b, C_r, and S_{skin} fall within [0 1] (C_b and C_r values are normalized). These components are quantized into 10 intervals. The range of the ith interval is given by $[(i-1)/10, i/10]$, where $i = 1, 2, ..., 10$. The features extracted are named as in Table 8.4. For example, the hue values between 0–0.1 is a feature and is named as H_1. Similarly, there are 50 color features (Table 8.4). The prior probabilities of these features are calculated by counting the frequency of the features in skin colored hand areas in images.[7] The features with maximum priors are H_1, H_2, S_4, C_{b5}, C_{r6}, and S_{skin10}. The color features are common for all hand postures (the position and frequency of the features may vary). Due to the commonality, the color features are utilized only for focussing attention on the hand, and not for the interclass discrimination of hand postures.

[7] 400 images (1 image per class per subject) are considered. During the training phase the hand area is selected manually.

Table 8.4 Color based features

Chrominance component features				Ranges	Skin similarity features	
H	S	C_b	C_r		S_{skin}	Ranges
H_1	S_1	C_{b1}	C_{r1}	0.0–0.1	S_{skin1}	0.0–0.1
H_2	S_2	C_{b2}	C_{r2}	0.1–0.2	S_{skin2}	0.1–0.2
H_3	S_3	C_{b3}	C_{r3}	0.2–0.3	S_{skin3}	0.2–0.3
H_4	S_4	C_{b4}	C_{r4}	0.3–0.4	S_{skin4}	0.3–0.4
H_5	S_5	C_{b5}	C_{r5}	0.4–0.5	S_{skin5}	0.4–0.5
H_6	S_6	C_{b6}	C_{r6}	0.5–0.6	S_{skin6}	0.5–0.6
H_7	S_7	C_{b7}	C_{r7}	0.6–0.7	S_{skin7}	0.6–0.7
H_8	S_8	C_{b8}	C_{r8}	0.7–0.8	S_{skin8}	0.7–0.8
H_9	S_9	C_{b9}	C_{r9}	0.8–0.9	S_{skin9}	0.8–0.9
H_{10}	S_{10}	C_{b10}	C_{r10}	0.9–1.0	S_{skin10}	0.9–1.0

a Discretized chrominance color components (H, S, Cb, Cr) in HSI and YCbCr color spaces. Each component is discretized into 10 subcomponents as shown. **b** Skin similarity features: Discretized skin color similarity (8.1) values

8.3.2.2 Shape and Texture Based Features

Shape and texture descriptors (Sect. 8.2.1) are extracted from the similarity-to-skin map. In ASR algorithm, 15 prototype patches with 4 patch sizes, are extracted from each of the 10 classes. To provide better invariance to variations in shape, 15 patches are taken from 15 different images of the same class. The patches contain filter responses along four orientations (0°, 45°, 90°, 135°). The total number of patches is 600, leading to 600 shape and texture based features.

8.3.3 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. Figure 8.6 shows the Bayes net utilized in ASR algorithm, which is developed based on the model proposed in [5]. The Bayes Net Toolbox (BNT) [11] is useful in implementing the Bayes net. The probabilistic model is given by (8.2) (LHS represents the joint probability of the Bayes net shown in Fig. 8.6 and RHS represents the probability of hand $P(O)$, the probability of location $P(L)$, and the conditional probabilities corresponding to the nodes in the Bayes net).

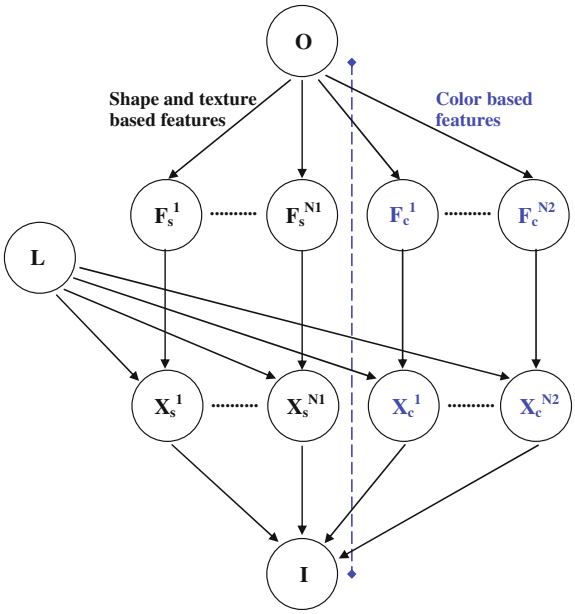


Fig. 8.6 Bayes net used in ASR system. O—the object (hand), L—the location of the hand, I—the image, ${F_s}^1$ to ${F_s}^{N1}$—N1 binary random variables that represent the presence or absence of shape and texture features, ${F_c}^1$ to ${F_c}^{N2}$—N2 binary random variables that represent the presence 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.

$$
\begin{aligned}
&P(O, L, {X_s}^1, \ldots, {X_s}^{N1}, {X_c}^1, \ldots, {X_c}^{N2}, {F_s}^1, \ldots, {F_s}^{N1}, {F_c}^1, \ldots, {F_c}^{N2}, I) \\
&\quad = P(O) \times P(L) \times \prod_{i=1}^{i=N1} \left\{ P({X_s}^i/L, {F_s}^i) P({F_s}^i/O) \right\} \\
&\qquad \times \prod_{j=1}^{j=N2} \left\{ P({X_c}^j/L, {F_c}^j) P({F_c}^j/O) \right\} \\
&\qquad \times P(I/{X_s}^1, \ldots, {X_s}^{N1}) P(I/{X_c}^1, \ldots, {X_c}^{N2})
\end{aligned} \tag{8.2}
$$

$$
\begin{aligned}
P(L/I) \propto P(L) \times \prod_i &\left\{ \sum_{{F_s}^i {X_s}^i} P({X_s}^i/L, {F_s}^i) P({F_s}^i/O) P(I/{X_s}^i) \right\} \\
\times \prod_j &\left\{ \sum_{{F_c}^j {X_c}^j} P({X_c}^j/L, {F_c}^j) P({F_c}^j/O) P(I/{X_c}^j) \right\}
\end{aligned} \tag{8.3}
$$

Table 8.5 Descriptions of conditional probabilities (priors, evidences and posterior probability)

Conditional probability	Represents	Calculation
$P({F_s}^i/O)$	Top-down shape and texture feature priors; probability of shape and texture features being present, given the presence of hand	By counting the frequency of occurrence of features[a] within the training images (maximum one count per image)
$P({F_c}^j/O)$	Top-down color feature priors; probability of color features being present, given the presence of hand	By counting the frequency of occurrence of features (Table 8.4) within the hand region in training images (400 images, single image per class per subject, are considered)
$P(I/{X_s}^i)$	Bottom-up evidence for shape and texture features; provides the likelihood that a particular location is active for shape and texture features	By the shape and texture feature extraction of test images (Sect. 8.2.1)
$P(I/{X_c}^j)$	Bottom-up evidence for color features; provides the likelihood that a particular location is active for color features	By the color feature extraction of test images (Sect. 8.3.2.1)
$P(L/I)$	Posterior probabilities of location, which acts as the saliency map	By the belief propagation algorithm [14]

[a] A feature is present if it is above a threshold value. Otherwise it is absent. In ASR algorithm, the threshold is set at 75 % of the maximum value of corresponding feature in training data

Feature-based attention depends on task-based priors and evidences. Posterior probabilities of locations (8.3), which serve as a 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 8.5). A belief propagation algorithm [14] is utilized for calculating posterior probabilities. Evidences from test images (for shape-texture features, $P(I/{X_s}^i)$, and color features, $P(I/{X_c}^j)$) are modulated by feature preferences (learned priors for shape-texture features, $P({F_s}^i/O)$, and color features, $P({F_c}^j/O)$). Locations of preferred features can be identified from posterior probability $P(L/I)$, which forms the saliency map.

8.3.4 Hand Segmentation and Classification

Hand region is segmented using the thresholded saliency map. For segmentation, a bounding box is created around the most salient (top 30 %) locations in an image. The shape and texture based features of hand region are extracted next. The same prototype patches selected during training (Sect. 8.2.1) are utilized for feature extraction. The C_2 SMFs are extracted by taking maximum over positions [23] of the C_2 response matrices. The value of the best match between a stored prototype and the input image is kept and the rest is discarded. An SVM classifier with linear kernel is utilized for the classification.

Figure 8.7 provides a pictorial summary of the ASR system, showing the image pre-processing, feature extraction, attention, and classification stages.

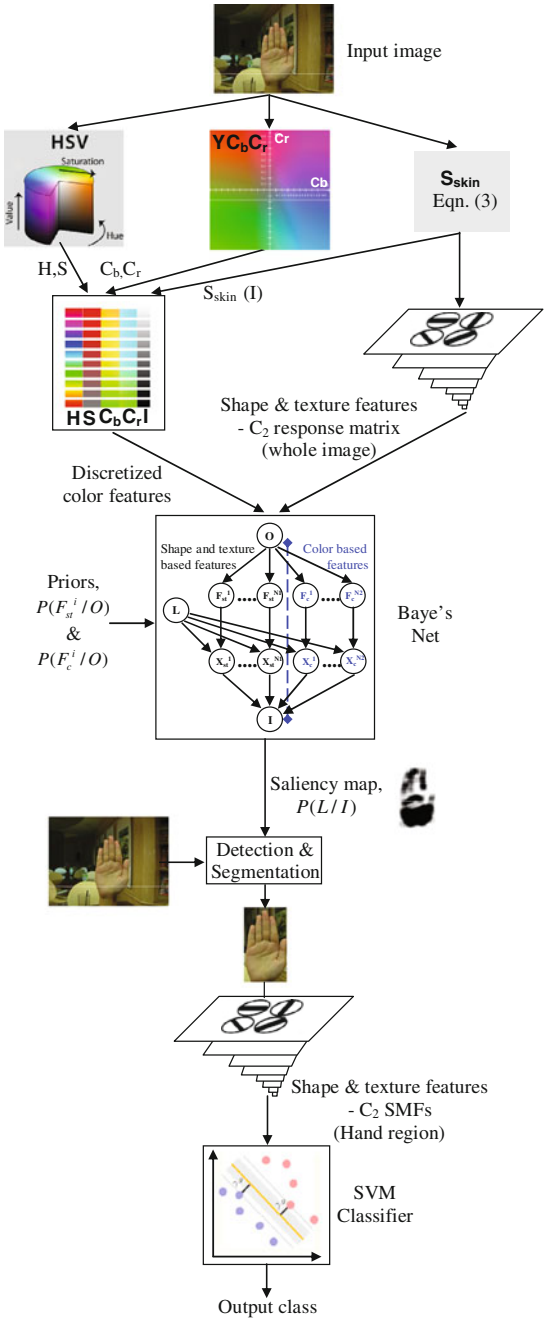


Fig. 8.7 An overview of the attention based hand posture segmentation and recognition (ASR) system

Fig. 8.8 Sample images from NUS hand posture dataset-II (data subset A), showing posture classes 1–10

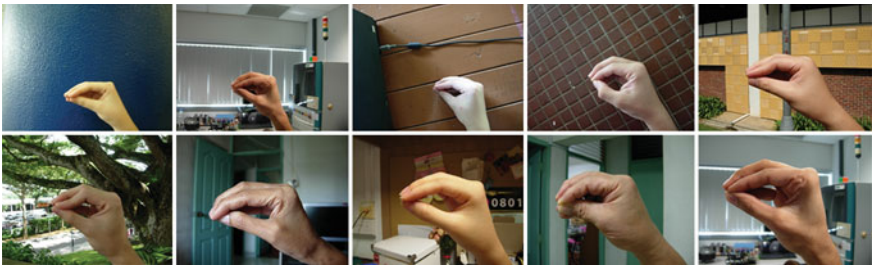


Fig. 8.9 Sample images (class 9) from NUS hand posture dataset-II (data subset A), showing variations in hand posture sizes and appearances

8.4 Experimental Results and Discussion

The ASR algorithm's performance is reported in this section. A 10 class complex background hand posture dataset is utilized for performance evaluation.

8.4.1 The Dataset and the Experimental Set-up

The number of available hand posture datasets is limited. The 10 class NUS hand posture dataset-II (Figs. 8.8 and 8.9) [16, 18] has complex background hand postures from 40 subjects with various ethnicities.[8] The hand postures were shot in and around National University of Singapore (NUS), against complex natural backgrounds, with various hand shapes and sizes. Both male and female subjects in the age range of 22–56 years are included. The subjects were asked to show 10 hand postures, 5 times each. They were asked to loosen hand muscles after each shot, in order to incorporate the natural variations in postures. The dataset consists of 3 subsets as detailed in Table 8.6.

[8] The dataset is available for academic research purposes: http://www.vadakkepat.com/NUS-HandSet/.

Table 8.6 Different subsets in NUS hand posture dataset-II

Subset	Consists
A	2,000 hand posture color images (40 subjects, 10 classes, 5 images per class per subject, image size: 160×120) with complex backgrounds
B	750 hand posture color images (15 subjects, 10 classes, 5 images per class per subject, image size: 320×240) with noises like body/face of the posturer or the presence of a group of humans in the background
C	2,000 background images without hand postures (used for testing the hand detection capability)

The ASR algorithm is evaluated in two aspects: hand posture detection and recognition. The hand posture detection capability is tested using data subsets A and C. The hand posture recognition capability is tested using data subsets A and B.

8.4.2 Hand Posture Detection

The hand postures are detected by thresholding the saliency map. To calculate the detection accuracy, saliency map is created using posterior probabilities of locations, for the set of hand posture and background images. Posterior probabilities above a threshold value indicate the presence of hand. Figure 8.10 shows the Receiver Operating Characteristics (ROC) of hand detection task (the ROC curve is plotted by decreasing threshold) by 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.

8.4.3 Hand Region Segmentation

Figure 8.11 shows the segmentation of hand region using the skin color similarity and the saliency map. The segmentation using skin color similarity performs well when the background does not contain skin colored regions (Fig. 8.11, column 1). However, natural scenes may contain skin colored objects (more than 70 % of the images in the NUS hand posture dataset-II have skin colored regions in the background). Segmentation using skin color similarity fails in such cases (Fig. 8.11, column 2 and 3). The ASR algorithm succeeded in segmenting complex hand images with skin colored pixels in backgrounds.

Figure 8.12 shows 50 sample images from the dataset (5 from each class) and the corresponding saliency maps. The hand regions are segmented using saliency maps (Fig. 8.11).

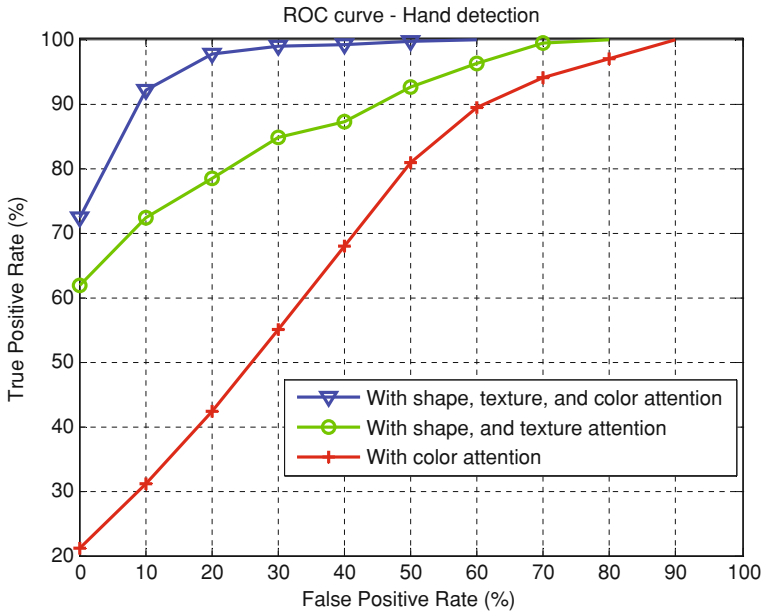


Fig. 8.10 Receiver Operating Characteristics of hand detection task. The graph is plotted by decreasing the threshold of the posterior probabilities of locations to be a hand region. Utilization of only shape-texture features provided reasonable detection performance (*green*) whereas utilization of only color features lead to poor performance (*red*) due to the presence of skin colored backgrounds. However, the ASR algorithm provided best performance (*blue*) when the color features are combined with shape-texture feature

8.4.4 Hand Posture Recognition

The hand posture recognition algorithm is tested using 10 fold cross validation using the data subset A. For cross validation the dataset is divided into 10 subsets each containing 200 images from 4 subjects. 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 8.7). On comparison, the best recognition rate (94.36 %) is achieved with shape, texture and color based attention.

When attention is implemented using shape and texture features, the ASR algorithm provided good improvement in accuracy (87.72 %) compared to the accuracy (75.71 %) achieved by the C_2 feature [23] based system without attention. The color feature attention alone provided lesser accuracy (81.75 %) compared to the accuracy with shape and texture attention. Accuracy for color feature attention drops with skin colored pixels in the background. Color features are extracted using point processing, whereas the shape-texture features are extracted using neighborhood processing. This is another reason for lesser accuracy with color feature attention. Color features combined with shape and texture features resulted in best accuracy (94.36 %).

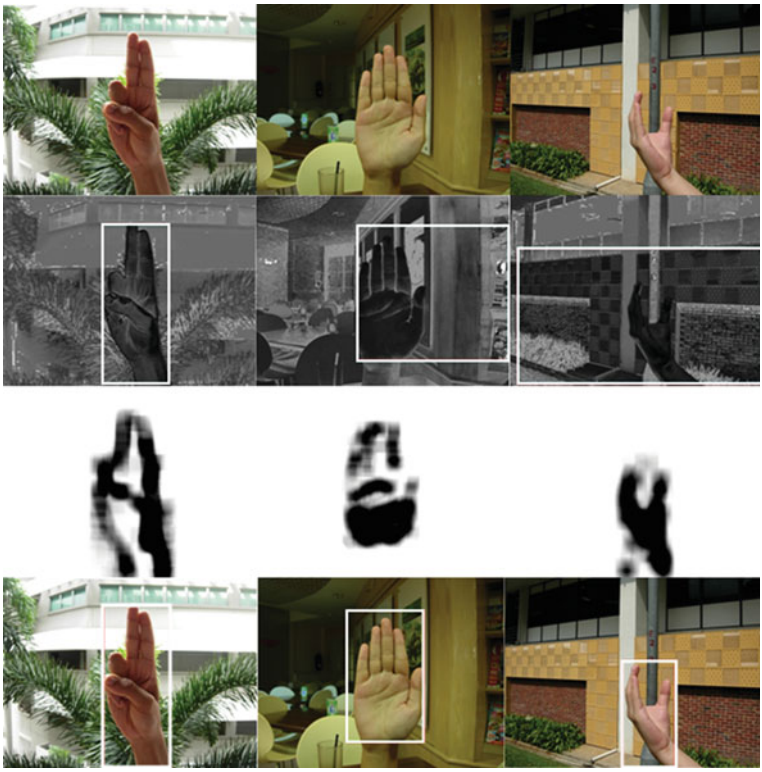


Fig. 8.11 Segmentation of hand region using similarity-to-skin map and saliency map. Each column shows the segmentation of an image. *Row 1* shows the original image, *row 2* shows the corresponding similarity-to-skin map (darker regions represent better similarity) with segmentation by thresholding, *row 3* shows the saliency map (only the top 30 % is shown), and *row 4* shows the segmentation using saliency map. The background of *image 1* (*column 1*) does not contain any skin colored area. The segmentation using skin similarity map succeeded for *image 1*. The backgrounds of *images 2* and *3* (*columns 2* and *3*) contain skin colored areas. The skin color based segmentation partially succeeded for *image 2*, but failed for *image 3* (which contains more skin colored background pixels compared to *image 2*). Segmentation using the saliency map (*row 4*) succeeded in all the 3 cases

Table 8.7 also shows a comparison of the accuracy provided by the ASR and EGM algorithms. The EGM algorithm [26] provided 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 images with skin colored complex backgrounds. The ASR algorithm has robustness to skin colored backgrounds as it utilizes shape and texture patterns with color features. The shape-texture selectivity of the C_2 feature extraction system is improved as the prototype patches are extracted from the geometrically significant and textured positions of the hand postures.

Fig. 8.12 Different sample images from the NUS hand posture dataset-II and the corresponding saliency maps. Five sample images from each class are shown. The hand region in an image is segmented using saliency map

Table 8.7 Hand posture recognition accuracies: data subset A

Method		Accuracy (%)
ASR system	Attention using shape, texture and color features	94.36
	Attention using shape and texture features	87.72
	Attention using color features	81.75
C_2 features without attention [23]		75.71
Elastic graph matching (EGM)[a] [26]		69.80

[a] The EGM algorithm in [26] is implemented as it is for the comparative study. The same sample divisions are utilized to test both ASR and EGM algorithms

8.4.5 Performance with Human Skin and Body Parts as Noises

The NUS hand posture dataset-II subset B is useful to test recognition capabilities on images with human in the background as noise. The data subset B contains images with noises like body or face of the posturer, or a group of other humans in the background (Fig. 8.13).

Training of the ASR algorithm is carried out using 200 images (4 subjects) from data subset A and the testing is done using data subset B (Table 8.6). As the ASR algorithm combines shape-texture features with color features, it is able to detect

Fig. 8.13 Sample images from NUS hand posture dataset-II data subset B. The subset contains images with human skin and body parts as noises

Table 8.8 Hand posture recognition accuracies: data subset B

Method	Accuracy (%)
ASR algorithm[a]	93.07
C_2 features without attention [23]	68.40
Elastic graph matching (EGM) [26]	62.13

[a]Attention using shape, texture and color features. Training is carried out using 200 images from data subset A and testing is done using data subset B

Table 8.9 Comparison of the recognition time

Method	Time (s)
ASR algorithm	2.65
Elastic graph matching (EGM) [26]	6.19

the hand region in images in spite of noise due to other skin colored human body parts (arm or face of the posturer, or other humans in the background). Table 8.8 shows the recognition accuracy in comparison with the C_2 feature based algorithm and EGM. The ASR algorithm provided a recognition rate (93.07 %) higher than the other methods.

8.4.6 Comparison of the Recognition Time

Table 8.9 provides a comparison of the average recognition times of the ASR and EGM algorithms (image size: 160×120 pixels, implemented in MATLAB computing platform). The ASR algorithm has a lesser recognition time compared to the EGM algorithm. The ASR algorithm can be made suitable for real-time applications by improving the response time of the shape and texture feature extraction system.

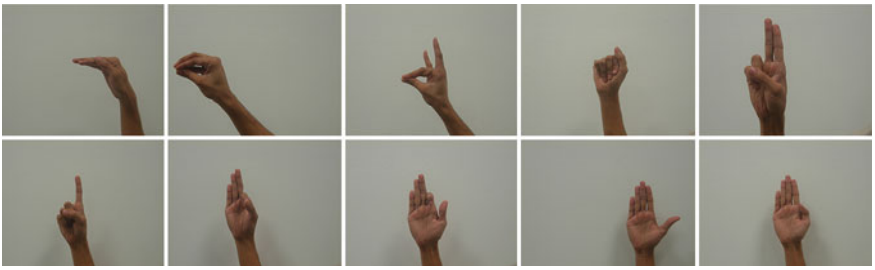


Fig. 8.14 Sample images from the NUS hand posture dataset-I, showing posture classes 1–10

8.4.7 Recognition of Hand Postures with Uniform Backgrounds

The ASR system is tested with a simple background dataset, the NUS hand posture dataset-I[9] [16, 19] (Fig. 8.14). Ten fold cross validation provided an accuracy of 96.88 %. The effectiveness of the ASR algorithm for recognition of postures with uniform backgrounds is evident. However, attention based approach does not have much impact in the case of uniform background postures. The system without attention provided an accuracy of 95.83 % which is comparable with the attention based system. This implies that the attention based system is necessary only for the recognition in complex environments.

8.5 Summary

This chapter discussed the attention based segmentation and recognition (ASR) system for recognizing 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 algorithm is tested with a 10 class complex background dataset, the NUS hand posture dataset-II.

The ASR algorithm has a person independent performance. It provided good hand posture detection and recognition accuracy in spite of variations in hand sizes. The algorithm provided reliable performance against cluttered natural environments including skin colored complex backgrounds. The performance of ASR algorithm is reported with color attention, shape and texture attention, and a combination of color, shape and texture attention. In comparison, the algorithm provided the best recognition accuracy when the combination of color, shape, and texture attention is utilized.

[9] The dataset is available for free download: http://www.vadakkepat.com/NUS-HandSet/.

8.5.1 Possible Extension and Application of ASR Algorithm

The ASR algorithm can be extended for other shape recognition tasks like human body posture recognition for 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, body postures provide more reliable texture features compared to hand postures. The body pose can be estimated part-by-part or hierarchically (for example, skin colored regions first and then textured regions).

Acknowledgments Figures and tables in this chapter are adapted from the following article with kind permission from Springer Science+Business Media: International Journal of Computer Vision, Attention Based Detection and Recognition of Hand Postures Against Complex Backgrounds, Vol.101, Issue No.3, 2013, Page Nos. 403-419, Pramod Kumar Pisharady, Prahlad Vadakkepat and Loh Ai Poh.

References

1. V. Athitsos, S. Sclaroff, Estimating 3d hand pose from a cluttered image. IEEE Conf. Comput. Vis. Pattern Recogn. **2**, 432–439 (2003)
2. E. Bienenstock, C. von der Malsburg, A neural network for invariant pattern recognition. Europhys. Lett. **4**(1), 121–126 (1987)
3. C. Bishop, *Neural Networks for Pattern Recognition* (Oxford, Oxford University Press, 1995)
4. J.M. Chaves-González, M.A. Vega-Rodrígueza, J.A. Gómez-Pulidoa, J.M. Sánchez-Péreza, Detecting skin in face recognition systems: a colour spaces study. Digit. Signal Process. 20(03), 806–823 (2010)
5. S. Chikkerur, T. Serre, C. Tan, T. Poggio, What and where: a bayesian inference theory of attention. Vis. Res. 50(22), 2233–2247 (2010)
6. P. Dayan, G.E. Hinton, R.M. Neal, The helmholtz machine. Neural Comput. **7**(5), 889–904 (1995)
7. L. Itti, C. Koch, Computational modelling of visual attention. Nat. Rev. Neurosci. **2**(3), 194–203 (2001)
8. L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
9. M.J. Jones, J.M. Rehg, Statistical color models with application to skin detection. IEEE Conf. Comput. Vis. Pattern Recogn. **1** (1999)
10. M. Kolsch, M. Turk, Robust hand detection. IEEE Conf. Autom. Face Gesture Recogn. 614–619 (2004)
11. K. Murphy, Bayes net toolbox for matlab (2003), http://code.google.com/p/bnt/
12. E. Niebur, C. Koch, Computational architectures for attention, in *The Attentive Brain*, ed. by R. Parasuraman (Cambridge, MIT Press, 1998) pp. 163–186
13. E.J. Ong, R. Bowden, A boosted classifier tree for hand shape detection. IEEE Conf. Autom. Face Gesture Recogn. 889–894 (2004)
14. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann Publishers, California, 1988)
15. S.L. Phung, A. Bouzerdoum, D. Chai, Skin segmentation using color pixel classification: analysis and comparison. IEEE Trans. Pattern Anal. Mach. Intell. **27**(01), 148–154 (2005)
16. P.K. Pisharady, Computational intelligence techniques in visual pattern recognition. Ph.D. Thesis, National University of Singapore (2011)

17. P.K. Pisharady, P. Vadakkepat, 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
18. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Attention based detection and recognition of hand postures against complex backgrounds. Int. J. Comput. Vis. **101**(03), 403–419 (2013)
19. P.K. Pisharady, P. Vadakkepat, A.P. Loh, Hand posture and face recognition using a fuzzy-rough approach. Int. J. Humanoid Rob. **07**(03), 331–356 (2010)
20. T. Poggio, E. Bizzi, Generalization in vision and motor control. Nature **431**, 768–774 (2004)
21. R. Rao, Bayesian inference and attentional modulation in the visual cortex. Neuro Rep. **16**(16), 1843–1848 (2005)
22. M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex. Nat. Neurosci. **2**(11), 1019–1025 (1999)
23. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
24. C. Siagian, L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 300–312 (2007)
25. J. Triesch, C. Malsburg, Sebastien marcel hand posture and gesture datasets: Jochen triesch static hand posture database (1996), http://www.idiap.ch/resource/gestures/
26. J. Triesch, C. Malsburg, A system for person-independent hand posture recognition against complex backgrounds. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1449–1453 (2001)
27. J. Triesch, C. Malsburg, Robust classification of hand postures against complex backgrounds. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition* (Killington, VT, USA), October 1996, pp. 170–175
28. J.K. Tsotsos, S.M. Culhane, Y.H. Wai, W.Y.K. Lai, N. Davis, F. Nuflo, Modelling visual attention via selective tuning. Artif. Intell. **78**(1–2), 507–545 (1995)
29. Y. Wu, T.S. Huang, View-independent recognition of hand postures. IEEE Conf. Comput. Vis. Pattern Recogn. **2**, 88–94 (2000)

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

[Referred in Chap. 4, Sect. 4.2.1]
Figure A.1a–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 are interclass feature overlaps. Learning classification rules from the features having distribution similar to that shown in Fig. A.2 (which has higher interclass overlap) is difficult and may lead to misclassification. The proposed algorithm (Chap. 4) neglects such features, and excludes the corresponding rules from the classifier rule base. This increases the classification accuracy, and provides better margin of classification.

Once the feature values A_L and A_H are identified, classification is done by the voting process using *Rule 1* and *2* (Eqs. 4.6 and 4.7).

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8,
© Springer Science+Business Media Singapore 2014

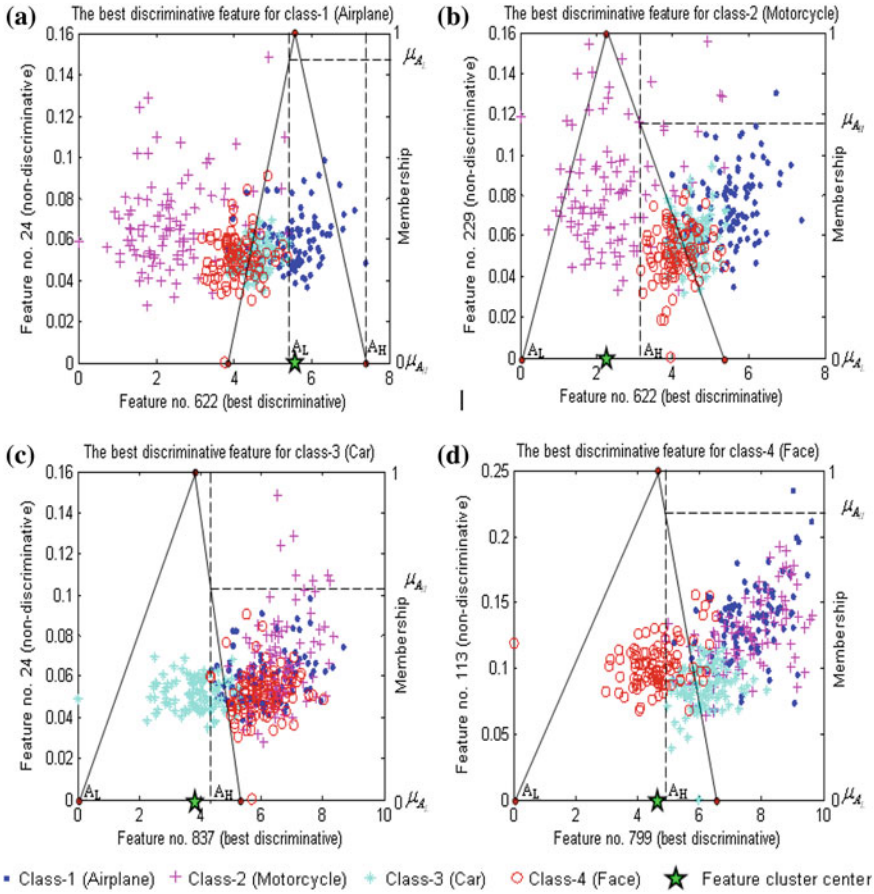


Fig. A.1 Illustration of the formation of fuzzy membership functions, and the calculation of $\{\mu_{A_L}, \mu_{A_H}\}$ and $\{A_L, A_H\}$ for object dataset. Subfigures **a**–**d** show a two dimensional distribution (only two feature *axes* are shown) of training samples in the object dataset (class 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 is 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 $\{A_L, A_H\}$ (Sect. 4.2.1), for the four object classes

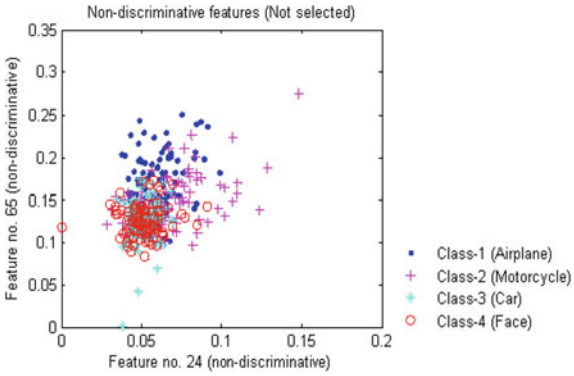


Fig. A.2 Two dimensional distribution of samples in the object dataset, with *x* and *y-axes* representing two non-discriminative features. The features have high interclass overlap with the cluster centers closer to each other. Such features are filtered out by the feature selection algorithm

Index

A
Acronyms, list of, xiii
Attention based segmentation and recognition, ASR, 107
Attention, Bayesian model of, 112
Attribute reduction, 22

B
Bayes net, 113, 119
Biologically inspired features, 15
Boosting, 86
Boosting-based fuzzy-rough multi-cluster, BFRMC, 81
Bunch graph, 98

C
C_2 features, 16
Cancer classification, 52
Certain rules, 83
Chromosome, 86
Classification, 47, 65, 102
Cluster centers, 84
Cluttered backgrounds, 7
Color features, 117
Complex background, 107
Computational complexity, 50, 72
Crisp equivalence class, 83
Crossover, 14, 86
Crossover rate, 89

D
Data graph, 30
Dynamic gestures, 23

E
Edges, 30
Eigen space method, 32

F
Face recognition, 58, 74
Feature attention, 112
Feature cluster centers, 84
Feature extraction, 110
Feature selection, 21, 44, 69, 103
Fitness criteria, 87
Fuzzy discretization, 83
Fuzzy equivalence class, 43, 83
Fuzzy partitioning, 84
Fuzzy rough classifier, 42
Fuzzy rough multi-cluster classifier, boosting based, 83
Fuzzy rough sets, 13
Fuzzy similarity relation, 83
Fuzzy upper approximation, 82
Fuzzy-rough multi cluster classifier, 64
Fuzzy-rough multi cluster, FRMC, 63
Fuzzy-rough sets, 82
Fuzzy-rough single cluster, FRSC, 41

G
Gabor jets, 31
Genetic algorithm, 14, 86
Genetic algorithm, boosting enhanced, 83
Graph algorithm, 30
Graph matching, 96

H
Hand gesture databases, 32

P. K. Pisharady et al., *Computational Intelligence in Multi-Feature Visual Pattern Recognition*, Studies in Computational Intelligence 556, DOI: 10.1007/978-981-287-056-8,
© Springer Science+Business Media Singapore 2014

Hand gesture recognition, issues in, 6
Hand gesture recognition, survey of, 23
Hand model, 24
Hand posture detection, 124
Hand posture recognition, 57, 76, 98, 101, 125
Hand postures, 23
Hand region segmentation, 124
Hidden Markov Model, 24

I
Image features, 6
Iterative boosting, 89
Iterative rule learning, 82, 86

K
Kinect, 3

L
Low dimensional fuzzy rules, 83
Lower and upper approximations, 45

M
Margin of classification, 63, 68
Model based methods, 31
Model graph, 30
Mutation, 14, 86
Mutation rate, 89

N
Neural network, 26

O
Object recognition, 58

P
Pattern recognition, pipeline of, 4
Plausibility factor, 88
Population, 86
Possible rules, 83, 86
Pre-filter, 70

R
Recognition time, 128
Rough sets, 12

S
Saliency map, 6, 121
Shape-texture features, 119
Similarity-to-skin map, 116
Spatial attention, 112
Standard Model Features, 16
Statistical and syntactic analysis, 31
Subtractive clustering, 43,65
Symbols, list of, xiii

U
Uniform background, 129

V
Vertices, 30
Visual attention, 6
Visual pattern, recognition of, 3