

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Symmetry and Feature Selection in Computer Vision

Permalink

<https://escholarship.org/uc/item/5v46938v>

Author

Sun, Yu

Publication Date

2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Symmetry and Feature Selection in Computer Vision

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Yu Sun

June 2012

Dissertation Committee:

Dr. Bir Bhanu, Chairperson

Dr. John Heraty

Dr. Amit Roy-Chowdhury

Copyright by
Yu Sun
2012

The Dissertation of Yu Sun is approved:

Committee Chairperson

University of California, Riverside

ABSTRACT OF THE DISSERTATION

Symmetry and Feature Selection in Computer Vision

by

Yu Sun

Doctor of Philosophy, Graduate Program in Electrical Engineering

University of California, Riverside, June 2012

Dr. Bir Bhanu, Chairperson

In the dissertation, two advanced computer vision techniques, named symmetry and feature selection, are proposed. The wide existence of symmetry in many image objects generates the motivation of using symmetry as a high level feature in region growing image segmentation and region-of-interest (ROI) detection in brain MRI sequences. The symmetry is explicitly applied in different forms as symmetry affinity matrix, high-level segmentation cue, statistical analysis and 3D asymmetry volume in classification features. The incorporation of symmetry provides a new effective feature to achieve the performance improvement. In the second field of my research, the feature selection with Sequential Floating Forward Selection (SFFS) as the search strategy, and with the Bayesian classifier as the evaluation metric, is applied in content-based image retrieval (CBIR), semi-supervised learning with relevance feedback, local kernel based distance metric, image classification, and online ensemble learning. It provides more compact and optimal feature sets to generate robust learning models. Experimental results on wide range of image datasets indicate the advantages of using symmetry and feature selection in computer vision tasks.

Table of Contents

List of Tables.....	x
List of Figures.....	xii
1 Introduction.....	1
1.1 Symmetry and Feature Selection in Computer Vision.....	1
1.2 Contributions of the Dissertation	4
1.3 Dissertation Organization	5
2 Symmetry Integrated Image Segmentation.....	7
2.1 Introduction.....	7
2.2 Related Work and Contributions.....	11
2.2.1 Related Work.....	11
2.2.2 Contributions	12
2.3 Symmetry Integrated Region Growing Segmentation.....	13
2.3.1 Discrete Reflection Symmetry Detection and the Symmetry Affinity Matrix.....	15
2.3.2 Symmetry Integrated Region Growing.....	17
2.3.3 Performance Evaluations of Segmentation and Symmetry	23
2.3.4 Multi-objective Optimization for Segmentation and Symmetry	26
2.3.5 Algorithm for the Proposed Segmentation Method.....	27
2.4 Experimental Results	28
2.4.1 Datasets and Parameters	28
2.4.2 Performance Metrics.....	29

2.4.3 Performance of the Proposed Method	30
2.4.4 Symmetry-integrated Region Growing vs. Other Segmentation	40
2.4.5 Symmetry-integrated Region Growing vs. Current Symmetry-Based Segmentation	42
2.4.6 Supervised vs. Unsupervised Evaluations	43
2.4.7 Statistical Validation of Results	45
2.4.8 Discussion of the Results	47
2.5 Conclusions	47
3 Automated Symmetry-Integrated Brain Abnormality Detection in MRI Sequences	49
3.1 Introduction	49
3.2 Related Works and Contributions	52
3.2.1 Related Works	52
3.2.2 Contributions	55
3.3 The Technical Approach	57
3.3.1 Symmetry Extraction	58
3.3.2 2D Segmentation and Extraction of Asymmetric Segments	61
3.3.3 3D Relaxation-based Segmentation of Symmetry Affinity Matrix	66
3.3.4 Asymmetric Region Extraction and ROI Detection	69
3.4 Experiments	72
3.4.1 Datasets and Parameters	72
3.4.2 Performance Metrics	74

3.4.3 The ROI detection results	75
3.4.4 Discussion of the results	75
3.4.5 Comparison with other Methods.....	82
3.5 Conclusions.....	85
4 Image Retrieval with Feature Selection and Relevance Feedback.....	87
4.1 Introduction.....	87
4.2 Technical Approach	88
4.2.1 Measure of Inconsistency from Relevance Feedback	90
4.2.2 Feature Selection Combined with User Feedback.....	91
4.3 Experiments	94
4.3.1 Datasets.....	94
4.3.2 CBIR Combined with Feature Selection and Relevance Feedback	95
4.4 Conclusions.....	97
5 Local Kernel Integrated Feature Selection for Image Classification.....	98
5.1 Introduction.....	98
5.2 Related Work and Contributions.....	100
5.2.1 Related Work.....	100
5.2.2 Contributions	102
5.3 Technical Approach	102
5.3.1 Image Segmentation and Feature Extraction.....	103
5.3.2 Local Feature Selection	105
5.3.3 Multiple Kernel Learning for Classifier Selection	109

5.4 Experiments	110
5.4.1 Datasets.....	111
5.4.2 Algorithms Compared and Parameters	112
5.4.3 Results of the Proposed Approach.....	112
5.4.4 Compare with Local Feature Selection	113
5.4.5 Compare with Global Feature Selection.....	114
5.4.6 Results of the Proposed Method with Various Segmentation and Feature Selection Methods	116
5.4.7 Statistical Validation of the Results.....	118
5.4.8 Discussion of the Results.....	118
5.5 Conclusions.....	118
6 Incremental Ensemble Learning for Classification	120
6.1 Introduction.....	120
6.2 Related Work and Contributions.....	123
6.2.1 Related Work.....	123
6.2.2 Contributions	125
6.3 Technical Approach	125
6.3.1 Image Feature Extraction.....	127
6.3.2 Discriminative Assignment of Training Data.....	128
6.3.3 Parallel Feature Selection and Classifier Learning.....	130
6.3.4 The Incremental Learning	132

6.3.5 The Ensemble Classification	134
6.4 Experimental Results	135
6.4.1 Datasets and Parameters	135
6.4.2 Incremental Assignment of Training Data	135
6.4.3 Incremental Ensemble Feature Selection and Classification.....	137
6.4.4 Comparison with other Classifier Systems.....	139
6.4.5 Incremental Learning with Additional Training Data	141
6.4.6 The Statistical Validation	142
6.4.7 Discussions of Results	143
6.5 Conclusions.....	144
7 Conclusions and Future Work	145
Bibliography	147
Appendix.....	160

List of Tables

Table 2.1. State-of-the-art Symmetry Based Image Segmentation Methods Integrating Symmetry: Summary and Their Limitations	12
Table 2.2. State-of-the-art Image Segmentation Methods Integrating Symmetry: List of Their Key Limitations	12
Table 2.3. Definition of symbols used in Section 2.3	14
Table 2.4. The Symmetry Detection Algorithm.....	15
Table 2.5. Region Growing Segmentation with Dynamic Region Weights Allocation Algorithm.....	18
Table 2.6. The Overall Algorithm for the Proposed Method	27
Table 2.7. Principles of State-of-the-Art Segmentation Methods.....	32
Table 2.8. Numerical Comparison of Segmentation Performance.....	33
Table 2.9. Numerical Comparison of Optimal Segmentation Performance: Supervised vs. Unsupervised Evaluations	43
Table 2.10. Statistical Validation on 15 Images from UCB Database, and on 93 Images from Caltech-101 Database	46
Table 3.1. Three Problems Related to the Automated ROI Detection Methods	52
Table 3.2. State-of-the-art ROI Detection Methods for MRI.....	54
Table 3.3. State-of-the-art Symmetry-based ROI Detection Methods for MRI	55
Table 3.4. Contributions of the Proposed Method	57
Table 3.5. Quantitative Analysis of MRI Dataset #A	77
Table 3.6. Quantitative Analysis of MRI Dataset #B.....	78
Table 3.7. Quantitative Analysis of MRI Dataset #C.....	79

Table 3.8. Quantitative Analysis of MRI Dataset #D	80
Table 3.9. Quantitative Analysis of Animal Sequences for all the Time Points	80
Table 3.10. Comparison With Other Methods	84
Table 3.11. Comparison of Animal Sequences for all the Time Points.....	84
Table 4.1. Comparison of retrieval precisions	97
Table 5.1. Summary of the related local feature selection methods	103
Table 5.2. Comparison of the final classification results, with corresponding feature subset dimensions and contents	117
Table 6.1. Symbols and their meaning used in this section	127
Table 6.2. Comparison of different methods for Caltech-101 database.....	139
Table 6.3. Comparison of different methods for event database	139
Table 6.4. Comparison of the different non-ensemble classification methods	140
Table 6.5. Comparison of different ensemble methods	141
Table 6.6. The hypothesis test of the proposed method	143

List of Figures

Fig. 2.1. (a) primitive symmetry categories, (b) combined (extended) symmetry categories, (c) discrete and continuous symmetry	8
Fig. 2.2. System diagram for symmetry-integrated image segmentation	15
Fig. 2.3. (a) Integration of symmetry in region growing, (b) graphic illustration of Equation (2.10)	21
Fig. 2.4. Symmetry-integrated image segmentation using the image from UCB dataset [43]: (a) original image, (b) SIFT points, (c) symmetry axis, (d) symmetry affinity of image, (e) symmetry-integrated segmentation, (f) the ground-truth segmentation provided by UCB dataset [43], (g) performance curve of segmentation and symmetry	30
Fig. 2.5. Symmetry-integrated segmentation using images from the Caltech-101.....	31
Fig. 2.6. Comparison of results on UCB database [43]: ‘Building’, ‘Man’ and ‘Woman_1’. (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) normalized cut with symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves	35
Fig. 2.7. Results for images, with different symmetry levels, from the UCB database [43], (a) original image, (b) ground-truth segmentation provided by UCB database, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) symmetry level, (f) segmentation improvement (from (d) to (c)). N/A: Not Applicable.....	36

Fig. 2.8. Images with occluded symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis detection, (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d))37

Fig. 2.9. Image ‘Man’ in Fig. 2.6, with (1) affine transform, (2) perspective transform, from UCB database [43], (a) transformed image, (b) symmetry axis, (c) ground-truth segmentation provided by UCB database, (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d))37

Fig. 2.10. Results for images, with articulated symmetry distortions, from the Caltech-101 database [42] (image (1)), and from the Internet (image (2)), (a) original image, (b) symmetry axis detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)).....37

Fig. 2.11. Results with images for incorrect symmetry detection, from the UCB database [42], (a) original image, (b) symmetry detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) distortions, (f) segmentation improvement ((d) to (c)).....38

Fig. 2.12. Images with multiple symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis (with high intensity as the dominant axis), (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement ((e) to (d))39

Fig. 2.13. Results with decreased segmentation performance by using symmetry, from the Caltech-101 database [42], (a) original image, (b) symmetry axis, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)). Note that the symmetry axes are incorrectly detected46

Fig. 3.1. System diagram for symmetry integrated region-of-interest (ROI) detection in MRI sequences.....56

Fig. 3.2. Sample results of symmetry extraction for injured brains of (1) human, (2) animal, and (3) human.....58

Fig. 3.3. Sample results of 2D/3D segmentation, for injured brains of (1) human, (2) animal, (3) human. In (b), pixels within the red boundaries are asymmetric segments.....61

Fig. 3.4. (a) Kurtosis: pdf for the Pearson distribution with kurtosis of infinity; 100; and 10; (b) Skewness: negative (left-tailed) and positive (right-tailed) distributions of skewness65

Fig. 3.5. Sample results for the extraction of asymmetric groups by 3D relaxation segmentation, for injured brains of (1) human, (2) animal, (3) human. Pixels within the red boundaries are asymmetric groups69

Fig. 3.6. Results for (1) human injury, (2) animal injury, (3) human injury 71

Fig. 3.7. The generation of 3D asymmetry volume of a MRI sequence72

Fig. 3.8. The illustration of true positive (TP), false positive (FP) and false negative (FN), and the associated true positive rate (TPR) and precision (PRS). The ROI detected by the approach, described in this paper, is shown in bold and the ROI given as the ground-truth is shown as non-bold..... 74

Fig. 3.9. Example results of the proposed method in different steps	77
Fig. 3.10. ROI of MRI dataset #C: a patient on the first day	78
Fig. 3.11. ROI of MRI dataset #D: a patient on the 7 th day	79
Fig. 3.12. Comparison of ROI detection results	18
Fig. 3.13. Comparison of ROI detection results: %ROI in volumetric measures.....	85
Fig. 4.1. The overall CBIR system diagram	90
Fig. 4.2. The feature selection diagram with user feedback	93
Fig. 4.3. Example images in databases, and performance comparison of the proposed method.....	96
Fig. 5.1. The overall system diagram for this paper	103
Fig. 5.2 Sampling of regions for each cluster, dot points: training sample of regions	109
Fig. 5.3. Segmentations and important regions (with P values) for the two databases, (a) sample images with class names, (b) region growing segmentation, (c) watershed segmentation, (d) normalized cut segmentation	111
Fig. 5.4. Local kernel method compared with local feature selection method, for Caltech-101 database, with feature selection methods, (a) SFFS-Bayesian, (b) mRMR, (c) entropy	113
Fig. 5.5. Comparison of classification results with feature selection iterations, (a) Caltech-101 database, (b) Butterfly database	115
Fig. 5.6. Statistics of selected features for the results shown in Table 5.2, for (a) Caltech-101 database, (b) Butterfly database	116
Fig. 6.1. The overall system diagram.....	126
Fig. 6.2. The discriminative training data assignment algorithm	129

Fig. 6.3. Comparison of training data assignment, (a) random assignment, (b) discriminative assignment.....	129
Fig. 6.4. The diagram for parallel feature selection and classifier learning.....	130
Fig. 6.5. The ensemble classification algorithm: Arc-x4.....	133
Fig. 6.6. Example images of the two image databases	135
Fig. 6.7. Comparison of classification results obtained from different number of ensemble classifiers learned from parallel feature selection, for the Caltech-101 database	137
Fig. 6.8. Incremental learning with additional training data.....	142
Fig. A1: Comparison of results on UCB database [43]: ‘Vase’, ‘Bear’ and ‘Woman_2’, (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing - no symmetry, (e) normalized cut - symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves	161
Fig. A2: Comparison of results on UCB database [43]: ‘Butterfly’ and ‘Fresco’. original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) normalized cut - symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves	162
Fig. A3: Examples of symmetry-integrated segmentation results using images from the Caltech-101 database	163

Fig. A4. A list of 93 images selected from the Caltech-101 image database

The number shown below an image corresponds to the image numbers in a category in the Caltech-101 database. We provide these numbers so that our results can be replicated. These 93 images are used for the statistical validation of the proposed method, in Section 2.4.7 and Table 2.10 164

Fig. A5: A list of 15 images selected from the UCB image database. These 15 images are used for the statistical validation of the proposed method, in Section 2.4.7 and Table 2.10 167

Chapter 1

Introduction

1.1. Symmetry and Feature Selection in Computer Vision

In many areas of commercial, academic and medical institutes, large collections of digital images are being created and analyzed. Traditionally, the only way of analyzing these collections was by keyword indexing, or simply by browsing. State-of-the-art computer vision techniques however, open the way to content-based image analysis. The term "Content-Based" means that the system will analyze the actual contents of the image rather than the metadata such as keywords, tags, and/or descriptions associated with the image. The term 'content' in this context might refer to features that can be derived directly from the image itself. In this dissertation, the symmetry is used as a high level

feature in image segmentation and object recognition tasks. Moreover, the feature selection is also applied in multiple computer vision tasks, e.g., the content based image retrieval (CBIR) and the image classification. Generally speaking, the computer vision tasks using the features facilitates the process of,

- Image preprocessing: that is, digitizing existing collections of analogue image forms, and then performing numerous preprocessing methods on the image, e. g., de-noising, contrast enhancement, smoothing and so on.
- Image feature extraction, processing and learning: that is, transforming the input image into set of features (feature extraction), and process the features to an effective forms (feature selection and learning).
- Computer vision and pattern recognition: different tasks performed based on the image features.

The feature based methods has become ubiquitous as a fundamental tool for analyzing and managing digital image, as the requirements of processing of large amount of image data increases and storage space becomes more affordable. The current research in this area mainly focus on feature extraction, feature selection/learning, and using image features on different computer vision tasks such as image segmentation, content based image retrieval (CBIR), or image classification and indexing.

As the main roll, the image features support low level features (e.g., color, shape, texture), which are adequate in traditional application domains such as image segmentation and object edge detection. However, the low level features have the problems of high dimensionality, large redundancy and low accuracy. On the other hand,

high level features (e.g., concepts, semantics) are proven to be more effective both in performance and time complexity. As a result, they are widely applied in much more complex and real-time tasks, e.g., object recognition and image retrieval. The wide existence of symmetry in many natural and manmade image objects generates the motivation of using symmetry as a high level feature in different computer vision tasks. In this dissertation, the symmetry is used in the following tasks,

- (1) Image segmentation: symmetry is used as a new constraint to improve the region growing image segmentation performance.
- (2) ROI detection in brain MRI: symmetry is integrated in several steps to make the region-of-interest (ROI) detection automatic and with higher performance.

The feature selection works as selecting a subset of relevant image features to generate a feature sets with reduced dimensions. The objective of feature selection is three-fold: improving the performance of the learning models, providing faster and more cost-effective models, and providing a better understanding of the underlying process that generated the image data. The feature selection also alleviates the problems of curse of dimensionality, feature redundancies and feature noises. In this dissertation, I apply the feature selection into complex computer vision and pattern recognition tasks to improve their performance,

- (3) Content based image retrieval (CBIR): retrieval model design by combination of user relevance feedback and feature selection.
- (4) Region-based image classification: classification design by local-kernel with feature selection.

(5) Online ensemble learning: classification design by ensemble classification, online incremental learning and parallel feature selection.

The contribution of the dissertation is shown in the following section. And the details of above five work will be explained in Chapters 2, 3, 4, 5, and 6.

1.2. Contributions of the Dissertation

This dissertation presents high level features, as well as the feature selection methods to handle computer vision problems and improve their performance.

For high level features, a new symmetry constraint is used into the region growing image segmentation. It segments the symmetric regions effectively with more symmetric and complete region boundaries. It also alleviates the problems of over-segmentation and incomplete regions. Experimental results shows it has better performance than other well-known image segmentation methods. Furthermore, the symmetry is also integrated in several steps for segmentation, region extraction and classification, in a region-of-interest (ROI) detection system on brain MRI. It is free from manual training of models and prior knowledge.

For feature selection, it is widely applied in different complex computer vision tasks. In content-based image retrieval (CBIR), the feature selection is combined with the user relevance feedback, as a new term called “measure of inconsistency”. It highly improves the retrieval accuracy, with lower user feedback iterations. In image classification, a new local kernel is invented in the Bayesian classification model, to enables the region-based image distance measure. A feature selection scheme is also performed to generate more

effective feature subsets. In online ensemble learning, a parallel feature selection is proposed to train multiple classifiers, which are then ensemble for a more effective classification model.

1.3. Dissertation Organization

This dissertation consists of six chapters.

Chapter 2 presents a symmetry integrated region based image segmentation method to improve the segmentation of images with symmetric objects. This method use symmetry as a high-level feature, along with traditional low level features such as colors and textures.

Chapter 3 presents an automatic symmetry based ROI detection method for brain MRI. The symmetry is used in segmentation to partition brain image into tissue regions, filter out symmetric regions using statistical computation on symmetry affinity, and classify the asymmetric regions into ROI using 3D asymmetry volume.

Chapter 4 presents a content based image retrieval (CBIR) system using feature selection and relevance feedback. In user feedback iteration, the relevant and non-relevant images generate a “measure of inconsistency”, which is then used to guide the feature selection procedure. The selected feature subsets are used in the K-nearest neighbor (KNN) search to retrieve the images upon the user query. The user feedback takes place again on the retrieved images, to build a closed CBIR loop.

Chapter 5 presents a local kernel based image classification scheme. A region based image distance is integrated into the kernel of Bayesian classifier, to train an image

classification system. The feature selection is performed based on the local (region-based) features.

Chapter 6 presents an ensemble learning using parallel feature selection. Multiple Bayesian classifiers are learned using different selected feature subsets. They are further ensemble using Arc-x4 to generate an improved image classification accuracy.

Chapter 7 summarizes this document and points out our future work.

Chapter 2

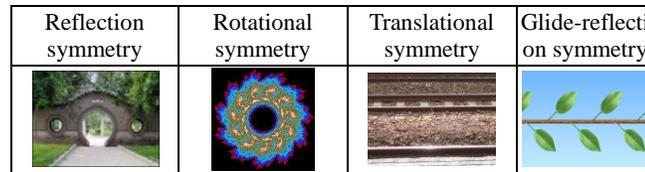
Symmetry Integrated Image Segmentation

2.1. Introduction

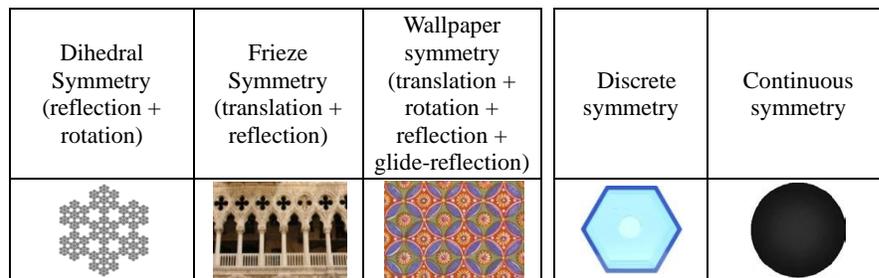
Symmetry is one of the important features that is present in all forms of objects, and it plays a crucial role in machine perception. Symmetry is an intrinsic property of an object which causes it to remain invariant to certain classes of transformations. In the field of computational symmetry, four primitive types of symmetry exist in the two dimensional (2D) Euclidean space [51]: (1) reflection symmetry, (2) rotational symmetry, (3) translational symmetry, and (4) glide-reflection symmetry, a combination of reflection by a line and a translation along that line. Four primitive symmetry types are shown in Fig. 2.1(a). Combinations of the primitive symmetry types generate more symmetry categories [51], as shown in Fig. 2.1(b). This paper is concerned with the segmentation of

2D images having reflection symmetry possessed by many natural and manmade objects.

In the computer vision and pattern recognition literature, symmetry has been used extensively for object boundary interpretation [1, 3], shape symmetry analysis [2, 5, 6, 44-48, 55, 56] and symmetry extraction [7-10, 50, 51, 53, 54, 57]. Since symmetry is a high level geometric feature compared to other lower level features like color and texture, there is an extensive literature concerning application of symmetry into higher level computer vision tasks. Many approaches have been developed for the segmentation and abnormality detection in brain in magnetic resonance images [16-19, 33]. There is also extensive work on face detection [20-23], human tracking and identification [24, 25, 60, 61], and image pattern detection [58, 59].



(a)



(b)

(c)

Fig. 2.1. (a) primitive symmetry categories, (b) combined (extended) symmetry categories, (c) discrete and continuous symmetry.

The above work on symmetry provides us the motivation for integrating symmetry into an image segmentation algorithm. The work in this chapter incorporates high-level

symmetry feature for improved region growing image segmentation. It develops a systematic approach and provides detailed comparisons using publicly available databases.

Symmetry detection can be conducted at a local or a global level [51]. For the global symmetry detection [8, 10, 20, 25, 47], all object points, or the points in the entire image, contribute to the determination of symmetry. The computation of global symmetry is time efficient and always free from prior models, but it is sensitive to distortions. For the local symmetry detection [2, 6, 7, 22, 46, 50], the symmetry element is supported locally by some subset of an object. It is more robust to distortions, but has high time complexity, and generally it relies on prior geometric model. In the field of local symmetry detection, the local features are always used, e.g., the object contour and the gradient orientation. The method of [29] can detect both local and global symmetries, and multiple occurrences of symmetry.

The type of symmetry can be discrete or continuous [27, 51]. Under discrete symmetry group, its invariant transforms (related to its symmetry properties) have discrete (non-continuous) generators, e.g., the reflection symmetry by an axis and the rotational symmetry of a regular polygon. As shown in Fig. 2.1(c), the hexagon possesses discrete rotational symmetry, as only rotations by discrete angles preserve the original appearance. For the continuous symmetry group, its invariant transforms are continuous and smooth, e. g., the rotation of a circle (rotation by infinite number of angles preserves its original shape, as shown in Fig. 2.1(c)).

The existence of symmetry can be measured as a binary (exists or not) or a continuous (variable) feature. The work in [28, 29] treats the symmetry as a continuous feature, in which intermediate values of symmetry denote some intermediate amount of symmetry. Since symmetry in real-world is not perfect, it does not restrict the symmetry as a binary feature, where the object is either symmetric or non-symmetric. The work in this chapter detects the discrete reflection symmetry axis of an image (see Section 2.3.1.1), and it uses a continuous symmetry magnitude to measure the amount of symmetry in an image [29]. Based on the selection of a threshold for symmetry magnitude, the presence/absence of the symmetry axis can be detected.

As mentioned above, the global symmetry detection has the advantages of freedom from *a priori* model. It is considered to be useful in our region-based segmentation scheme. Although these segmentation methods vary in principle on how to form the regions, all of them have one thing in common – they all define a similarity measure related to their segmentation cues, e.g., color and texture. Thus, these methods have the potential to incorporate a symmetry cue. In this chapter, symmetry is combined as a new cue in region growing image segmentation method.

The rest of this chapter is organized as follows. In Section 2.2, we give an overview of the related work on symmetry-based image segmentation and identify our contributions. In Section 2.3, we present the details of technical approach for symmetry-integrated image segmentation. In section 2.4 we provide extensive comparison between symmetry integrated segmentation and other state-of-the-art segmentation methods. Finally, in Section 2.5 we conclude this chapter.

2.2. Related Work and Contributions

2.2.1. Related Work

Image segmentation attracts a great deal of attention in computer vision and pattern recognition. Although regions with coherences like color and texture are segmented successfully, most methods fail to achieve appropriate segmentation due to the unavailability of higher level features. Recently, the integration of symmetry into image segmentation as a high level feature, has attracted attention [12-17, 49], but the field is still immature.

Several reasons make the symmetry-integrated image segmentation a challenging problem. First, symmetry is a higher level feature. It is difficult to be combined with low-level features like color and texture. It makes segmentation a challenging and error-prone task. It is called the feature gap that commonly exists. In this paper, the feature gap is narrowed by using symmetry as a pixel-based affinity [10, 14], and it is integrated into other segmentation cues to form a unified constraint. Second, symmetry features like shape [2, 5, 6, 44-48, 55] are only used for object detection. This paper extends the use of symmetry by applying it as a segmentation cue. Third, there exists a gap between global and local symmetry integrations. Previous work applies local symmetry which segments only the local symmetric objects. Our method uses the global symmetry, which is able to refine the symmetry of the entire segmented image.

The symmetry-based image segmentation can be traced back to the work of [11]. In the current literature, only a limited number of papers can be found for symmetry-based image segmentation [12-17, 49]. Tables 2.1 and 2.2 provide a summary of their methods.

Table 2.1. State-of-the-art Symmetry Based Image Segmentation Methods Integrating Symmetry: Summary and Their Limitations.

Authors	Approaches	No. of limitations in Table 2.2
Liu et al. [12]	Segments a symmetric shape from an image, by Dijkstra's algorithm.	1,2,3,6
Shor et al. [13]	Segments symmetric parts of image by symmetry and color cues.	1,2,3,4
Gupta et al. [14]	Integrates global symmetry into edge weights of the normalized cut segmentation.	3,6,7
Riklin-Raviv et al. [15]	Combines local symmetry into an objective function of the level-set segmentation to segment the boundary of symmetric objects.	2,4,6,7
Jiao et al. [16]	MRI segmentation using symmetry to detect position and boundary of brain tumors.	2,3,6
Saha et al. [17]	Segments MRI using a fuzzy point symmetry based genetic clustering technique.	3,5
Cho et al. [49]	Segments symmetric patterns by matched pairs of local features via symmetry growing.	2,3,6

Table 2.2. State-of-the-art Image Segmentation Methods Integrating Symmetry: List of Their Key Limitations

Limitation Numbers	Key Limitations
1	They concern with local symmetry only [12, 13].
2	They do not segment the whole image. They only extract symmetric objects [12, 13, 15, 16, 49].
3	They only segment image into symmetric parts. Thus, multiple region properties like color and texture, are missed [12, 13, 14, 16, 17, 49].
4	They need prior knowledge or training data [13, 15].
5	They fail to combine symmetry with other cues to build a single segmentation criterion [17].
6	The approaches are sensitive to noise [12, 14, 15, 16, 49].
7	Symmetric regions cannot be refined when the number of segmented regions becomes large [14, 15].

2.2.2. Contributions

As compared to the previous work (see Tables 2.1 and 2.2), the contributions of the paper are:

1. Integrated symmetry and segmentation: This is the first work that integrates the

high-level symmetry concept into the low-level region-based image segmentation method.

2. Global symmetry detection: Our method addresses Limitation 1 (see Table 2.2) by using global symmetry detection, which is more robust to asymmetric distortions.
3. Multi-region segmentation: Limitation 2 and 7 (see Table 2.2) are overcome by region growing as a multi-region segmentation with symmetry (see Fig. 2.5).
4. Integration of symmetry with color and texture: Limitation 3 (see Table 2.2) is addressed by integration with symmetry. Thus, regions with different properties like color, texture and symmetry are segmented simultaneously.
5. No need of prior knowledge: Limitation 4 (see Table 2.2) is addressed by using symmetry affinity, which does not need any prior model (see Equation (2.10)).
6. Different cues into a single criterion: Limitation 5 (see Table 2.2) is overcome by using the symmetry with other constraints as a single criterion (see Equation (2.5)).
7. Robust to distortions: Limitation 6 (see Table 2.2) is overcome by global symmetry detection and symmetry as a continuous feature, that is more robust to distortions.
8. Both quantitative and qualitative analyses: This is the *first work* to use qualitative and quantitative analyses (see Fig. 2.6) in symmetry-integrated segmentation.
9. Segmentation of both symmetric and non-symmetric regions: It not only refines symmetric regions, but also segments non-symmetric regions properly (see Fig. 2.5).

2.3. Symmetry Integrated Region Growing Segmentation

The overall approach is summarized in Fig. 2.2. An input image is processed with

discrete reflection symmetry detection to obtain a global symmetry axis. It is used to compute the symmetry affinity, which is carried forward as the symmetry cue to be integrated into the region growing segmentation. Table 2.3 presents the definition of symbols used in this section.

Table 2.3. Definition of symbols used in Section 2.3.

Symbols	Definitions
(x_{p_i}, y_{p_i})	Two dimensional position of pixel i .
p_i	Symbol for i^{th} pixel.
r_i	Symbol for i^{th} region.
$\delta(p_i, r_j)$	Homogeneity between pixel p_i and neighboring region r_j .
$\delta_S(p_i, r_j)$	Symmetry homogeneity criterion.
$\delta_R(p_i, r_j)$	Region homogeneity criterion. It is a combination of $\delta_{Color}(p_i, r_j)$ and $\delta_{Texture}(p_i, r_j)$.
$\delta_{Color}(p_i, r_j)$	Color homogeneity criterion for pixel p_i and region r_j .
$\delta_{Texture}(p_i, r_j)$	Texture homogeneity criterion for pixel p_i and region r_j .
W_{Color}	Weights of color homogeneity criterion $\delta_{Color}(p_i, r_j)$.
$W_{Texture}$	Weights of texture homogeneity criterion $\delta_{Texture}(p_i, r_j)$.
F_{Color}	Color feature vector.
$F_{Texture}$	Texture feature vector.
C_{p_i}	Symmetry affinity value of pixel p_i .
C_{r_i}	Mean symmetry affinity value for region r_i .
$m(r_i, r_j)$	Region merging criterion for two neighboring regions.
δ_g, δ_m	Thresholds for pixel aggregation and region merging.
$Std_{R-color}$	Region's standard deviations (std.) of color features.
$Std_{R-texture}$	Region's standard deviations (std.) of texture features.
$Gra_{R-color}$	Region's gradient value of std. of color feature.
$Gra_{R-texture}$	Region's gradient value of std. of texture feature.

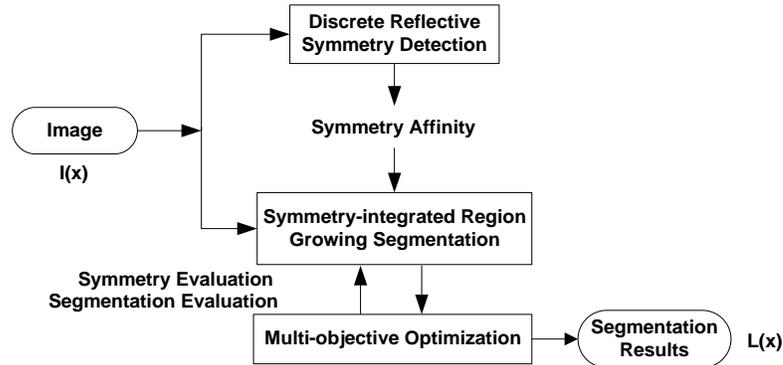


Fig. 2.2. System diagram for symmetry-integrated image segmentation.

2.3.1. Discrete Reflection Symmetry Detection and the Symmetry Affinity Matrix

2.3.1.1. Discrete Reflection Symmetry Detection

The reflection symmetry axis of an image is extracted by the global symmetric constellations of features [29]. The algorithm is capable of finding a dominant symmetry axis, when an image has one or multiple symmetric objects. Table 2.4 shows the key steps of the symmetry detection algorithm.

Table 2.4. The Symmetry Detection Algorithm

Input: the original image.	
Output: the computed global dominant symmetry axis of the image is found; or no axis is found.	
1.	Compute the SIFT feature points [30]. Find the pairs of locally symmetric points from the available SIFT points, by threshold of symmetry magnitude.
2.	Constellate the local symmetric pairs of points into different votes. <i>% The 'constellate' is equal to the unsupervised clustering of the feature points.</i>
3.	Find the set of votes where the number of votes is more than half of the highest vote, <i>% 'half' is a threshold</i>
if	the set is empty <i>% not enough symmetry in the image</i>
Then	no axis is extracted and the algorithm terminates. Set Equation (2.10) to 1. <i>% Algorithm is the same as the region growing without symmetry</i>
if	the set is not empty
Then	examine the votes in this set
if	only one vote remains <i>% one symmetric object in image</i>
Then	Extract the axis corresponding to this vote, as the global dominant axis of the image and terminate the algorithm.
if	multiple votes remain <i>% multiple symmetric objects in the image</i>
Then	Extract the axis corresponding to the highest vote. It is the global dominant axis of the image and terminate the algorithm.

2.3.1.2. The Symmetry Affinity Matrix

The symmetry axis is used to compute a symmetry affinity matrix, which is the correlation between original and the symmetrically reflected image. Each pixel has a symmetry affinity value between 0 (perfectly symmetric) and 1 (totally asymmetric), as shown in Fig. 2.4(d). It is computed by the Curvature of Gradient Vector Flow (CGVF) [10]. The Gradient Vector Flow (GVF) of an image is denoted by:

$$V = [u(x, y), v(x, y)] \quad (2.1)$$

Then, the CGVF is computed as:

$$Curv(x, y) = \frac{1}{|V|^3} [(v_x + u_y)uv - u_x v^2 - v_y u^2] \quad (2.2)$$

where $u_x = \partial u / \partial x$, $u_y = \partial u / \partial y$, $v_x = \partial v / \partial x$, $v_y = \partial v / \partial y$ are the first derivatives of pixel's GVF values along x and y directions. The symmetry affinity of a pixel

(x_{pi}, y_{pi}) is given by:

$$C(x_{pi}, y_{pi}) = \min_{k,v} \left(\sum_{x_{pj-k}=x_{pi}-m}^{x_{pj}+m} \sum_{y_{pj-v}=y_{pi}-m}^{y_{pj}+m} |Curv(x_{pi}, y_{pi}) - Curv(x_{pj-k}, y_{pj-v})| \right) \quad (2.3)$$

where (x_{pj}, y_{pj}) is the symmetric counterpart of (x_{pi}, y_{pi}) reflected by the axis. It is realized by searching local window of pixels with size $2m+1$ centered at the pixel (x_{pj}, y_{pj}) , and the minimum curvature distance is used as the symmetry affinity. The window size is set to $7*7$ ($m=3$) in the experiments. The symmetry affinity value of

Equation (2.3) measures the level of symmetry. In this paper, the level of symmetry quantifies the amount of symmetry exhibited by an image (or a pixel, or a region). The higher symmetry level means that an image is more similar to its mirrored counterpart reflected by the global symmetry axis (see Fig. 2.3(a)). The value of symmetry level of an image (or a region), is the average symmetry affinity value of its pixels (computed by Equation (2.3)). For a pixel, the symmetry level is equal to the pixel's symmetry affinity.

2.3.2. Symmetry Integrated Region Growing

The region growing starts the segmentation from initial *seeds* of pixels, and agglomerates their neighboring pixels having similar features, to form uniform regions iteratively. Our method aims to improve the region growing segmentation by integrating the symmetry cue, using the symmetry affinity matrix obtained from Section 2.3.1.2.

2.3.2.1. Pixel Aggregation Criterion $\delta(p_i, r_j)$

Region growing concerns the aggregation of a region by its neighboring pixels having similar properties measured by the homogeneity criteria, based on color, texture, shape, *etc.* Let us denote it as the homogeneity aggregation criterion $\delta(p_i, r_j)$. The criterion holds true when:

$$\delta(p_i, r_j) < \delta_g \tag{2.4}$$

The rationale behind the equation is that pixel p_i will be aggregated into neighboring region r_j if the region homogeneity criterion $\delta(p_i, r_j)$ between them is below a predetermined region growing threshold δ_g . This threshold can be tuned to allow more or less tolerance to the aggregation criterion, resulting in different segmentations.

Typically, the region homogeneity criteria used are color and texture, with a single region homogeneity criterion $\delta(p_i, r_j) = \delta_R(p_i, r_j)$. In this paper, the aggregation criterion is modified to integrate the symmetry cue, defined as:

$$\delta(p_i, r_j) = \delta_R(p_i, r_j) \delta_S(p_i, r_j) \quad (2.5)$$

Table 2.5. Region Growing Segmentation with Dynamic Region Weights Allocation Algorithm.

<i>Input:</i> Image to be segmented. Pixel_Label=1; Region_Stack=blank;	<i>Output:</i> The segmented regions of the image, with pixel labels. %% pixel label for segmented regions. %% stack of pixels belonging to a region.
1. Search the image row-wise. if there is no pixel unlabeled, finish the segmentation. elseif find an unlabeled pixel Region_stack=blank; %% new region to be grown. Region_stack <- P; %% put pixel P into the region stack. Update mean region features F_{Color} and $F_{Texture}$. $W_{Texture} = W_{color} = 0.5$; %% Initialize feature weights for the region. Label (P) = Pixel_Label; %% label the pixel as grown into region.	
2. For the pixel P, search all its 8-neighbor pixels, as $P_k, k \in [1, 8]$. if Label (P_k)=0; %% unlabeled pixel. Compute criterion δ between pixel P_k and the region in stack, with feature weights $W_{Texture} \cdot W_{color}$ and symmetry integration. %% see Equations (2.5), (2.6) and (2.10). if $\delta < \delta_g$ %% see Equation (2.4) Label (P_k) = Pixel_Label; %% label the pixel. Region_stack_old = Region_stack; Region_stack <- P_k %% grow the pixel into the region. Update region features F_{Color} and $F_{Texture}$. %% Dynamic region weights allocation.	
$W_{Texture_old} = W_{Texture}$; $W_{Color_old} = W_{Color}$; %% Compute gradient of standard deviation of color and texture features of region $Gra_{R-color} = Std_{R-color}(Region_stack) / Std_{R-color}(Region_stack_old)$ $Gra_{R-texture} = Std_{R-texture}(Region_stack) / Std_{R-texture}(Region_stack_old)$ if $Gra_{R-color} < Gra_{R-texture}$ % If std. changes showing more color uniformity than texture: %% Larger weight is put on color to prefer more color uniformity in criterion. $W_{Texture} = W_{Texture_old} \times Gra_{R-color} / Gra_{R-texture}$; $W_{Color} = 1 - W_{Texture}$; elseif $Gra_{R-color} > Gra_{R-texture}$ $W_{Color} = W_{Color_old} \times Gra_{R-texture} / Gra_{R-color}$; $W_{Texture} = 1 - W_{Color}$;	
Set P_k as new pixel P, go to step 2. else Go to step 2, search other unlabeled pixels. else Go to step 2, search other unlabeled pixels. if all neighboring pixels of P is searched and processed by step 2 Region_Stack=blank; %% finish growing of the current region. Pixel_Label=Pixel_Label+1; %% update label for a new region. Go to step 1.	

where we enforce symmetry constraint $\delta_S(p_i, r_j)$ along with the region homogeneity criterion $\delta_R(p_i, r_j)$ to guide the segmentation. The region homogeneity criterion $\delta_R(p_i, r_j)$ is the combination of color and texture cues, which will be introduced in Section 2.3.2.2. The symmetry constraint $\delta_S(p_i, r_j)$ is introduced in Section 2.3.2.3.

2.3.2.2. Region Homogeneity Criterion $\delta_R(p_i, r_j)$

The region homogeneity criterion $\delta_R(p_i, r_j)$, is given by:

$$\delta_R(p_i, r_j) = W_{Color} \delta_{Color}(p_i, r_j) + W_{Texture} \delta_{Texture}(p_i, r_j) \quad (2.6)$$

where $W_{Texture} + W_{color} = 1$. The weights W_{Color} and $W_{Texture}$ can be allocated in a dynamic manner, depending on whether a region shows more uniformity in color or texture, as described in the dynamic weights allocation with the region growing algorithm shown in Table 5. For a region R , let the standard deviation of its pixel-level color and texture feature distributions ($Std_{R-color}$ and $Std_{R-texture}$) denote its region uniformity. At each region growing iteration, a region absorbs one pixel, and the region's color and texture uniformities are changed as more pixels are aggregated. The algorithm is able to dynamically track the changes of color and texture uniformities, and assign weights to put more emphasis on color or texture as the region growing process is iterated. The larger weight will be assigned to the feature whose region uniformity is increased (with the decreased standard deviation).

We use HSV as the color feature [37]. It is composed of a vector that is a nonlinear transform of HSV values:

$$F_{Color}(\cdot) = (V \cdot S \cdot \cos(2\pi H), V \cdot S \cdot \sin(2\pi H), V) \quad (2.7)$$

where H , S and V correspond to HSV components of a pixel or average for a region. The color homogeneity criterion in Equation (2.6) can be expressed as:

$$\delta_{Color}(p_i, r_j) = \|F_{Color}(p_i) - F_{Color}(r_j)\| \quad (2.8)$$

which is the Euclidean distance of color features between pixel p_i and its neighboring region r_j .

The 8-dimensional texture feature $F_{Texture}$ is obtained by: (1) filtering an image with a bank of Gabor filters at 4 orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$), and (2) computing the mean and standard deviation of the filtered image or region. The texture feature of a pixel is extracted from its local window. Thus, the texture homogeneity criterion is:

$$\delta_{Texture}(p_i, r_j) = \|F_{Texture}(p_i) - F_{Texture}(r_j)\| \quad (2.9)$$

Both color and texture features are normalized into [0, 1].

2.3.2.3. Symmetry Homogeneity Criterion $\delta_S(p_i, r_j)$

The motivation of using symmetry constraint $\delta_S(p_i, r_j)$ is as follows: If both the pixel p_i and its neighboring region r_j are symmetric with their counterparts (both have low symmetry affinities), they will decrease the criterion δ_S , by which the pixel will more likely to be grown into the region to form an integrated symmetric shape. The symmetry constraint $\delta_S(p_i, r_j)$ in Equation (2.5) is given below:

$$\delta_s(p_i, r_j) = \frac{\frac{\pi}{2} + \arctan(\sqrt{(1+C_{pi})(1+C_{rj})})}{\pi} + \frac{1 + |\sqrt{C_{pi}} - \sqrt{C_{rj}}|}{2} \quad (2.10)$$

where C_{pi} and C_{rj} are symmetry affinities of pixel p_i and its neighboring region r_j .

This equation is non-linearly related to the symmetry affinity values. This constraint is developed for estimating whether pixel p_i can be grown into region r_j by the symmetry criterion. Equation (2.10) provides the following symmetry constraints: the first term means that if both patterns i and j indicate low symmetry affinities (highly symmetric) to their symmetric counterparts i' and j' , as seen in Fig. 2.3(a), pixel i is more likely to be grown into region j by decreasing $\delta_s(p_i, r_j)$. The second term means that the two patterns with closer values of symmetry affinities, will also reduce $\delta_s(p_i, r_j)$. As a result, the criterion $\delta_s(p_i, r_j)$ has a lower value under the two conditions given below,

- (a) Symmetry affinities of pixel i and region j have lower values;
- (b) Symmetry affinity values of pixel i and region j are closer with each other.

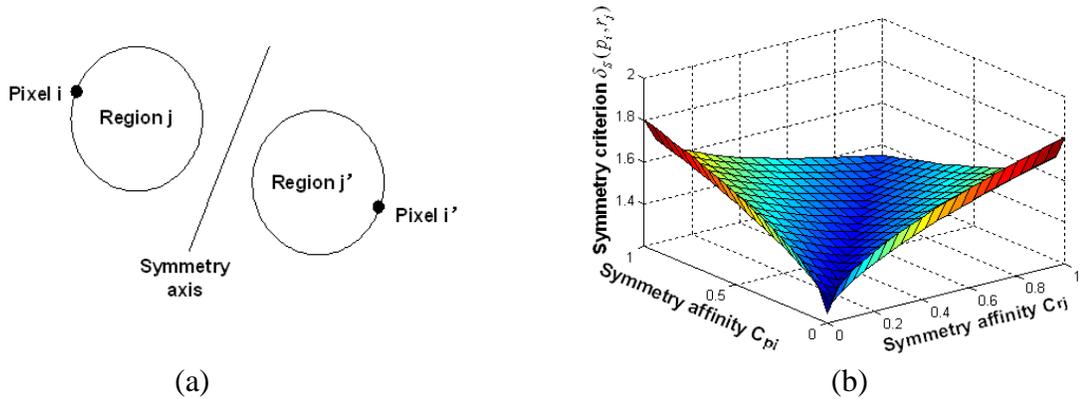


Fig. 2.3. (a) Integration of symmetry in region growing, (b) graphic illustration of Equation (2.10): plot of symmetry criterion $\delta_s(p_i, r_j)$ related to a pair of symmetry affinity values C_{pi} and C_{rj} .

The above relationship is explained by a plot of $\delta_s(p_i, r_j)$ in Fig. 2.3(b). It is clear that the lowest value of $\delta_s(p_i, r_j)$ is reached when both symmetry affinity values C_{p_i} and C_{r_j} have 0 values (both of them stay in perfect symmetric field). Consequently, both the lower and closer symmetry affinity values of the two patterns will lead to a lower value of the criterion $\delta_s(p_i, r_j)$. The lower value of symmetry criterion $\delta_s(p_i, r_j)$ will decrease the overall segmentation criterion $\delta(p_i, r_j)$ (see Equation (2.5)). Thus, the criterion $\delta(p_i, r_j)$ is more likely to pass the threshold δ_g (see Equation (2.4)). This means that patterns i and j in a more symmetric field are easier to be grown into an integrated symmetric region, and at the same time eliminate many small noisy regions within symmetric objects. Work in [14] also uses a symmetry criterion integrated into an edge weight in the graph-cut image segmentation method [39], and its limitations are stated in Table 2.1. Experimental results in Section 2.4.5 provide an analysis which will show the advantages of our method over that of [14].

2.3.2.4. Symmetric Region Merging Criterion $m(r_i, r_j)$

Initial segmentation by the aggregation criterion $\delta(p_i, r_j)$ (see Equation (2.5)) is an over-segmented result. During the region merging, neighboring regions are merged using the criterion $m(r_i, r_j) = \|F_{Color}(r_i) - F_{Color}(r_j)\| + \|F_{Sym}(r_i) - F_{Sym}(r_j)\|$, which is the Euclidean distances of mean color and mean symmetry affinity values of two regions r_i and r_j . A region with higher symmetry level with its symmetric counterpart, is more likely to be merged into neighboring region. For the two thresholds δ_g (Equation (2.4))

and δ_m (Section 2.2.4.1), related to the aggregation criterion $\delta(p_i, r_j)$ and region merging criterion $m(r_i, r_j)$, we establish a 2D parameter space of the two criteria, that is used for segmentation optimization (Section 2.3.4).

2.3.3. Performance Evaluations of Segmentation and Symmetry

In this paper, three evaluation schemes are used for estimating the segmentation and symmetry, as given below.

2.3.3.1. The Unsupervised Segmentation Evaluation

We use the following metric for unsupervised segmentation evaluation [38], and it is defined as,

$$EVA_SEG_{unsupervised} = 1 - \frac{1}{M \times N} \left(1 + \log(\sqrt{NR})\right) \sum_{i=1}^{NR} \left[\frac{e_{SEG}^2(r_i)}{1 + \log(N_i)} \right] \quad (2.11)$$

where M, N are the number of rows and columns of an image, and NR is the total number of segmented regions. The term $e_{SEG}^2(r_i)$ is the inter-region contrast of region r_i :

$$e_{SEG}^2(r_i) = \left(\sum_{j=1}^{N_i} \left\| F_{Color}(p_j) - \bar{F}_{Color}(r_i) \right\| + \left\| F_{Texture}(p_j) - \bar{F}_{Texture}(r_i) \right\| \right) / N_i \quad (2.12)$$

where $\|F_{Color}(j) - \bar{F}_{Color}(R_i)\|$ is the Euclidean distance of HSV color features between pixel p_j and its region r_i (mean HSV), and $\|F_{Texture}(p_j) - \bar{F}_{Texture}(r_i)\|$ is the Euclidean distance of texture features derived by Gabor filters. N_i is the number of pixels of i th region. Lower inter-region contrast indicates a better segmentation. $(1 + \log(\sqrt{NR}))$ and

$(1 + \log(N_i))$ are a punishments for over-segmentation and small segments, respectively.

The second term in the right side of Equation (2.11) is normalized within $[0, 1]$. The larger values of Equation (2.11) are for better segmentation. In this paper, segmentation results of Caltech-101 [42] database are optimized by unsupervised evaluation.

2.3.3.2. The Supervised Segmentation Evaluation

The supervised segmentation evaluation [41] is used as,

$$EVA_SEG_{supervised} = \frac{M_I + m \times \eta}{1 + m} \quad (2.13)$$

where M_I is the region matching evaluation term,

$$M_I = \sum_{j, \max_i Card(r_i^{Ref} \cap r_j^{Seg})} \frac{Card(r_i^{Ref} \cap r_j^{Seg})}{Card(r_i^{Ref} \cup r_j^{Seg})} \rho_j \quad (2.14)$$

$Card(\cdot)$ computes the number of pixels of a region. For the segmented region r_j^{Seg} , its reference region r_i^{Ref} is chosen from the ground-truth segmentation, with the maximum overlap with r_j^{Seg} . Larger the overlap means a better segmentation. The normalization term ρ_j is given by,

$$\rho_j = \frac{Card(r_j^{Seg})}{Card(I^{Seg})} \quad (2.15)$$

where I^{Seg} is the segmentation of the entire image. The term η in Equation (2.13) is a punishment for both over-segmentation and under-segmentation,

$$\eta = \begin{cases} NR_{Ref} / NR_{Seg} & NR_{Seg} > NR_{Ref} \\ \log(1 + NR_{Seg} / NR_{Ref})? & otherwise \end{cases} \quad (2.16)$$

Where NR_{Seg} (NR_{Ref}) is number of regions in real segmentation (ground-truth/reference segmentation). In conditions of both over-segmentation and under-segmentation, the above term decreases. m in Equation (2.13) is the weight parameter, set to 0.5 for all the experiments, that means to put the weight on punishment term for over-segmentation that is half of the weight of the region matching term. The larger the value of $EVA_SEG_{supervised}$ the better the segmentation is. The supervised evaluation requires the ground-truth segmentation, which prevents its wide application. In this paper, the segmentation results of UCB database [43] (with ground truth benchmark) is optimized and analyzed by the supervised evaluation.

2.3.3.3. The Symmetry Evaluation

In this paper, a new symmetry evaluation of a segmented image is defined as:

$$EVA_SYM = 1 - \frac{1}{NR} \sum_{i=1}^{NR} e_{SYM}^2(r_i, r_i) \quad (2.17)$$

For symmetry evaluation of Equation (2.17), NR is the number of segmented regions, and $e_{SYM}(r_i, r_i)$ is the difference in region properties between region r_i and its symmetric counterpart region r_i according to the symmetry axis. The region properties used are: region's centroid, mean color value, and its orientation. For each region r_i , the smaller $e_{SYM}(r_i, r_i)$ means that the region r_i is more symmetric to its counterpart r_i .

The second term in the right side of Equation (2.17) is normalized within $[0, 1]$. A larger value of Equation (2.17) is for the better. But the symmetry axis detection (Section 2.3.1.1) cannot be optimized by these thresholds. The thresholds for symmetry detection (see Table 2.4) are fixed for all the results shown in this paper.

2.3.4. Multi-objective Optimization for Segmentation and Symmetry

It is able to search the segmentation results with optimal performance for both segmentation and symmetry. It is formulated as a multi-objective optimization (MOP), which is the process of optimizing multiple objectives subject to certain constraints. We use Non-dominated Sorting Genetic Algorithm (NSGA-II) [31], a multi-objective optimization algorithm to search for optimum matched segmentation parameters (δ_g and δ_m), by using measures of the objective functions of segmentation and symmetry (see section 2.3.3). Our optimization problem (see Fig. 2.2) can be formulated as follows: given an image $I(x)$, the system outputs a segmentation $L(x)$, with a combinatorial objective function $F(L(x))$, composed of evaluations of both segmentation and symmetry as Equations (2.11) or (2.13), and (2.17):

$$F(L(x)) = \begin{bmatrix} EVA_SEG(L(x)) \\ EVA_SYM(L(x)) \end{bmatrix} \quad (2.18)$$

where $EVA_SEG_{xx}(L(x))$ is Equation (2.11) or (2.13). The goal is to get a segmentation $L(x)$ where both segmentation and symmetry are optimized. It's formulated as a Multi-Objective Optimization (MOP) defined below:

By searching the parameter space, seek an optimal segmentation result $L_0(x)$ from all

possible results $L(x)$ in segmentation space Ψ , such that:

$$F(L_0(x)) = \arg\max_{L(x) \in \Psi} F(L(x)) \quad (2.19)$$

It aims to seek a segmentation that optimizes both the segmentation and symmetry performance $F(L(x))$, along with its optimal parameter of thresholds (δ_g and δ_m) for aggregation criterion $\delta(p_i, r_j)$ and region merging criterion $m(r_i, r_j)$.

Table 2.6. The Overall Algorithm for the Proposed Method.

<p>Input: the original image. Output: the segmented regions of the image.</p> <p>Global Symmetry Detection</p> <ol style="list-style-type: none"> 1. Extract SIFT interesting points from the image; 2. Compute global symmetry axis by symmetric pairs of SIFT points; <ul style="list-style-type: none"> <i>if</i> symmetry axis is detected <i>if</i> multiple symmetry axes are detected <ul style="list-style-type: none"> Extract the axis belonging to the dominant symmetric object. Compute symmetry affinity by Curvature of GVF (see Eq. (2.3)). <p>Symmetry-integrated Region Growing Segmentation</p> <ol style="list-style-type: none"> 4. Compute region homogeneity criterion δ_R: color criterion by HSV basis, texture criterion by Gabor filters (Eq. (2.6)); 5. Compute symmetry criterion δ_S by (Eq. (2.10)). <ul style="list-style-type: none"> <i>if</i> symmetry axis is detected <ul style="list-style-type: none"> Use Eq. (2.10) to compute δ_S by symmetry affinity. <i>else</i> $\delta_S = 1$. 6. Combine δ_R and δ_S as a single criterion δ (Eq. (2.5)), with its threshold δ_g (see Section 2.4.1 and Eq. (2.4)); 7. Run the region growing by the integrated criterion δ (see Table 2.5); 8. Use HSV color space and symmetry to compute region merging criterion $m(r_i, r_j)$, with its threshold (see Section 2.4.1); 9. Perform region merging by the merging criterion; finish segmentation. <p>Segmentation Optimization</p> <ol style="list-style-type: none"> 10. Evaluate segmentation and symmetry by Eq. (2.11) (2.13) (2.17); 11. Multi-objective optimization by NSGA-II, in parameter space δ_g and δ_m, using the following rules: <ul style="list-style-type: none"> <i>if</i> segmentation and symmetry performance from step 9 are <i>acceptable</i> (see Section 2.4.1) <ul style="list-style-type: none"> End the optimization, finish segmentation; <i>if not acceptable</i> <ul style="list-style-type: none"> Search different parameter setting of δ_g and δ_m, Jump to step 4, and run the segmentation and optimization again.

2.3.5. Algorithm for the Proposed Segmentation Method

The overall algorithm for the system is given in Table 2.6.

2.4. Experimental Results

In this section we present both quantitative and qualitative analysis to demonstrate the improvements in image segmentation by the integration of symmetry. The symmetry-integrated region growing is compared to the region growing [34] without the symmetry integration. Thus, the segmentation improvement is carried by the symmetry integration alone. Our method shows superior performance over other commonly used segmentation approaches [35, 36, 39]. Moreover, our method also outperforms the symmetry-integrated normalized cut [14].

2.4.1. Datasets and Parameters

The proposed method was tested on two commonly used image databases, demonstrating different levels of object symmetries. The two image databases used are:

(a) *The Caltech-101 image database* [42]. It contains images of both natural and manmade objects belonging to 101 categories. Segmentation results are shown in Fig. 2.5. They are optimized using unsupervised segmentation evaluation (without ground-truth) of Equation (2.11).

(b) *The Berkeley segmentation dataset and benchmark* (UCB) [43]. It contains hand-labeled (ground-truth) segmentations of 1000 Corel dataset images. Example images and their delineated ground-truth segmentations are shown in Fig. 2.6. The segmentation results on this dataset are optimized using supervised segmentation evaluation (with ground-truth) of Equation (2.13).

The parameter space for segmentation optimization is composed of 2 thresholds: the

aggregation criterion threshold δ_g , introduced in section 2.3.2.1, and the region merging criterion threshold δ_m . The value for δ_g varies between [0.015, 0.035] and the range for δ_m is [0.02, 0.05]. These ranges are obtained by experiments and they are unchanged. The multi-objective optimization [31] is run on the search space of these two parameters, with objective functions of both symmetry and segmentation evaluations introduced in section 2.3.3. The optimization stops if the results are *acceptable* as follows: (a) Both segmentation and symmetry performances are better than the pre-defined thresholds (0.62 and 0.89 for segmentation and symmetry, respectively). The values are set based on our experimental experience; (b) The combination of the performance reaches its optimal value reported by NSGA-II [31]. The optimization stops with the optimal segmentation if both conditions are met, otherwise it continues by searching different parameters until maximum number of iterations (equals to 500 in this paper) is reached.

2.4.2. Performance Metrics

Three performance metrics are used in experiments.

(a) Performance curve of supervised segmentation measurement of Equation (2.13), with respect to the symmetry measurement of Equation (2.17) on the UCB database [43], as in subplot (g) in Fig. 2.4 and subplots (i) in Fig. 2.6.

(b) The ROC plot, a plot of true positive versus false positive of the region pixels (with respect to ground-truth segmentation), is shown in subplots of (j) in Fig. 2.6.

(c) *Optimal* segmentation obtained by (i) supervised evaluation (Equation (2.13)) with

ground-truth segmentation, for the UCB database, as shown in segmentations in Figs. 2.4, 2.6, 2.7-2.9 and 2.11-2.12, or by (ii) unsupervised evaluation (Equation (2.11)) without ground-truth segmentation, of Caltech-101 database, as for segmentations in Figs. 2.5 and 2.10. In both (i) and (ii) the evaluations are also optimized by symmetry evaluation of Equation (2.17).

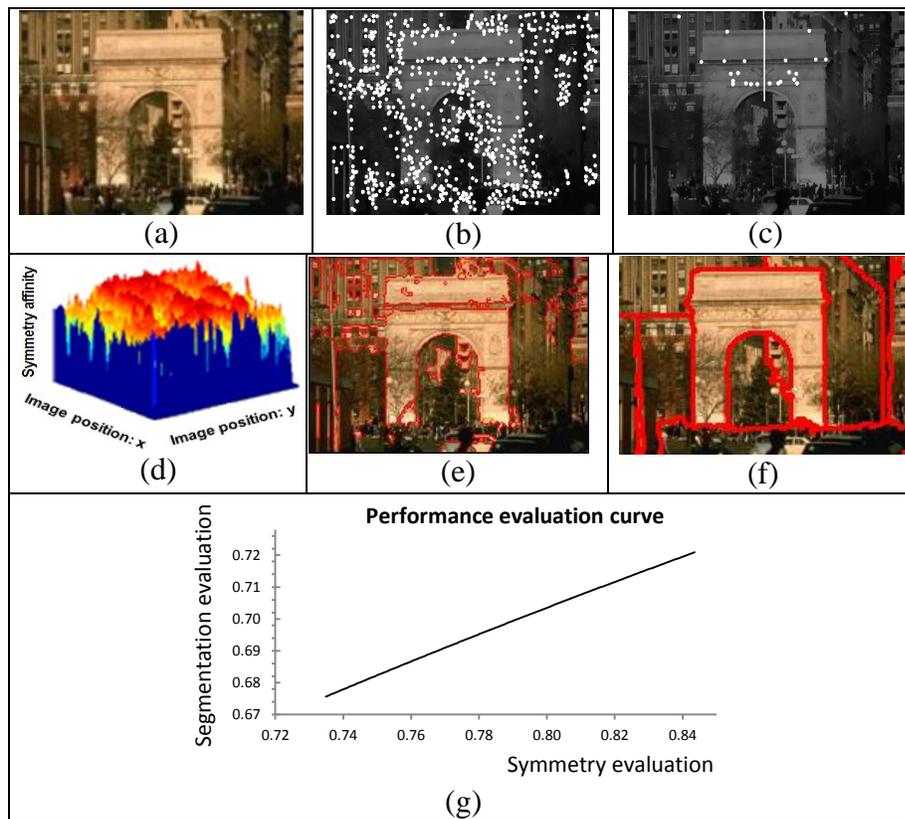


Fig. 2.4. Symmetry-integrated image segmentation using the image from UCB dataset [43]: (a) original image, (b) SIFT points, (c) symmetry axis, (d) symmetry affinity of image, (e) symmetry-integrated segmentation, (f) the ground-truth segmentation provided by UCB dataset [43], (g) performance curve of segmentation and symmetry.

2.4.3. Performance of the Proposed Method

2.4.3.1. Realization of the Proposed Method

In Fig. 2.4, we show our segmentation by an image of the symmetric ‘Triumphal

Arch' [43]. Fig. 2.4(d) shows large symmetry affinity values in red pseudo-color, which indicates asymmetric pixels, and small values in yellow, indicating symmetric pixels. Fig. 2.4(g) shows the performance curve, measured by Equations (2.17) and (2.13). Different points on the curve correspond to evaluations of segmentation and symmetry, by running the segmentation using different parameters. The segmentation and symmetry are improved simultaneously. Other symmetry-integrated segmentation are shown in Fig. 2.5. Please refer to Fig. A-3 in the Appendix for more results.

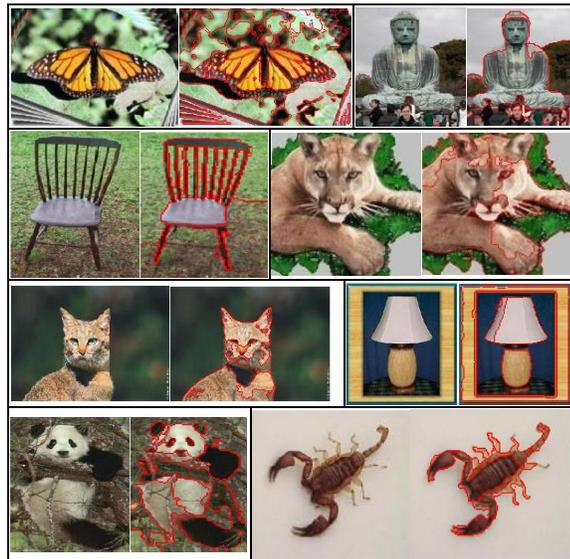


Fig. 2.5. Symmetry-integrated segmentation using images from the Caltech-101 [42].

2.4.3.2. Symmetry-integrated Region Growing vs. Region Growing No Symmetry

In the curves of subplots (i) in Fig. 2.6, also in Figs. A1-A2 in Appendix, the black curve and the dotted green curve are the performance of symmetry-integrated region growing segmentation and the region growing without symmetry, respectively. The only difference in the two methods is the integration of symmetry. Comparison between the two performance curves shows the following two advantages of symmetry-integration.

(1) The overall segmentation performance is improved compared to the regular region growing, and the improvement comes only from the integration of symmetry.

(2) In regular region growing, its segmentation performance does not improve (subplots (i) of image ‘Man’ in Fig. 2.6), even starts to decrease (subplots (i) of image ‘Building’ in Fig. 2.6), with the improvement of symmetry. But the segmentation on black curve still improves at high symmetry evaluation scores.

Table 2.7. Principles of State-of-the-Art Segmentation Methods.

Current methods	Principles	Parameter Space (Thresholds)
Region Growing [34]	Grows neighboring pixels into the seeds to form the segments.	(1) Pixel aggregation; (2) Region merging.
Normalized cut [39]	Partitions the image into segments by minimizes the edge weights.	Number of segmented regions.
Normalized cut -symmetry [14]	Combines symmetry into the regular normalized cut segmentation [39].	Number of segmented regions.
Watershed [36]	Pixels with highest magnitude in the gradient form a segment.	Region merging.
Meanshift [35]	Performs mean shift filter on pixel, and merges windows to form regions.	(1) Filter bandwidth; (2) Region merging.

Lack of segmentation improvement with the increase in symmetry is due to the over-segmentation. It deteriorates the segmentation, but symmetry still improves since small symmetric regions are segmented. Our method solves this problem by segmenting symmetric objects into complete regions. So the over-segmentation is overcome and a high symmetry evaluation score (by Equation (2.17)) is retained. The ROC curve of subplots (j) in Fig. 2.6 (and Figs. A1-A2 in Appendix) shows that our method has higher true positive than the one without symmetry. Table 2.8 shows the segmentation improvement from no symmetry to symmetry integration. The largest improvement of 8.39% comes from the

image ‘Fresco’, with a large symmetric object. Numerous small regions are eliminated by the symmetry cue, as compared in Fig. A-2(c) and (d) of ‘Fresco’, in the Appendix.

Table 2.8. Numerical Comparison of Segmentation Performance: Images in Fig. 2.6, and Figs. A1-A2 in Appendix [52].

Images in UCB dataset	Comparison: proposed method			Comparison: symmetry-based normalized cut [14]			Watershed [36]	Meanshift [35]
	With symmetry	No symmetry	% improvement	With symmetry	No symmetry	% improvement		
Building (Fig. 2.6)	75.48%	72.57%	+2.60%	69.99%	68.36%	+2.38%	74.62%	63.37%
Man (Fig. 2.6)	72.58%	71.67%	+1.27%	66.42%	65.01%	-2.48%	67.29%	62.83%
Woman_1 (Fig. 2.6)	71.44%	70.57%	+1.23%	68.76%	68.13%	+0.92%	66.52%	61.28%
Vase ([52])	76.70%	76.42%	+0.37%	69.13%	69.01%	+0.17%	68.34%	61.03%
Bear ([52])	75.82%	73.70%	+2.88%	71.29%	71.17%	+0.17%	72.84%	67.90%
Woman_2 ([52])	73.75%	73.29%	+0.63%	73.13%	72.84%	+0.40%	71.92%	67.45%
Butterfly ([52])	76.73%	75.36%	+1.86%	61.64%	60.71%	+1.53%	68.36%	71.65%
Fresco ([52])	82.42%	76.04%	+8.39%	76.30%	76.57%	-0.35%	77.58%	46.41%

2.4.3.3. Results on Images with Different Symmetry Levels – Region Growing with/without Symmetry

The segmentation results obtained through images with different levels of symmetry can be used to show the efficacy of the proposed method. The symmetry level in Fig. 2.7(e) is measured by the average symmetry affinity value of the image, and it is quantified into six categories. The segmentation performance is measured by the supervised evaluation (see Equation (2.13)), same for the results in Figs. 2.8-2.9 and 2.11-2.12. The segmentation performance improvement (see Fig. 2.7(f)) by using symmetry (see Fig. 2.7(c)), compared from the same method without symmetry (see Fig. 2.7(d)), indicates that images with higher symmetry level achieve a larger segmentation improvement. With the absence of symmetry (see images (1) and (2) in Fig. 2.7), no

symmetry axis is detected. Thus, the symmetry constraint (see Equation (2.10)) is set to 1, and the performance is the same as the one without symmetry.

2.4.3.4. Results on Images with Symmetry Distortion – Effect of Occlusion, Affine/Perspective Transform, Articulation, and Incorrect Symmetry Detection

- **Occlusion:** Many of the real world images have symmetric objects with occlusions. Fig. 2.8 shows segmentation with symmetric objects occluded by trees. The symmetry axis can be detected effectively (see Fig. 2.8(b)). Under partial occlusions, the symmetry integration (see Fig. 2.8(d)) can improve the segmentation (see Fig. 2.8(f)), compared from the same method without symmetry (see Fig. 2.8(e)).
- **Affine/Perspective Transform:** Fig. 2.9 shows the symmetry integration under distortions. The affine transform shown in Fig. 2.9(1) is composed of linear transformations (rotation, scaling or shear) and a translation. It preserves the parallelism of lines. The perspective transform shown in Fig. 2.9(2) illustrates that from the view of human eyes (or camera), the parts of the object in the distance appear smaller than the parts close by. The perspective transform preserves the straight lines of objects. Fig. 2.9(b) shows that the symmetry axes for transformed faces are extracted, and the symmetry integration can improve the segmentation (see Fig. 2.9(f)) under non-rigid distortions.

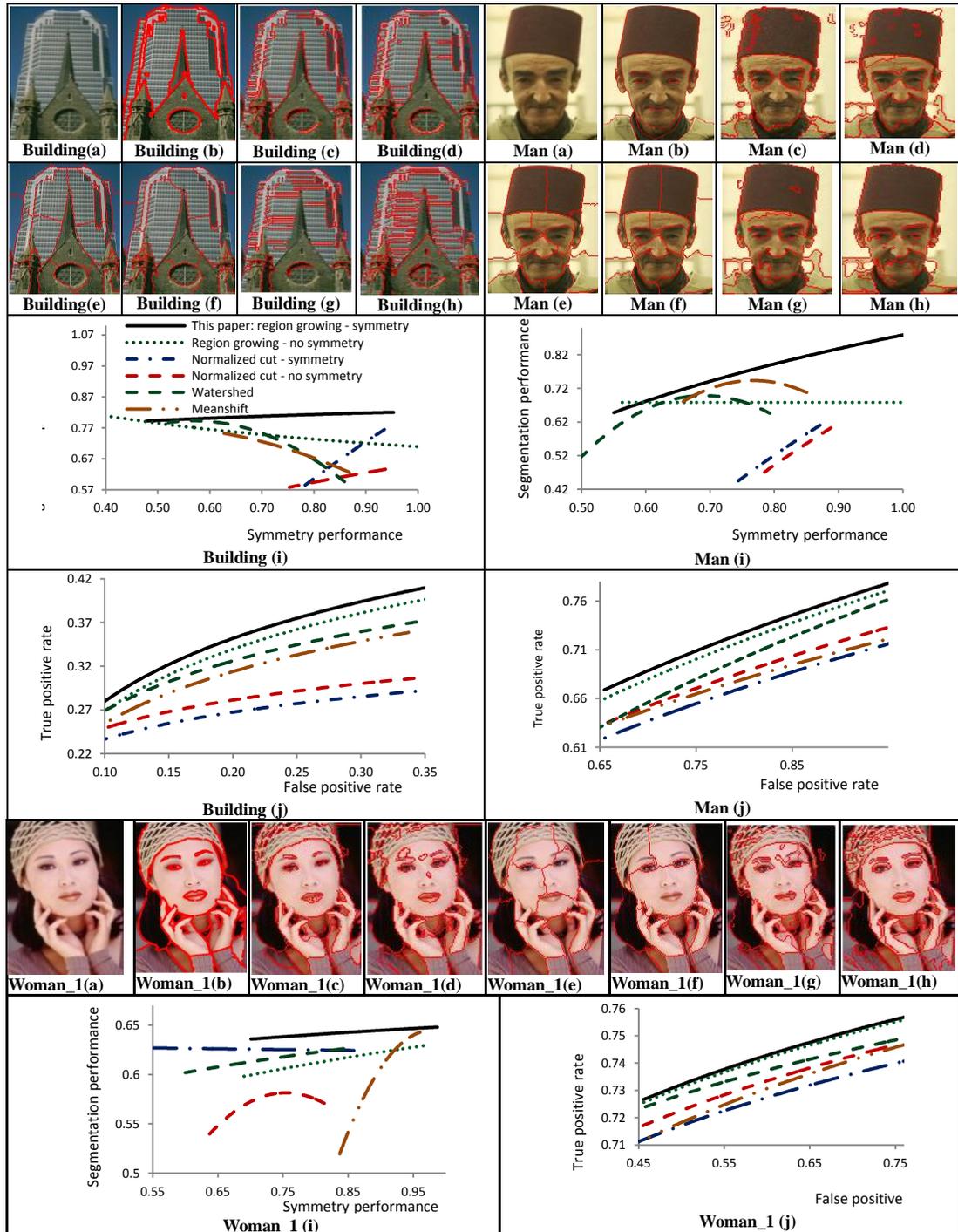


Fig.2.6. Comparison of results on UCB database [43]: ‘Building’, ‘Man’ and ‘Woman_1’. (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) normalized cut with symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves.

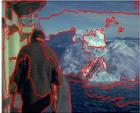
	(a)	(b)	(c)	(d)	(e)	(f)
(1)					N/A	+0%
(2)					N/A	+0%
(3)					0.827 very Low	+0.91%
(4)					0.615 low	+1.03%
(5)					0.535 medium	+1.26%
(6)					0.572 medium- high	+2.06%
(7)					0.350 high	+2.47%
(8)					0.174 very high	+2.54%

Fig. 2.7. Results for images, with different symmetry levels, from the UCB database [43], (a) original image, (b) ground-truth segmentation provided by UCB database, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) symmetry level, (f) segmentation improvement (from (d) to (c)). N/A: Not Applicable.

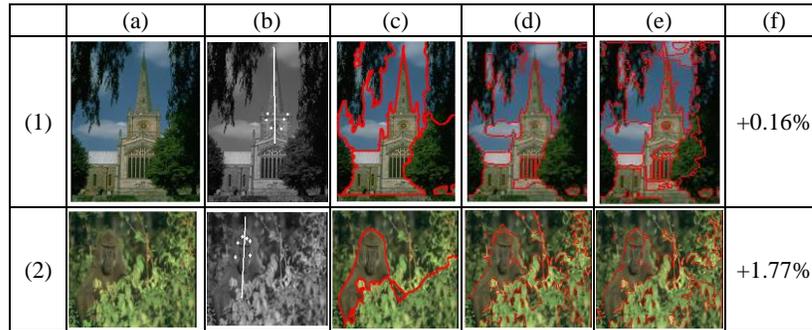


Fig. 2.8. Images with occluded symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis detection, (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d)).

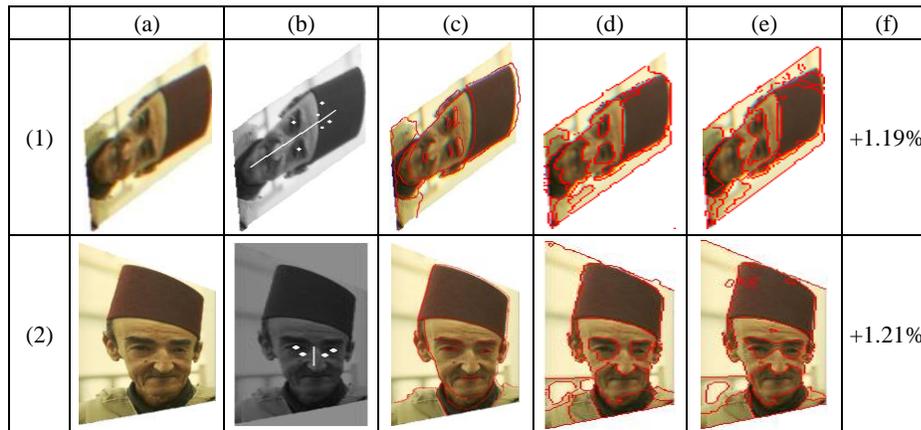


Fig. 2.9. Image 'Man' in Fig. 2.6, with (1) affine transform, (2) perspective transform, from UCB database [43], (a) transformed image, (b) symmetry axis, (c) ground-truth segmentation provided by UCB database, (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d)).

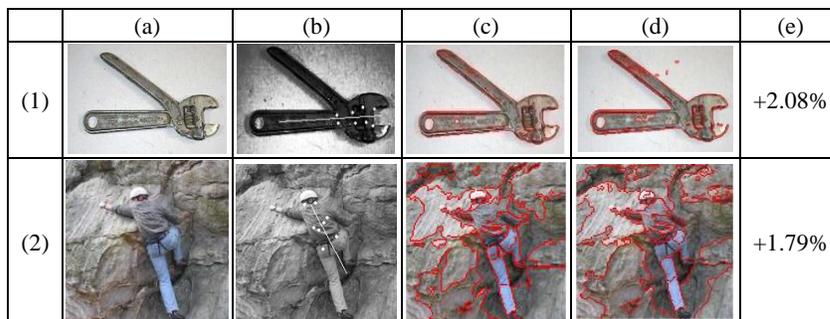


Fig. 2.10. Results for images, with articulated symmetry distortions, from the Caltech-101 database [42] (image (1)), and from the Internet (image (2)), (a) original image, (b) symmetry axis detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)).

	(a)	(b)	(c)	(d)	(e)	(f)
(1)					Perspective distortion	+0.27%
(2)					Occlusion	+0.43%
(3)					Articulation	+0.24%

Fig. 2.11. Results with images for incorrect symmetry detection, from the UCB database [42], (a) original image, (b) symmetry detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) distortions, (f) segmentation improvement ((d) to (c)).

- **Articulation:** The articulation refers to the object composed of two or more joint components, and each component has rigid movement. Fig. 2.10 shows how the symmetry integration improves segmentation of the images with articulated symmetry distortions. Since images in Fig. 2.10 are collected from the Caltech-101 database or from the Internet, without the ground-truth segmentation, we use Equation (2.11) for the unsupervised segmentation evaluation. Image (1) shows the clamp with asymmetric handles, and image (2) shows a human with articulated arms and legs. Fig. 2.10(b) shows that global symmetric axes are correctly extracted. Fig. 2.10(e) indicates the segmentation improvements achieved by using the symmetry integration.

- **Incorrect Symmetry Detection:** Fig. 2.11 shows the incorrect symmetry axis extraction, because of large distortions for perspective, occlusion and articulation, respectively. In these three conditions, Fig. 2.11(f) shows that the performance of symmetry integrated segmentation is no worse than that of the same method without

symmetry. The conclusion is that even under incorrect or failed symmetry detection, the symmetry integration is not worse than that of using no symmetry at all.

	(1)	(2)	(3)
(a)			
(b)			
(c)			
(d)			
(e)			
(f)	+1.67%	+4.66%	+6.87%

Fig. 2.12. Images with multiple symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis (with high intensity as the dominant axis), (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement ((e) to (d)).

2.4.3.5. Results on Images with Multiple Symmetric Objects

Complex conditions of symmetry exist in images with multiple symmetric objects. Within multiple symmetry objects shown in Fig. 2.12, the global symmetry detection is able to extract multiple symmetry axes in an image (see Fig. 2.12(b)), and choose the

symmetry axis belonging to the most dominant symmetric object, as the global symmetry axis of the image. The dominant symmetric objects in images (1) and (2) in Fig. 2.12 are both the rightmost objects, and their symmetry axis (in bright color) is used as the global symmetry axis of the image. Another condition of symmetry is shown as image (3), where all the three astronauts contribute to a same symmetry axis, and they share the same cluster of global symmetric pairs of SIFT points. Image (3) highlights the advantage of using the global symmetry detection, which can detect symmetry within the entire image, and make use of multiple symmetric objects to derive a global axis. It cannot be done by using local symmetry detection only. Fig. 2.12(f) shows that under condition of multiple symmetric objects, the symmetry integration also improve the segmentation, compared to the same method without symmetry.

2.4.4. Symmetry-integrated Region Growing vs. Other Segmentation Methods

2.4.4.1. Qualitative Comparison

We obtain image segmentation improvements as compared to other segmentation methods that do not exploit symmetry. The principles of currently popular image segmentation methods compared are shown in Table 2.7. In Fig. 2.6 (and Figs. A1-A2 in Appendix) we demonstrate the segmentation improvements by symmetry integration, using eight example images from UCB database with ground-truth segmentations provided. The segmentation results are optimized by NSGA-II and measured using both the supervised performance evaluation of Equation (2.13) and the symmetry evaluation of Equation (2.17).

Results (d)-(h) in Fig. 2.6 (and Figs. A1-A2 in Appendix) have different levels of segmentation defects and noisy regions in symmetric objects, compared to symmetry-integrated segmentation in (c). The incorporation of symmetry cue is the main source of improvement. The symmetric regions are more likely to be aggregated by the symmetry constraint, by eliminating small noisy regions within the symmetric objects, thus more complete and proper symmetric boundaries are generated. The most complete and clear symmetric objects are segmented by the proposed method. For the result (c) of image ‘Man’ in Fig. 2.6, our approach can segment the symmetric face without incorrect segments, while the other results fail to accomplish so. Similar improvement can be seen in image ‘Building’ in Fig. 2.6, where the central part of the building is segmented with less flaws and noisy regions than other methods. One of other advantages of our method is that we not only refine symmetric regions, but also segment background non-symmetric regions more properly.

2.4.4.2. Quantitative Comparison

The subplots (i) in Fig. 2.6 (and Figs. A1-A2 in Appendix) show the curves of symmetry versus segmentation performances, measured by supervised segmentation evaluation of Equation (2.13) and symmetry evaluation of (2.17), respectively. Each point in the curve is a symmetry and segmentation performance by running segmentation of an image by different parameter values. From comparisons in subplots (i), following conclusions can be made:

- (1) The curve of the proposed method has the highest segmentation performance in all images.

(2) The curve of the proposed method also reaches the highest symmetry performance measures.

The above improvements of segmentation and symmetry, comes from integrating the symmetry cue to improve the segmentation by refining both the symmetric objects and non-symmetric backgrounds. Subplots (j) in Fig. 2.6 (and Figs. A1-A2 in supplemental material) show the ROC plot, and our method has the highest true positive rate. The ROC plot quantitatively shows that the proposed method is closest to the ground-truth segmentation. Table 2.8 shows the comparison among segmentation performances (Equation (2.13)) measured on the optimal segmentation results. All segmentations are optimized by NSGA-II. The proposed method has the highest performance in all images.

4.5 Symmetry-integrated Region Growing vs. Current Symmetry-Based Segmentation

We also compare our approach with the method in [14], which is a symmetry-integrated segmentation combining symmetry feature into regular normalized cut segmentation to refine the symmetry level of the segmented regions. As we can seen in subplots (i) in Fig. 2.6 (and Figs. A1-A2 in Appendix), both normalized cut with and without symmetry, have worse segmentation performance than region growing with and without symmetry, and they also have lower symmetry measurement. We can infer from the scalar comparisons in Table 2.8 that, the symmetry-integrated region growing reaches higher segmentation improvements than [14]. Take the image ‘Bear’ in Table 2.8 as an example, the improvement from normalized cut to symmetry-integrated normalized cut is only 0.17%, while the improvement from regular region growing to the symmetry–integrated region growing is high as 2.88%. For an extreme case of ‘Fresco’ in

Table 2.8, the performance obtained by symmetry integrated normalized cut is even decreased by 0.35%, while the improvement of region growing by symmetry integration is high as 8.39%. Also for the ROC curves (subplots (j)) of all 3 images in Fig. 2.6, the true positive of symmetry-integrated normalized cut is even worse than that of normalized cut with no symmetry. In conclusion, the symmetry integrated in normalized cut does not always improve the segmentation. The symmetry integrated in region growing improves the segmentation in all cases, and it reaches higher improvement compared to [14]. The normalized cut separates perceptually coherent region into many parts in large number of segments. It prevents the work of [14] with segmentation improvement.

Table 2.9. Numerical Comparison of Optimal Segmentation Performance: Supervised vs. Unsupervised Evaluations

Images in UCB dataset (Fig. 2.6, and Figs. A1-A2 in Appendix)	(a) Optimal segmentation obtained by <i>supervised</i> evaluation (Eq. (2.13))		(b) Optimal segmentation obtained by <i>unsupervised</i> evaluation (Eq. (2.11))	
	(1) Segmentation performance (Eq. (2.13))	(2) Symmetry performance (Eq. (2.17))	(3) Segmentation performance (Eq. (2.13))	(4) Symmetry performance (Eq. (2.17))
Building	75.48%	97.26%	70.33%	96.17%
Man	72.58%	98.48%	71.62%	98.43%
Woman_1	71.44%	98.79%	70.74%	97.66%
Vase	76.70%	96.02%	73.19%	96.95%
Bear	75.82%	99.27%	73.80%	98.24%
Woman_2	73.75%	95.44%	72.75%	93.50%
Butterfly	76.73%	86.10%	66.02%	81.78%
Fresco	82.42%	72.48%	68.44%	67.53%

2.4.6. Symmetry-integrated Region Growing: Supervised vs. Unsupervised Evaluations

Since two different segmentation evaluation criteria (Equations (2.11) and (2.13)) are used in this paper, in this subsection, the effectiveness of these two evaluations are compared as shown in Table 2.9, on eight images from the UCB database (see Fig. 2.6

and Figs. A1-A2 in Appendix). Note that segmentation of images from the UCB database, are optimized by the supervised evaluation (Equation (2.13)), and segmentation of images from the Caltech-101 database, are optimized by the unsupervised evaluation (Equation (2.11)). But in this subsection, the segmentation of images from the UCB database is optimized by both Equation (2.13) and (2.11) to compare the results of the two evaluation criteria, by the following steps:

(1) In column (a) of Table 2.9, segmentation is optimized with the supervised segmentation evaluation (Equation (2.13)). The goodness of the optimized segmentation is evaluated using Equation (2.13) (see column (1) in Table 2.9). The 2nd column in Table 2.8 has the same realization.

(2) In column (b) of Table 2.9, segmentation is optimized with the unsupervised evaluation (Equation (2.11)). The goodness of the optimized segmentation is also evaluated by Equation (2.13) (see column (3) in Table 2.9).

(3) The symmetry performance shown in columns (2) and (4) are both evaluated by Equation (2.17).

It is clear from Table 2.9 that the optimal segmentation results obtained by the supervised evaluation, are closer to the ground-truth segmentation, with higher evaluation score than that obtained by unsupervised evaluation (see comparison between columns (1) and (3)). Thus, the supervised evaluation is preferred to guide the optimization for a better segmentation, if the ground-truth is available.

2.4.7. Statistical Validation of Results

The proposed method is validated by statistical results with 15 images from the UCB database, and with 93 images from the Caltech-101 database (see these images listed in Figs. A4-A5 the Appendix). Symmetry axes are detected correctly in all the 108 images. Table 2.10 shows the comparison of statistical results on images from the two databases. Note that the mean and standard deviation are computed from optimal segmentation performances of the images. We use the supervised performance evaluation (see Equation (2.13)) for the UCB database, but use unsupervised evaluation (see Equation (2.11)) for the Caltech-101. Table 2.10 shows that the proposed method outperforms all the other methods. The percentage of improvement in parenthesis with the positive number, in the last five rows, in Table 2.10 is the segmentation improvement achieved by the proposed symmetry integration method, compared to the method in the same cell. The performance, in the parenthesis in the second row in each cell, are the highest and lowest performance of the method, respectively. Note that even a 1% numerical improvement in segmentation leads to a significant visual improvement in segmentation results.

All the 108 images (with correct symmetry axis detected) achieved performance improvement by using the symmetry cue (see Table 2.10). Additionally, we also tested our algorithm on 374 images (from Caltech-101 database) in which the symmetry axes are incorrectly detected. In this situation, still over 99.45% of the images obtained improved segmentation performance by using the symmetry cue. There are only two exceptional cases as shown in Fig. 2.13 where the improvement did not take place.

However, the decrease in performance is minimal in these two exceptional cases. With other 598 images (from Caltech-101 database) where no symmetry axes are detected (not enough symmetry level in images), the performance of the proposed method is the same as the one without using symmetry, for all these images. In conclusion, the proposed method has robust performance as evidenced by experiments on large image datasets.

Table 2.10. Statistical Validation on 15 Images from UCB Database, and on 93 Images from Caltech-101 Database (See Figs. 2.4-2.5 in Supplemental Material [52] for Images)

	UCB Database		Caltech-101 Database	
	Mean segmentation performance	Standard deviation of segmentation performance	Mean segmentation performance	Standard deviation of segmentation performance
Region growing - with symmetry	76.54% (87.56%, 75.93%)	4.31%	83.26% (91.01%, 67.41%)	6.17%
Region growing - no symmetry	72.53% (+ 5.53%) (83.78%, 68.55%)	4.57%	76.29% (+ 9.13%) (82.87%, 59.70%)	6.30%
Normalized cut - with symmetry	67.83% (+ 12.84%) (81.61%, 62.09%)	4.79%	72.60% (+ 15.80%) (81.04%, 63.51%)	6.74%
Normalized cut - no symmetry	66.42% (+ 15.24%) (76.42%, 61.84%)	4.90%	70.39% (+ 18.28%) (77.63%, 61.96%)	6.39%
Watershed	69.73% (+ 9.77%) (80.11%, 57.94%)	6.16%	68.51% (+ 21.53%) (74.92%, 58.18%)	6.33%
Meanshift	61.07% (+ 25.33%) (75.30%, 44.23%)	6.54%	64.03% (+ 30.03%) (73.46%, 45.00%)	6.82%

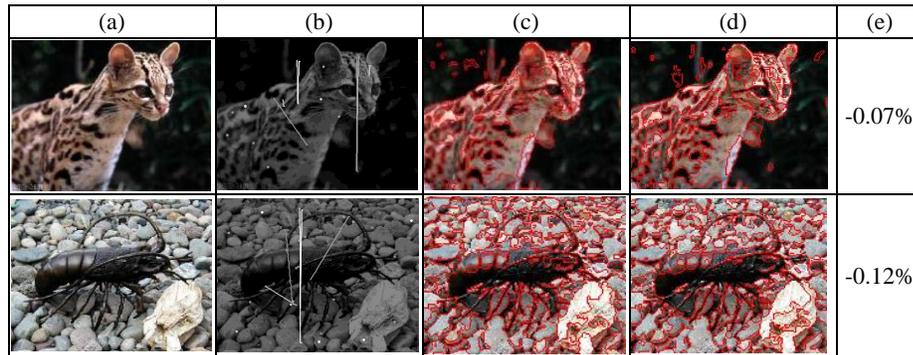


Fig. 2.13. Results with decreased segmentation performance by using symmetry, from the Caltech-101 database [42], (a) original image, (b) symmetry axis, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)). Note that the symmetry axes are incorrectly detected.

2.4.8. Discussion of the Results

Based on the experimental results on hundreds of images shown here and in [52], we note the following points:

1. Quality of segmentation: The symmetry constraint generates more symmetrical regions, which decreases the number of small segments. Due to the robustness against noise property of the global symmetry and symmetry affinity, noisy regions are aggregated into surrounding regions if they show symmetry property.

2. Different levels of symmetry: The higher the symmetry presents in an image, the higher is the improvement for symmetry-integrated image segmentation.

3. Symmetry axis: The proposed method highly depends on the symmetry axis detection. But under condition of incorrect symmetry detection (see Fig. 2.11) and no symmetry detected (see images (1) and (2) in Fig. 2.7), the performance of the proposed method is not worse than that of the method without symmetry (see Section 2.4.7).

4. Symmetry refinement: It is possible to use the segmented regions that are symmetric with their reflected regions to provide a feedback to the symmetry detection algorithm for the computation of a refined axis of symmetry. This, in turn, will provide a better image segmentation.

2.5. Conclusions

In this paper, a new symmetry integrated scheme is proposed for region based image segmentation to improve its performance. We accomplish this goal by incorporating symmetry into the region growing segmentation, in terms of the symmetry affinity matrix. We carry out experiments on a wide variety of images and provide thorough analysis.

Both qualitative and quantitative experimental results indicate that with the symmetry constraints enforced by symmetry affinity, both the symmetry and segmentation performance are improved compared to several popular current segmentation methods. This is the *first* paper in the computer vision and pattern recognition field that demonstrates the improvement of pixel-level image segmentation by incorporating the high-level symmetry cue and performing thorough qualitative and quantitative analyses on large datasets. The non-optimized code takes ~54 sec. to run (for a 640×480 color image) on a PC with Intel Core 2 Quad CPU 2.40GHz and 3GB of RAM. The region growing segmentation takes 87% of the total running time. The future work will focus on increasing the computational efficiency of the method.

Chapter 3

Automated Symmetry-Integrated Brain Abnormality Detection in MRI Sequences

3.1. Introduction

Magnetic resonance imaging (MRI) is a medical imaging technique most commonly used in radiology to visualize the structure and function of a body. It provides detailed images [85] of a body in any plane with a higher discrimination than other radiology imaging methods such as computed tomography (CT), single photon emission computed tomography (SPECT), etc. The goal of this paper is to find automatically find the selected subsets of pixels/voxels within a dataset, that are characterized as abnormal (injured or diseased) regions/volumes in the brain MR images. We call these regions as the

regions-of-interest (ROIs).

The research on automated methods for the evaluation of magnetic resonance (MR) images has its origin in the manual ROI labeling which is typical of radiological treatments. Specifically, mining of brain ROIs that appear in an MRI sequence has been an important task in image processing, image analysis and pattern recognition fields [62-70, 75]. This can assist health care professionals with automated computer techniques for diagnosis and appropriate treatment. Traditionally, the boundary of a ROI in a MR image is manually traced. The computer-aided diagnosis on brain MRI reduces the manual workload. An efficient ROI detection algorithm is important for diagnosis, planning, and treatment. However, there is a fuzziness in pattern recognition. It is caused by the partial volume effect, in which a single voxel represents more than one type of tissue, mainly caused by the process of transforming a continuous observation of the real world into an image with a finite resolution and limitations of MRI techniques leading to the loss of high definition details of a tissue. Due to these effects, there exist specific kinds of vagueness such as blurred boundaries, variabilities caused by movement of the patient's head, and inter-image illumination and positions of brain, which cause the propagation of detection errors. ROIs of certain diseases, for example, incipient tumors and subtle lesions in themselves have low contrast compared to the neighboring tissues. They are difficult to recognize even by experts. These problems still exist even after registration, normalization and use of prior brain tissue models. As a result, it is very challenging for the automated ROI detection methods to outline ROIs as close to the manual annotations obtained by experts.

Recently, with a rapid progress of image processing and analysis techniques, a variety of computer-aided applications including both automated and semi-automated image segmentation and detection methods are being considered for clinical use that can significantly reduce the time needed to make such methods practical. Previous studies [62-67], based on both 2D analysis (only on data of single 2D MRI slice) and 3D image analysis (on data from all the MRI slices in a sequence that contains 3D information of the brain), detect brain ROIs reasonably well by using training sets and prior models of tissues as well as using efficient preprocessing techniques, like registration [67, 71, 75]. Numerous features such as image intensity, texture, shape, 3D volume, etc, are used in model matching schemes. However, locating a specific ROI by these methods requires a variety of training and specific prior knowledge of statistical distributions of tissues.

In this paper, reflection *symmetry* is integrated with image analysis as a *new* kind of high-level feature. This integration allows a fully automated brain ROI detection method, without prior models, and is applicable to a wider range of MRI data with different patient ages and ROI characteristics. Symmetry is an intrinsic property of an object which causes it to remain invariant to certain classes of transformations. Familiar 2D symmetry categories are [51]: (1) reflection symmetry, (2) rotational symmetry, (3) translational symmetry, where the objects remain invariant with reflection (by an axis), rotation (with certain angle centered at a point), and translation (with specific distance or direction), respectively, and (4) glide-reflection symmetry, a combination of reflection by a line and a translation along that line. The 3D reflection symmetry of a 3D brain object refers to a 2D plane with which a brain is symmetric by a mirrored reflection. For a 2D brain MRI

slice, the 2D reflection symmetry refers to an axis along which the brain is highly symmetric. This paper detects 2D reflection symmetry for brain MRI slices only. However, the 3D information from multiple 2D MRI slices is used to aid in the automated ROI detection. This is realized by a comparison among the neighboring slices to filter out noisy non-ROIs. Additionally, the proposed method aims to detect almost all kinds of abnormal regions, including the tumors, lesions and the injury regions.

The remainder of this chapter is organized as follows. Section 3.2 gives an overview of the related work and the contributions of this paper. Section 3.3 provides the technical details of the approach. Section 3.4 gives experimental results with both qualitative and quantitative analysis. Finally, Section 3.5 provides the conclusions of the paper.

3.2. Related Works and Contributions

3.2.1. Related Works

There exist many challenges associated with precise automated detection of brain ROIs. From the pathology point of view, the brain ROIs are generally different in size and shape, and may appear at any location with different image intensities. Some ROIs also deform other normal and healthy tissue structures. Due to imaging procedures, three problems commonly exist in the quality of MRI slices. They are described in Table 3.1.

Table 3.1. Three Problems Related to the Automated ROI Detection Methods

Problem#	Problem Definitions
(1)	Blurred boundaries between different tissues and structures caused by the loss of resolution and contrast during the collection and digitization of a MR image.
(2)	Movement artifacts caused by patient's head movement, during prolonged scanning, result in recording errors, and the brain is not aligned among MRI slices.
(3)	Variations in different MRI slices due to the changes in the imaging environment. It represents an additional challenge for image pre-processing.

Problem (1) is usually solved by incorporating the prior models [62-67] to enhance the discrimination among brain tissues. When boundaries of different tissues (including the ROI) are too blurred to be extracted by low-level image features (like gray scale and image gradient), high-level tissue models are needed to obtain a satisfactory tissue classification/extraction. *Problems (2) and (3)* are mostly solved by image registration [67, 75] which performs the alignment among MRI slices, for both positions and illuminations. However, the registration needs complex algorithms and also prior models. Generally, the above solutions are not automated and always require large amounts of pre-processing and user interaction.

State-of-the-art ROI extraction techniques use mainly two kinds of methods: tissue classification/segmentation and abnormality extraction. The tissue classification [62, 63] approach starts with brain segmentation based on a prior tissue model and extracts ROIs from classified clusters. Unfortunately, in order to obtain satisfactory classification results, large amounts of training data or complex prior models are required. The domain of training phase which is only used for a specific category of ROI only, strictly restricts the range of applications. The abnormality extraction approach [64, 65] generally builds a stochastic model for normal brain tissues, and simultaneously detects abnormalities that do not fit the model. However, it is challenging to build a complete prior model in order to cover enough tissue information. Another abnormality extraction method is called digital subtraction [66, 79], which tracks structure or volume changes of brain imaged at different time intervals. The accuracy of subtraction relies highly on registration among different MRI sequences [67, 71, 75]. As a result, most of the current ROI extraction

methods depend heavily on the quality of preprocessing and prior knowledge, and most importantly, they are not fully automated.

Table 3.2. State-of-the-art ROI Detection Methods for MRI: Summary, Their Advantages (+) and Limitations (-).

Authors	Principle of Techniques	Comments
Birgani et al., [62]	Brain MRI segmentation and ROI classification by fuzzy c-means (FCM).	+ Unsupervised classification, and no prior model is needed. -Need to set the no. of classes for FCM, need the distribution of ROI.
Bhanu et al., [78], Bianchi [86]	Recursive region splitting and merging.	+ Automatic segmentation and ROI detection can be automated. - Results include many non-ROIs.
Kabir et al., [63]	Lesion detection MAP segmentation and digital atlas.	+ Segmentation separates different brain models simultaneously. - Need multiple prior models.
Leemput et al., [64]	Lesion detection using a stochastic model for normal brain images.	+ Brain tissues are segmented effectively. -The prior model is needed.
Cuadra et al., [65]	Brain lesion detection by lesion growth.	+ The lesion boundary is outlined. - Need the prior model for the lesion; registration is also needed.
Ratan et al., [81]	Tumor detection on multi-parameter analysis and watershed segmentation.	+ Watershed segmentation can segment MRI. - The tumor detection mislocates non-tumor as tumor regions.
Anbeek et al., [82]	Lesion detection using KNN classification and probabilistic segmentation.	+ The KNN classifier, with a training model, can separate lesions. - Need to train the KNN classification model.
Hojjatoleslami et al., [83]	Segmentation of large brain lesions by region growing.	+ Region growing from inside the lesion outlines to lesion boundary. - Need to locate the seed inside the lesion manually.
Corso et al., [84]	Tumor segmentation by weighted aggregation and Bayesian classifier.	+ Can hierarchically recognize different models within a single tumor region. - Need the prior classification model.
Chen et al., [92]	MRI brain image segmentation by graph cut approach.	+ The graph cut can outline tissue boundary properly. -The ROI segments have to be located by the user.
Rathi et al., [93]	Segmentation of brain MRI by HSOM and wavelet features.	+ Segmentation by hierarchical self organizing map (HSOM). - The vector quantization lowers the image resolution; need prior model.

As a result, state-of-the-art ROI detection methods (summarized in Table 3.2) generally fail to solve *problems (1)-(3) automatically*, as they use only low-level features such as image intensities, edges and gradients, or otherwise employ predefined tissue models. The incorporation of reflection *symmetry*, which exhibits the difference between ROIs and normal brain tissues as a high-level feature, can avoid the registration and prior

models to obtain a good ROI detection accuracy and to make detection fully automated. Research on asymmetry of the brain can be traced back to [89], and it has been investigated previously [75, 68-69, 70, 88] in ROI detection. Its advantages and limitations compared to the proposed method are listed in Table 3.3. The first six methods listed in Table 3.3 also deal with *problems (1)-(3)* using symmetry, but they either require registration and prior model, or they do not use symmetry as effectively as our method.

Table 3.3. State-of-the-art Symmetry-based ROI Detection Methods for MRI: Summary, Their Advantages (+) and Limitations (-).

Authors	Principle of Techniques	Comments
Saha et al., [75]	Brain MRI segmentation using a fuzzy point symmetry based genetic clustering technique.	+ Clustering by point symmetry based distance rather than Euclidean distance; no <i>a priori</i> info. - Time consuming; copes with internal symmetry within a region.
Bergo et al., [68]	MRI segmentation based on the texture symmetry.	+ Free from a template; good generality. - Not robust to changes in parameters; uses only local symmetry.
Ray et al., [69]	Locate brain abnormality by bounding box around it using the symmetry analysis.	+ No registration; no training image; real-time implementation. - Need the reference (template) image; boundary for abnormal regions is not well outlined.
Khotanlou et al., [70]	ROI detection using segmentation, asymmetric region extraction and fuzzy classification.	+ ROIs are included in the asymmetric regions. -The asymmetric regions include too many noisy non-ROI regions due to ineffective image segmentation method.
Thirion et al., [89]	Asymmetry measure of brain structures, by 3D vector field operators.	+ The asymmetry measure is automatic. - The ROI is not precisely located; only abnormal dissymmetry is analyzed.
Li et al., [91]	Tumor segmentation using symmetry of human brain and level set method.	+ Non-tumor tissues are filtered by volumetric symmetry analysis. - The initial level set boundary of tumor has to be set one by one.
This paper	Detect brain ROIs by symmetry integration in several steps for segmentation, region extraction and classification.	+ Integrates symmetry in all steps; no prior model; no training data; good generality; effective segmentation algorithm; uses global symmetry rather than local or internal symmetry. - Very low contrast ROIs are missed.

3.2.2. Contributions

The proposed method not only overcomes *problems (1)-(3)* (described in Section 3.2.1) to a great extent by integrating reflection symmetry instead of registration and prior models, but it also overcomes the limitations of the other symmetry-based ROI

detection methods as shown in Table 3.3 by using symmetry more effectively.

We formulate the idea of using symmetry based on the observation that for most abnormality detection methods, though different in principle, there is a common criterion that abnormal ROIs are detected by their properties that deviate from the expected normal and healthy tissue properties. Specific to our case, since most of the ROIs are asymmetric with their mirror regions against the axis of reflection symmetry for a MRI slice while the other healthy brain structures are highly symmetric, we are able to detect ROIs by integrating symmetry into the image analysis. According to this characteristic, ROIs are within the asymmetric regions. Therefore, asymmetry is regarded as a distinct property of ROIs that deviates from other normal symmetric tissues. Furthermore, multidimensional (3D) region-based symmetry analysis tends to be more robust to noise and other interferences. By an integration of symmetry, we overcome the limitations of the other symmetry-based approaches and this paper makes the contributions shown in Table 3.4.

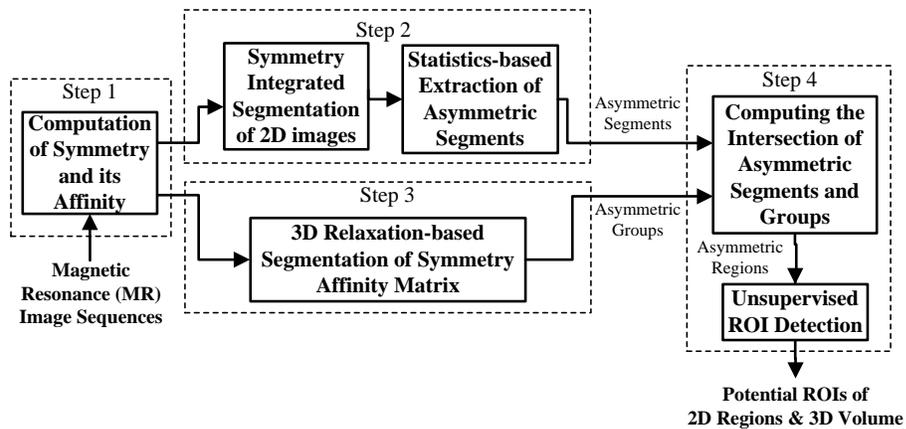


Fig 3.1. System diagram for symmetry integrated region-of-interest (ROI) detection in MRI sequences.

Table 3.4. Contributions of the Proposed Method

Contributions	Comments
<i>Problem (1)</i> : the blurred boundary and random location of ROIs (see Table 3.1), is solved by using the symmetry automatically, without prior models.	Asymmetric regions can be extracted automatically by symmetry-based image segmentation followed by asymmetric region extraction. In this process, symmetric tissues are discarded by a symmetry measure, and no prior model is needed to classify those tissues (see Sections 3.3.2, 3.3.3 and 3.3.4).
<i>Problem (2) and (3)</i> (see Table 3.1) are solved by symmetry automatically, without registration.	The symmetry affinity computed by edge information of gradient vector flow (see Sections 3.3.1) is invariant to the illumination variability among MR slices. So registration is not required.
Compared to the other symmetry-based work in Table 3.3, limitation of local symmetry is overcome by global symmetry extraction.	Global symmetry information is relatively more robust to local noise and asymmetric distortions, and it is relatively much more invariant in rotation, translation, and scale compared to local symmetry (see Section 3.3.1).
Limitation of noisy regions is overcome by symmetry-based image segmentation [73].	Symmetry-based segmentation can segment the brain tissue much better such that there are less noisy regions (see Section 3.3.2), as compared to other segmentation methods in Table 3.3.
Symmetry is used in different forms in multiple steps.	Extensive use of the symmetry information makes the method fully automated (see Fig. 3.1).
3D symmetry information from 2D MRI slices is proposed.	3D symmetry information from 2D MRI slices is used in both the asymmetric region extraction (see Sections 3.3.2, 3.3.3 and 3.3.4) and the final classification of the asymmetric regions into the ROI (see Section 3.3.4).

3.3. The Technical Approach

The flowchart of the overall approach is shown in Fig. 3.1. In our approach, symmetry is used in multiple steps in different forms, to filter out normal regions, and detect the abnormalities. In step 1, a symmetry affinity matrix is obtained, which is used as a measurement of symmetry level in later steps. In step 2, reflection symmetry is integrated into a segmentation algorithm such that most of the symmetric parts in an image are segmented appropriately. This prevents misclassification of symmetric parts into asymmetric regions in a later step. Subsequently, in step 2, statistical characteristics (kurtosis and skewness) of the symmetry affinity matrix (step 1) are computed and they are used to extract asymmetric segments from segmented regions. In step 3, symmetry affinity matrices for different MRI slices are segmented in 3D using an improvement of a gradient relaxation method [74] by the incorporation of 3D information from 2D MRI

slices. In step 4, first the results from steps 2 and 3 are fused to obtain refined asymmetric regions, and then an unsupervised classifier is used to extract ROIs from the asymmetric regions.

3.3.1. Symmetry Extraction

Normal human brains possess a high degree of *reflection symmetry* in 3D (also called *bilateral symmetry*) although they are not perfectly symmetrical. The procedure of symmetry extraction (see step 1 in Fig. 3.1) consists of two major components: (a) computation of symmetry axis, and (b) computation of symmetry affinity matrix.

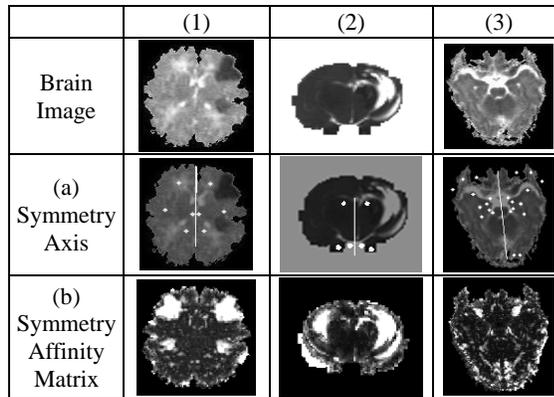


Fig 3.2. Sample results of symmetry extraction for injured brains of (1) human, (2) animal, and (3) human.

- **Computation of Symmetry Axis:** The symmetry axis of the brain is defined as the plane that best separates the two hemispheres of the symmetric brain structure. We use the global symmetric constellations of features [84] to detect a reflective symmetry axis presented in MR brain images (slices) in 2D. Pairs of local reflective symmetric points are first selected from the available SIFT feature points [84], and they are further transformed into the linear Hough space to vote for a dominant symmetry axis, as shown in Fig. 3.2(a). The orientation and scale invariance property of SIFT points enables the

successful detection of the symmetric pairs with rigid distortions. As a result, the global symmetry axis of brain under the conditions of head movement and variability of imaging environment, can be extracted without registration.

- **Computation of Symmetry Affinity Matrix:** Once the global symmetry axis is detected, it is utilized to compute a symmetry affinity matrix. This matrix computes the relation between the original image and its symmetrically reflected image, and it plays an essential role to enforce the high-level symmetry integration into low-level image segmentation. The value of each element in an affinity matrix represents a pixel's symmetry level when compared to its symmetric counterpart reflected along the axis of symmetry. A pixel that is perfectly symmetric is valued 0 and the totally asymmetric pair of pixels is valued 1. Each pixel has a continuous symmetry affinity value between 0 and 1, as shown in Fig. 3.2(b). Symmetry affinity is measured by using both the Curvature of Gradient Vector Flow (CGVF) [10, 14] and the gray scale difference. Let us denote the Gradient Vector Flow (GVF), V , of a pixel (p_x, p_y) as:

$$V = [u(p_x, p_y), v(p_x, p_y)] \quad (3.1)$$

where u and v are two dimensional gradient vector flow values of a pixel. The value of V is computed by minimizing an energy function as stated in [90], and the resulting GVF matrix represents the image contour. Then, the CGVF is calculated as:

$$Curv(V) = \frac{1}{|V|^3} [(v_x + u_y)uv - u_x v^2 - v_y u^2] \quad (3.2)$$

and $u_x = \partial u / \partial x$, $u_y = \partial u / \partial y$, $v_x = \partial v / \partial x$, $v_y = \partial v / \partial y$ are the first derivatives of

pixel's GVF values along the x and y directions, respectively. For a pixel (p_{x_i}, p_{y_i}) , its symmetry affinity by curvature is given by:

$$C_{-curv}(p_{x_i}, p_{y_i}) = \min_{k,q} \left(\sum_{\substack{p_{x_j}+m \\ p_{x_{j-k}}=p_{x_j}-m}} \sum_{\substack{p_{y_j}+m \\ p_{y_{j-q}}=p_{y_j}-m}} \left| \text{Curv}(p_{x_i}, p_{y_i}) - \text{Curv}(p_{x_{j-k}}, p_{y_{j-q}}) \right| \right) \quad (3.3)$$

where (p_{x_j}, p_{y_j}) is the symmetric counterpart of (p_{x_i}, p_{y_i}) reflected by the axis. It is realized by searching in a local window of pixels with size $2m+1$ centered at the pixel (p_{x_j}, p_{y_j}) , and finding the minimum curvature distance as the symmetry affinity. The window size is set to $5*5$ ($m=2$) in experiments. If these two points located at mirroring sides of an axis are within a symmetric field, the symmetry affinity of the pixel will be smaller. The final symmetry affinity value is defined by,

$$C(p_{x_i}, p_{y_i}) = C_{-curv}(p_{x_i}, p_{y_i}) \cdot C_{-gray}(p_{x_i}, p_{y_i}) \quad (3.4)$$

where the first term $C_{-curv}(p_{x_i}, p_{y_i})$ on the right side is given by Equation (3.3) and the second term $C_{-gray}(p_{x_i}, p_{y_i})$ is the gray scale difference between pixel (p_{x_i}, p_{y_i}) and its reflected symmetry pixel (p_{x_j}, p_{y_j}) . The final symmetry affinity value of equation (3.4) is normalized between 0 and 1 for all pixels in a symmetry affinity matrix. A sample symmetry affinity matrix of brain MRI is shown in Fig. 3.2(b), where brighter parts indicate asymmetric pixels, which include potential ROIs. The edge information in CGVF is more reliable than the gray scale value in computation of symmetry affinity.

3.3.2. 2D Segmentation and Extraction of Asymmetric Segments

Image segmentation is performed and the segmented regions are used to extract asymmetric segments. The procedure (see step 2 in Fig. 3.1) consists of two parts: (a) symmetry integrated region growing image segmentation to obtain image segments, and (b) statistic-based extraction of asymmetric segments.

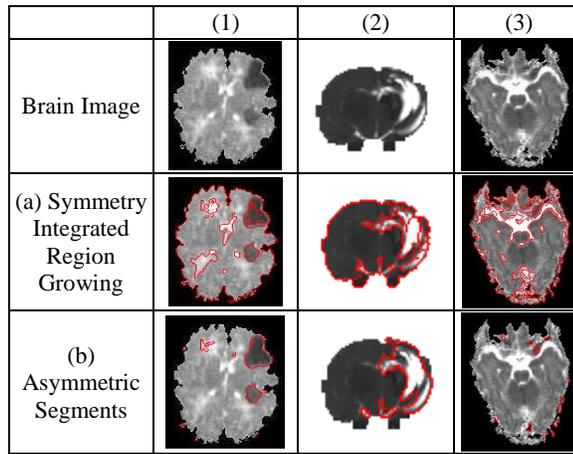


Fig 3.3. Sample results of 2D/3D segmentation, for injured brains of (1) human, (2) animal, (3) human. In (b), pixels within the red boundaries are asymmetric segments.

- **Symmetry-integrated Region Growing for Image Segmentation:** In this paper, a symmetry-integrated region growing segmentation [73] is used to segment brain into different tissues. Region growing aggregates neighboring pixels having similar characteristics to form uniform region segments iteratively. Let us denote $\delta(p_i, r_j)$ as the homogeneity aggregation criterion, where p_i is the pixel under consideration and r_j is the region to which p_i is to be grown into. The criterion for pixel p_i growing to region r_j is,

$$\delta(p_i, r_j) < \delta_a \quad (3.5)$$

where δ_a is a predefined aggregation criterion threshold (see Section 3.4.1 for setting the value of this threshold). After the pixel aggregation, neighboring regions with homogeneity criterion less than a predefined region merging criterion threshold δ_m , are merged into one region segment (see Section 3.4.1 for setting the value of this threshold). Traditional region growing for segmentation accepts image intensity as a homogeneity criterion for pixel aggregation,

$$\delta(p_i, r_j) = \delta_{gray}(p_i, r_j) = \|F_{gray}(p_i) - F_{gray}(r_j)\| \quad (3.6)$$

which is the Euclidean distance of gray scale features between pixel p_i and its neighboring region r_j . Since the brain is a highly symmetric structure, a symmetry constraint derived from the symmetry affinity of an image is integrated in the region growing process to assure that most symmetric parts are segmented appropriately. Our symmetry criterion is given by:

$$\delta_{symmetry}(p_i, r_j) = \frac{\frac{\pi}{2} + \arctan(\sqrt{(1+C_{p_i})(1+C_{r_j})})}{\pi} + \frac{1 + |\sqrt{C_{p_i}} - \sqrt{C_{r_j}}|}{2} \quad (3.7)$$

where C_{p_i} and C_{r_j} in equation (3.7) are symmetry affinity values of pixel p_i and neighboring region r_j (mean value), respectively, obtained by Equation (3.4). In equation (3.7), the first term controls the symmetry level, which means that if both patterns p_i and r_j indicate low symmetry affinities (highly symmetric), they are more likely to be aggregated by decreasing the constraint $\delta_{symmetry}(p_i, r_j)$. The second term

favors similar symmetry affinities. As a result, the symmetric pairs of tissues are segmented more properly guided by this criterion. In our work, this symmetry criterion is combined with gray scale intensity and texture to build an integrated criterion for region growing as given below,

$$\delta(p_i, r_j) = \delta_{\text{symmetry}}(p_i, r_j) \cdot [\delta_{\text{gray}}(p_i, r_j) + \delta_{\text{texture}}(p_i, r_j)] \quad (3.8)$$

where $\delta_{\text{symmetry}}(p_i, r_j)$ is given by Equation (3.7), and $\delta_{\text{gray}}(p_i, r_j)$ uses the intensity difference of pixel p_i and the mean value of its neighboring region r_j as the gray-level criterion (see Equation (3.6)). Since gray scale feature is sensitive to the noise and variations of intensity among MRI slices, a texture feature $\delta_{\text{texture}}(p_i, r_j)$ is also applied by using a set of Gabor filters. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination [88].

We call equation (3.8) as the *symmetry integrated image segmentation*. After the segmentation, regions with high symmetry level will be segmented more symmetrically. Example segmentation results are shown in Fig. 3.3(a). More segmentation results are given in Figs. 3.10 and 3.11.

This approach segments proper boundaries of all regions (symmetric or asymmetric). Note that Gupta et al. [14] used a different symmetry integration (edge-weight) approach to enhance the symmetry level in a graph-cut image segmentation. This symmetry integration has very limited segmentation improvements, compared to our method [73]. The segmented regions obtained in this section are further processed to extract the

asymmetric segments in the following step.

- **Statistics-based Extraction of Asymmetric Segments:** The region-growing algorithm with symmetry constraints for image segmentation separates different brain tissues by ensuring that naturally symmetric parts are segmented properly, i.e., symmetry is preserved. As the next step, the extraction of asymmetric segments, separates the segmented regions into symmetric and asymmetric segments. The purpose of asymmetric segment extraction is to cover all the ROIs while at the same time allow the number of other asymmetric regions belonging to the normal brain tissues to be as small as possible. The kurtosis and skewness measures of symmetry affinity matrix, corresponding to the regions, are used in the detection of asymmetric segments which may correspond to ROIs. For a region consisting of n pixels, its kurtosis and skewness are given by:

$$\text{Kurtosis: } g_4 = \frac{\mu^4}{\delta_2^2} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^2} \quad (3.9)$$

$$\text{Skewness: } g_3 = \frac{\mu^3}{\delta_2^{1.5}} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^{1.5}} \quad (3.10)$$

where x_i is a sample of n values (for a segmented region) in a distribution, and its value is equal to symmetry affinity value of pixel p_i . \bar{x} is the mean value of the samples in a distribution, and it is equal to the mean symmetry affinity value of a region.

Kurtosis is a measure of the level of "peakiness" of the probability distribution

function of a random variable. A larger kurtosis means that the distribution indicates a higher and narrower peak, as shown in Fig. 3.4(a). Kurtosis is used to detect an abnormality based on the fact that it measures the deviation of an abnormal distribution from the normal background [76]. We compute kurtosis of symmetry affinity values of the regions obtained using the symmetry-integrated region growing to detect asymmetric segments. This is based on the observation that the asymmetric segments in the symmetry affinity matrix yield candidate ROIs, where symmetry affinity values of pixels are very high (brighter pixels in Fig. 3.2(b)) and smoothly distributed. For each segment the kurtosis of its symmetry affinity distribution is computed using Eq. (3.9), resulting in a single kurtosis value for each region. Large kurtosis of a region means more deviation in its symmetry affinity distribution, which may lead to asymmetry.

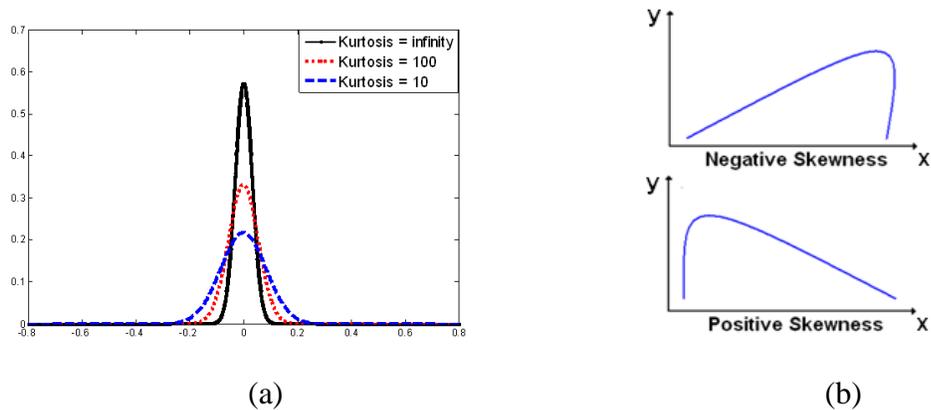


Fig 3.4. (a) Kurtosis: pdf for the Pearson distribution with kurtosis of infinity; 100; and 10; (b) Skewness: negative (left-tailed) and positive (right-tailed) distributions of skewness.

Skewness is another cue for the detection of asymmetric segments. The negative skewness of symmetry affinity value of a region means that its distribution is left-tailed to the mean value, and most of the values are far away than the mean value, as shown in the

top plot of Fig. 3.4(b). We consider the mean skewness value of every region as zero. In this situation, the more negative skewness means that the region's symmetry affinity distribution favors more asymmetry, and more of the symmetry affinity values of the region are larger than zero. The asymmetric segment extraction is summarized as follows:

(1) Discard symmetric regions whose mean symmetry affinity is quite low; note that a highly symmetric region will have a low symmetry affinity value.

(2) For each of the remaining regions, compute its kurtosis minus skewness combination $g = g_4 - g_3$ from Eq. (3.9) and (3.10), and build a region histogram for the combination g of all the candidate regions. For a segmented region in an image, if $(g_4 - g_3) > \delta_{ks}$, where δ_{ks} is a predefined threshold, then this region is extracted as an asymmetric segment.

Since a larger kurtosis g_4 and more a negative skewness g_3 both indicate a more asymmetric region, larger $(g_4 - g_3)$ of a region indicates that its pixels are distributed with higher symmetry affinity values. The value of the threshold δ_{ks} is set in Section 3.4.1. The extracted asymmetry segments are shown in Fig. 3.3 (b).

3.3.3. 3D Relaxation-based Segmentation of Symmetry Affinity Matrix

The segmentation of symmetry affinity matrix (obtained in section 3.3.1) is realized by extending a 2D relaxation method for image segmentation [74] to 3D. For this paper, the algorithm (see step 3 in Fig. 3.1) iteratively separates the histogram of symmetry affinity matrix into two groups of pixels, corresponding to symmetric (λ_1) and

asymmetric (λ_2) parts of a MR image. For an image composed of N pixels, which fall into two classes λ_1 and λ_2 , this algorithm is realized by a criterion function that maximizes consistency and reduces ambiguity of pixels in an image, as shown below,

$$C(prob_1, \dots, prob_N) = \sum_{i=1}^N prob_i \cdot q_i \quad (3.11)$$

where $prob_i$ is the probability that the i th pixel belongs to class λ_1 and λ_2 , and the compatibility of i th pixel q_i with its neighbors is given by its mean neighborhood probability. The consistency C in Eq. (3.11) is measured by the inner product of $prob_i$ and q_i , and the ambiguity is measured by entropy of $prob_i$. The original symmetry affinity of each pixel valued between 0 and 1 is assigned as pixel probability $prob_i$. q_i is obtained by the weighted sum of symmetry affinities of its 8-neighboring pixels from all slices at the same 2D location. This mean 3D neighborhood probability q_i is given by:

$$q_i(\lambda_k) = \frac{1}{8m} \sum_{j \in V_i} w_j \cdot prob_j(\lambda_k), \quad k = 1, 2 \quad (3.12)$$

where m is the number of slice in the MRI sequence, and $8m$ is the total number of pixels in the 3D neighborhood V_i of the i th pixel. $prob_j(\lambda_k)$ is equal to the symmetry affinity value of j th pixel in V_i , where λ_k represents k th class. λ_1 is the asymmetry class, and λ_2 is the symmetry class. These are the two classes in this paper. w_j is the weight of the j th pixel, where a lower weight is assigned to pixels that are farther in the

other MRI slices. The values of w_j satisfy the following two constraints: $\sum w_j = 1$ and $w_{j\pm 1} = 0.5w_j$. The first constraint ensures the normalization of the probability $prob_j$. The second constraint means that the weight of slice $j \pm 1$, which is one slice farther from the current slice j under consideration, is half the value of slice j . The following iterative process [74] separates the distribution of symmetry affinity histogram into symmetric and asymmetric classes (groups), by updating the symmetry affinity $prob_i$ of i th pixel:

$$prob_i^{(n+1)}(\lambda_k) = prob_i^{(n)}(\lambda_k) + \rho_i^{(n)}[q_i(\lambda_k) - 1], \quad k = 1, 2 \quad (3.13)$$

where ρ_i in iteration n is updated as:

$$\rho_i^{(n)} = \begin{cases} \alpha_1 \rho_{i_{\max}}^{(n)}, & \text{if } 2q_i(\lambda_1) - 1 > 0 \\ \alpha_2 \rho_{i_{\max}}^{(n)}, & \text{if } 2q_i(\lambda_1) - 1 < 0 \end{cases} \quad (3.14)$$

and

$$\rho_{i_{\max}}^{(n)} = \begin{cases} \frac{1 - prob_i^{(n)}(\lambda_1)}{2q_i(\lambda_1) - 1}, & \text{if } 2q_i(\lambda_1) - 1 > 0 \\ \frac{prob_i^{(n)}(\lambda_1)}{1 - 2q_i(\lambda_1)}, & \text{if } 2q_i(\lambda_1) - 1 < 0 \end{cases} \quad (3.15)$$

where α_1 and α_2 in Eq. (3.14) are the control parameters such that $\alpha_1 + \alpha_2 = 1$ ($\alpha_1 = \alpha_2 = 0.5$ in this paper for all datasets). The initial value $prob_i^{(0)}$ is equal to the symmetry affinity value of the pixel. Normally, two iterations are enough to group symmetric and asymmetric pixels in the histogram of a symmetry affinity matrix, where the mean value of the symmetry affinity matrix is used to partition its histogram. The

segmented symmetry affinity pixels are used to extract the asymmetric groups of pixels, which have higher mean symmetry affinity value than symmetric groups. The sample results are shown in Fig. 3.5.

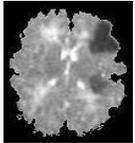
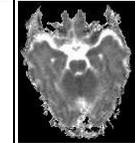
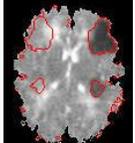
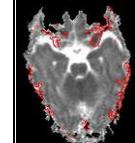
	(1)	(2)	(3)
Brain Image			
Asymmetric Groups			

Fig 3.5. Sample results for the extraction of asymmetric groups by 3D relaxation segmentation, for injured brains of (1) human, (2) animal, (3) human. Pixels within the red boundaries are asymmetric groups.

3.3.4. Asymmetric Region Extraction and ROI Detection

In this section, both asymmetric segments (obtained in Section 3.3.2) and asymmetric groups (obtained in Section 3.3.3) are used to extract asymmetric regions. Finally, the asymmetric regions are classified into the desired regions-of-interest (ROIs). This procedure (see step 4 in Fig. 3.1) consists of two major components: (a) computing the intersection of asymmetric segments and groups, and (b) unsupervised ROI detection.

- **Computing the Intersection of Asymmetric Segments and Groups:** The potential asymmetric regions are obtained by computing the intersection of the results from asymmetric segments (see Section 3.3.2) and asymmetric groups (see Section 3.3.3). The asymmetric segments which have over 50% (kept constant for all the results) overlap with asymmetric groups, are chosen as the final asymmetric regions. The overlap rate of 50% is computed by the pre-assumption that the opportunity of overlap and non-overlap

between the two results are equal. This overlap process is able to further eliminate any noisy non-ROI regions and to refine the final asymmetric regions, while at the same time preserving all ROIs. Sample results in Fig. 3.6(a) show that all the asymmetric ROIs are obtained, while the number of other normal asymmetric regions is minimized. Since symmetry is used as a continuous feature, some normal brain segments that are lightly asymmetric can be filtered out. Other normal brain parts that are lightly asymmetric, can be merged into a larger symmetric brain parts by the image segmentation.

- **Unsupervised ROIs Detection:** Asymmetric regions obtained above are the potential candidates for extracting the final ROIs. An unsupervised Expectation Maximization (EM) classifier with a Gaussian Mixture Model (GMM) [77] is used to classify candidate asymmetric regions (see Fig. 3.6(a)) into two classes: ROIs vs. non-ROIs. The final classification is performed by using a 2-D feature vector, composed of a potential region's mean gray scale intensity in all the slices and its mean 3D volume. Since the ROIs are compact in specific 3D positions, they provide correlated 2D positions in 2D slices. The 3D volume is obtained by binarization of the final asymmetric regions from all the slices (see Fig. 3.6(a)), where the pixels belonging to asymmetric regions are labeled as 1, and the pixel belonging to symmetric regions are marked as 0. The binary images of all the slices in an MRI sequence are added to obtain a 3D volume, as shown in Fig. 3.7, where brighter pixels indicate a larger volume value. It is clear that the MRI slices in the same sequence share the same volume. The value of volume indicates how frequently the asymmetric regions appear in slices at the same 2D position. In other words, if an ROI exists in 3D MRI, its volume added up from 2D slices in the same projected 2D location will have a high

volume. The mean value of 3D volume of each asymmetric region in Fig. 3.6(a) is computed and used as a region feature for classification, where the other feature is a region's mean gray scale value across all the slices. The classification by EM+GMM [87] using these two features is unsupervised and fully automated. Generally there are multiple asymmetric regions among all the slices in a sequence. The classified group of asymmetric regions with larger mean 3D volume is identified as the ROI class (see Fig. 3.6(b) as an example). Since the gray scale value will misclassify normal regions into ROIs (if they have similar gray scale values), the use of asymmetry as 3D volume will discard these normal asymmetric regions, and it is more effective compared to 2D asymmetry information used before. Meanwhile, the combination of 2D slices and 3D information enables the detection of the high asymmetric fields, where the ROIs are most likely to be located in the same 2D position over all the slices. The following two post-processing steps are performed to obtain the final ROIs: (1) Small regions with areas below a threshold (20 pixels in all the experiments) are removed as non ROIs; (2) Holes within ROIs are filled.

As an example, the final classified ROIs are shown in Fig. 3.6(b).

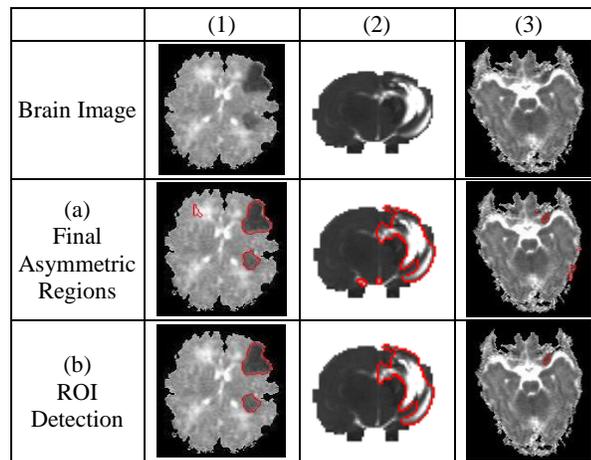


Fig 3.6. Results for (1) human injury, (2) animal injury, (3) human injury.

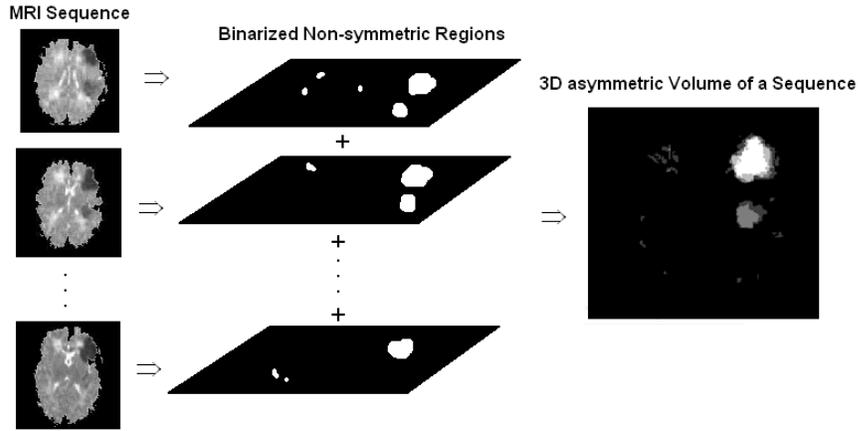


Fig 3.7. The generation of 3D asymmetry volume of a MRI sequence.

3.4. Experiments

In this section, we present experiments on three MRI datasets and perform both quantitative and qualitative analysis to demonstrate the effectiveness of the proposed symmetry-based ROI detection method.

3.4.1. Datasets and Parameters

- **Datasets:** The proposed method was tested on two brain MRI datasets from humans and animals provided by Loma Linda University Medical Center. The databases used are:

(a) *The MRI brain sequences of human with injury.* It is composed of two MRI sequences taken from two different patients with brain injuries as ROIs. The two patients in this dataset are labeled #A and #B, and the two sequences contain 10 and 9 MRI slices, respectively. Both sequences are MR images with ADC Map. The resolution is 192×128 for each image.

(b) *The MRI brain sequences of animal (rat) with injury.* This dataset is composed of 6 MRI sequences taken on 6 different dates (1 day pre, 1st day, 7th day, two weeks, three

weeks and four weeks), of the same rat brain with injury. The injury is injected manually. Each of the 6 sequences contains 16 MRI slices. A total of 96 MR images are used. The sequences taken on different dates clearly show the evolution of injury level with time. This paper provides the visual results of two sequences taken on the 1st day and after one week, labeled as #C and #D, respectively. The data are T-2 weighted MR images. The resolution is 128×192 for each image.

For both datasets described in (a) and (b), the medical experts from Loma Linda University Medical Center provided the ground-truth ROIs for evaluation purposes. Note that we do not work with T2 and ADC images directly, but with the images converted into TIFF image format. Thus, our algorithms do not make use of specific T2 and ADC values for ROI detection. The masks for the brain part of images (surrounded by the black background) are extracted by the hierarchical region splitting (HRS) method [86].

- **Parameters:** The parameter space for the method is composed of three thresholds:

(1) The pixel aggregation criterion threshold δ_a and the region merging criterion threshold δ_m for the region growing image segmentation (see Section 3.3.2). The value for pixel aggregation threshold δ_a is $2 \times 10^{-4} / std_image$ where std_image is the standard deviation of an image, making it a global threshold. The region merging threshold δ_m (see Section 3.3.2) is $1.2 / std_image$.

(2) The kurtosis/skewness threshold (δ_{ks} , see Section 3.3.2) for the extraction of asymmetric segments: $\delta_{ks} = (3.4 \times std_sym_affin) / (mean_sym_affin)$, where std_sym_affin and $mean_sym_affin$ are standard deviation and mean value of symmetry

affinity (see Equation (3.3)) of a MR image, respectively. The threshold is also determined by performing experiments and kept constant for all the datasets.

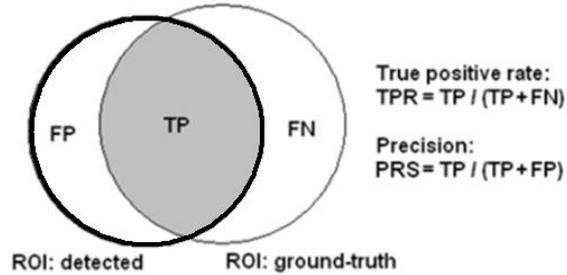


Fig 3.8. The illustration of true positive (TP), false positive (FP) and false negative (FN), and the associated true positive rate (TPR) and precision (PRS). The ROI detected by the approach, described in this paper, is shown in bold and the ROI given as the ground-truth is shown as non-bold.

3.4.2. Performance Metrics

The experimental results are evaluated using both qualitative and quantitative metrics. The qualitative metric includes visual ROI detection results and their comparison to the ground-truth ROI. The three quantitative measures used in the experiments are:

(1) The true positive rate (TPR) and (2) the precision (PRS). As indicated in Fig. 3.8, the TPR and PRS express the overlap between the ROI detected by the proposed method and the ground-truth ROI. Based on the ground-truth ROI, both TPR and PRS are true metrics when used to compare the performance among different methods.

(2) The receiver operating characteristic (ROC), which is the plot of true positive rate (TPR) versus the false positive rate (FPR) of the detected ROIs, with respect to the ground-truth ROIs. Each MRI sequence in the experimental results has a ROC plot. It is composed of several points, each of which is a TPR-FPR point of ROI of the volumetric results (overall TPR and FPR of ROIs in MRI sequence), with different parameter

settings. In this paper, the threshold δ_{ks} (see Section 3.3.2) for asymmetric segments extraction criterion $g = g_4 - g_3$ (from Equations (3.9) and (3.10)) is used. By tuning the threshold, different ROIs are obtained corresponding to various points on a ROC curve.

3.4.3. The ROI detection results

A challenging case appears when the ROIs do not exist in the MR slice (e.g., D2 in Fig. 3.11). In this case, no ROIs are detected by our approach, as expected, and the TPR and PRS are both 0 (see D2 in Table 3.8).

3.4.4. Discussion of the results

- **Qualitative Analysis:** Fig. 3.9 shows examples of the symmetry integrated segmentation and the final asymmetric regions and ROIs that are detected for the two datasets. More detailed example results by the proposed method for the two MR sequences are displayed in Figs. 3.10-3.11. As indicated from subplot (b) in Figs. 3.10-3.11, the image segmentation process is able to segment different brain tissues, especially for the ROIs, whose boundaries are outlined properly. As a result, a successfully segmented potential ROI will help to improve the later ROI detection performance. As suggested in Figs. 3.10-3.11, from subplot (c) to (d), most of the symmetric regions are separated from the final asymmetric regions. The final asymmetric brain parts (Figs. 3.10-3.11 (d)) are almost equal to the final ROIs (Figs. 3.10-3.11 (e)). There are some small regions and several asymmetric regions on the symmetric reflected side of a ROI, e.g., subplot (c) of MRI slices C5, C7, C11, D5 and D9 in Figs. 3.10 and 3.11, where the asymmetric regions include some non-ROI regions in the other

hemisphere of brain ROI. However, these asymmetric parts are eliminated by using the EM+GMM classifier (see comparison between asymmetric regions in (d) and the final ROI regions in (e) in Fig. 3.10-3.11), and it is evident that in most of the MRI slices, the asymmetric parts in subplot (d) are the same as the final ROI in subplot (e). This indicates the power of using symmetry in ROI detection.

The final ROIs in Fig. 3.10-3.11 are close to the ground-truth ROI, except in several cases where the boundaries of the ROI do not correspond to the ground-truth. This is seen in slice C5 in Fig. 3.10, where the ROI boundary is not as continuous as that of the ground-truth. Also for the slice C8 in Fig. 3.10, the detected ROI has a small hole within its region. Although their boundaries are slightly in error with respect to the ground-truth ROI, the error rate is still acceptable. Note that the low true positive and precision rates occur when the area of ROIs are too small, e. g., as in slices C2 and C3 shown in Table 3.7. In this case, the ROC plot uses slices from C4 to C12 as the source data.

- **Quantitative Analysis:** The quantitative results of all the datasets are shown in Tables 3.5-3.8. The proposed method achieved satisfactory true positive rate and precision in all the datasets. In Table 3.6, the overall true positive rate is far below the precision, due to the fact that in dataset #B, the false positive (ROI pixels detected by the proposed method that are non-overlapping with the ground-truth ROI pixels) is larger than the false negative rate (ground-truth ROI pixels that are not detected by the proposed method). This is seen in slice B7 in Fig. 3.11. On the other hand, for the Dataset #D, the true positive rate is larger than the precision, as shown in Table 3.8. Tables 3.5-3.8 show that most of slices have high precisions and true positive rates. The overall precision of

the sequence is between 76%-91%, and the overall true positive rate is between 82%-96%. Additionally, the ROC plots in Tables 3.5-3.8 show the performance. Table 3.9 shows the ROI detection results on the second datasets on all time points (including sequences #C and #D), indicating the evolution of the severity of injury at times. The proposed method reaches satisfactory detection accuracy at all the time points.

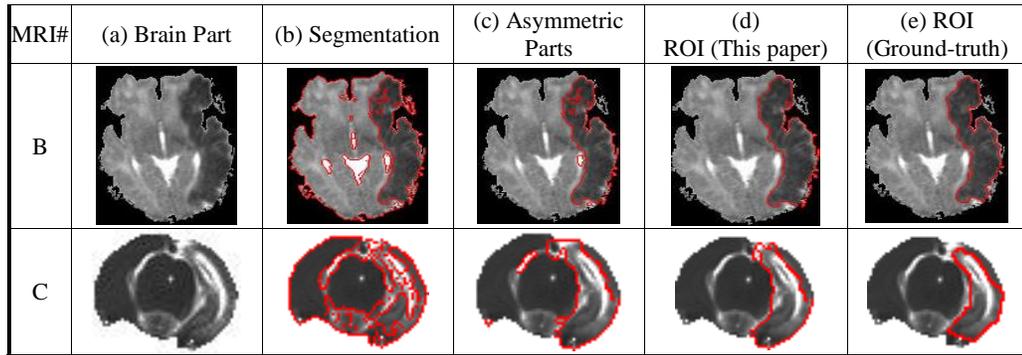


Fig. 3.9. Example results of the proposed method in different steps.

Table 3.5. Quantitative Analysis of MRI Dataset #A.

MRI#	Area of Brain	Area of ROI	% ROI	True Positive Rate	Precision
A1	4421	0	0	0	0
A2	5583	72	1.29%	80.56%	87.88%
A3	6707	190	2.83%	70.53%	92.41%
A4	7839	488	6.23%	98.16%	85.23%
A5	8667	700	8.08%	96.71%	94.29%
A6	9472	611	6.45%	91.65%	95.73%
A7	10166	546	5.37%	91.58%	92.25%
A8	10013	466	4.65%	90.34%	85.92%
A9	9089	341	3.75%	73.02%	94.68%
A10	8153	65	0.80%	98.46%	68.82%
Overall	80110	3845	4.34%	90.31%	90.70%

ROC Plot for Volumetric Analysis	

Table 3.6. Quantitative Analysis of MRI Dataset #B

MRI#	Area of Brain	Area of ROI	% ROI	True Positive Rate	Precision
B1	11957	53	0.44%	94.34%	67.57%
B2	10750	675	6.28%	94.96%	79.93%
B3	11258	2805	24.92%	88.84%	96.81%
B4	10952	3193	29.15%	87.63%	91.83%
B5	12057	2682	22.24%	84.11%	81.33%
B6	11638	2756	23.68%	85.85%	96.10%
B7	9767	2191	22.43%	76.22%	95.59%
B8	11402	2685	23.55%	77.99%	97.30%
B9	7832	201	2.57%	84.08%	55.05%
Overall	97613	17241	17.66%	84.31%	91.20%

ROC Plot for Volumetric Analysis

The ROC plot shows a curve that starts at a False Positive Rate of 0.08 and a True Positive Rate of 0.87. As the False Positive Rate increases to 0.48, the True Positive Rate increases to approximately 0.93. The y-axis is labeled 'True positive rate' and ranges from 0.85 to 0.93. The x-axis is labeled 'False positive rate' and ranges from 0.08 to 0.48.

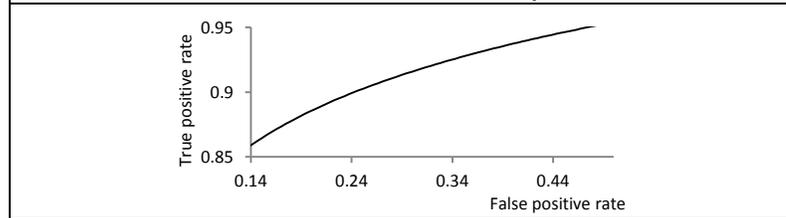
MRI #	(a) Brain Part	(b) Segmentation	(c) Asymmetric Regions	(d) ROI (This paper)	(e) ROI (Ground-truth)
C5					
C6					
C7					
C8					
C9					
C10					

Fig 3.10. ROI of MRI dataset #C: a patient on the first day.

Table 3.7. Quantitative Analysis of MRI Dataset #C

MRI#	Area of Brain	Area of ROI	% ROI	True Positive Rate	Precision
C1	823	12	1.46%	0	60.00%
C2	831	17	2.05%	11.76%	7.41%
C3	938	78	8.32%	6.41%	12.82%
C4	1159	54	4.66%	77.78%	67.74%
C5	1495	339	22.68%	85.25%	82.10%
C6	1735	434	25.01%	84.33%	83.18%
C7	1684	509	30.23%	92.73%	91.65%
C8	1848	611	33.06%	83.47%	94.44%
C9	1766	566	32.05%	93.11%	89.78%
C10	1828	393	21.50%	81.68%	62.45%
C11	1380	518	37.54%	84.75%	89.96%
C12	243	32	13.17%	62.50%	62.50%
C13	824	160	19.42%	64.38%	81.10%
C14	690	122	17.68%	40.16%	71.01%
C15	411	0	0	0	0
C16	243	0	0	0	0
Overall	17898	3845	21.48%	82.11%	82.43%

ROC Plot for Volumetric Analysis



MRI#	(a) Brain Part	(b) Segmentation	(c) Asymmetric Regions	(d) ROI (This paper)	(e) ROI (Ground-truth)
D2					
D5					
D6					
D7					
D8					

Fig 3.11. ROI of MRI dataset #D: a patient on the 7th day.

Table 3.8. Quantitative Analysis of MRI Dataset #D.

MRI#	Area of Brain	Area of ROI	% ROI	True Positive Rate	Precision
D1	881	0	0	0	0
D2	993	0	0	0	0
D3	952	33	3.47%	30.30%	83.33%
D4	1053	75	7.12%	33.33%	96.15%
D5	1515	240	15.84%	82.08%	86.03%
D6	1582	397	25.09%	97.23%	92.79%
D7	1838	509	27.69%	91.94%	86.35%
D8	1629	279	17.13%	93.19%	80.25%
D9	1819	420	23.09%	87.14%	88.83%
D10	1666	428	25.69%	96.50%	87.69%
D11	1416	238	16.81%	93.28%	73.03%
D12	1152	91	7.90%	100%	48.15%
D13	833	0	0	0	0
D14	459	6	1.31%	0	0
D15	0	0	0	0	0
D16	0	0	0	0	0
Overall	17788	2716	15.27%	89.76%	83.35%

ROC Plot for Volumetric Analysis

The ROC plot displays the True Positive Rate (Y-axis, ranging from 0.6 to 0.9) against the False Positive Rate (X-axis, ranging from 0.09 to 0.49). The curve shows a strong positive correlation, indicating high diagnostic accuracy. Key data points from the plot include (0.09, 0.65) and (0.49, 0.88).

Table 3.9. Quantitative Analysis of Animal Sequences for all the Time Points.

Time Points	Area of Brain	Area of ROI	% ROI	True Positive Rate	Precision
1 day pre	19997	5409	27.05%	80.58%	84.01%
1 day	17898	3845	21.48%	82.11%	82.43%
7 day	17788	2716	15.27%	89.76%	83.35%
2 weeks	20658	3035	14.69%	90.05%	88.90%
3 weeks	19928	3656	18.35%	88.01%	93.93%
4 weeks	20434	4300	21.04%	85.37%	97.54%

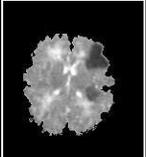
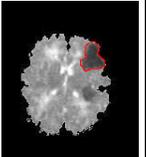
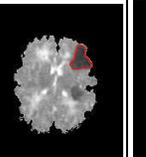
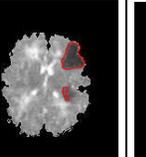
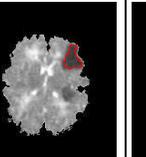
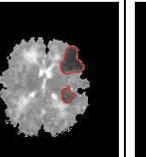
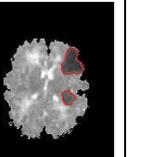
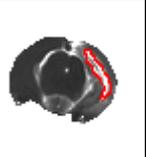
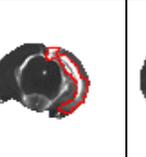
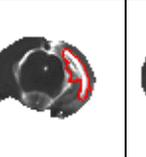
	Original Image	ROI - Hierarchical [85]	ROI - Watershed [80]	ROI - KNN [81]	ROI - Region Growing [82]	ROI – this paper	ROI – ground-truth
A5							
C7							

Fig 3.12. Comparison of ROI detection results.

- **Effect of the blurred boundary:** Slices in almost all datasets have the problems of very blurred ROI boundaries. Due to the use of gradient vector flow in image segmentation, many of the ROI detection results successfully detected, e. g., the slices A5, B2, C6-C9 and D5 shown in Figs. 3.10-3.11.

- **Effect of the global parameters:** Note that all the MR slices are run under global parameters (see Section 3.4.1). The global parameter has the advantages of high generality, low data dependent and it is automatic, but it may suffers from low detection accuracy, like slices C10, C12, C14, D3 and D4 as shown in Tables 3.7 and 3.8, or like slices of A3 and A9 in Table 3.5 where the true positive rate is much lower than the precision. However, most of the MR slices reach high accuracies under global parameters. On the other hand, the number of slices with low accuracies is small enough so that they do not largely affect the accuracy of the entire MR sequence, and most of them belong to the slices at the beginning or the end of the sequences, in which both the brain and injury areas are very small.

- **Effect of variation of gray-scale distribution within ROI:** In some of the slices, the gray-scale distribution within ROI is not unique, which makes the ROI detection more challenging. In our method, the use of gradient vector flow and the symmetry cues (besides the gray-scale feature), copes with the multiple distributions effectively. Refer to MR slices of C7-C9 and D9-D10 in Figs. 3.10 and 3.11 as examples.

3.4.5. Comparison with other Methods

The proposed method is compared with four other ROI detection methods [86, 81-83] on the same datasets, introduced as follows,

- **Comparison with the hierarchical region splitting (HRS) method [86]:** It is an automated ROI detection method which segments the MRI image hierarchically. The hierarchy was implemented using the Otsu's algorithm [78, 80]. Table 3.9 and 3.10 show that the proposed method outperforms the method of [86] by a large amount due to the following two reasons. *Firstly*, the global segmentation by Otsu's algorithm is not that effective since it uses the gray level histogram only, and no additional features like symmetry or texture are considered. ROIs are always segmented into two clusters in this situation. *Secondly*, after each segmentation iteration, small regions are removed, and many small ROI regions are also mistakenly eliminated without using symmetry and even gray scale to discriminate them.

- **Comparison with the Watershed segmentation method [81]:** The limitation of this ROI detection method comes from the over-segmentation and unsmoothed region boundary produced by watershed segmentation. The final ROI detection generates large number of false positive errors. The user needs to select the ROI manually from the

segmented regions, that is not automatic.

- **Comparison with the K-nearest neighbor (KNN) method [82]:** It uses the K-Nearest Neighbor (KNN) (K=9) classification and probabilistic segmentation to detect ROIs. Although a robust KNN model can be trained by prior tissue knowledge, this method is not automated and it is time consuming, and its performance is lower than our method.

- **Comparison with the region growing method [83]:** It outlines ROIs using region-growing segmentation, which is the same segmentation method that has been used in the proposed approach, but without the incorporation of symmetry. Its performance is lower than the proposed method due to the fact that it does not use enough region features to segment ROIs. Additionally, the user needs to know the location of ROIs in advance and assign a seed within the ROI for region-growing. This makes the method extremely user-dependent and not automatic.

- **Discussion of the comparison:** The quantitative results in Table 3.10 shows the advantages of using the proposed method, which outperforms all the other methods in terms of the true positive rate and the precision. The compared methods are also summarized and analyzed in Table 3.2.

Fig. 3.12 shows example ROI detection results by different detection methods. It is clear that the results of the proposed method are closest to the ground-truth ROI boundaries. Additionally, Table 3.11 performs the comparison on the second datasets on all the time points. Note that although other methods accept prior knowledge for ROI detection, their accuracies are not as high as the proposed method. The reason lies on the limitation of prior knowledge, which cannot cover all the distinct properties of ROI

regions, and it is also highly data dependent. Fig. 3.13 shows the comparison of %ROI (ratio of detected ROI area with the entire brain) by the different methods. The ratio is measured by the number of pixels. It is clear that the proposed method is closest to the ground-truth %ROI, in all time points. On the other side, the other methods have much higher variations from the ground-truth %ROI.

Table 3.10. Comparison With Other Methods

True positive rate				
Dataset#	#A	#B	#C	#D
This paper	90.31%	84.31%	82.11%	84.31%
Hierarchical [86]	72.85%	81.38%	61.00%	58.53%
Watershed [81]	82.38%	77.54%	68.45%	63.47%
KNN [82]	76.70%	81.27%	73.59%	79.01%
Region Growing [28]	87.20%	81.36%	78.47%	80.12%
Precision				
Dataset#	#A	#B	#C	#D
This paper	90.70%	91.20%	82.43%	91.20%
Hierarchical [86]	83.54%	88.76%	57.30%	55.97%
Watershed [81]	76.32%	71.65%	43.40%	63.88%
KNN [82]	77.24%	78.03%	75.11%	80.35%
Region Growing [28]	83.60%	76.19%	78.81%	84.10%

Table 3.11. Comparison of Animal Sequences for all the Time Points.

		1 day pre	1 day	7 days	2 weeks	3 weeks	4 weeks
True Positive Rate	This Paper	80.58%	82.11%	84.31%	90.05%	88.01%	86.37%
	Hierarchical [86]	70.62%	68.17%	82.90%	69.34%	84.40%	55.56%
	Watershed [81]	66.72%	68.45%	63.47%	74.08%	72.35%	77.64%
	KNN [82]	74.18%	73.59%	79.01%	82.11%	69.07%	81.30%
	Region Growing [28]	80.22%	78.47%	80.12%	87.96%	81.55%	80.20%
Precision	This Paper	84.01%	82.43%	91.20%	88.90%	93.93%	97.54%
	Hierarchical [86]	64.00%	79.17%	88.64%	81.90%	83.14%	73.06%
	Watershed [81]	58.90%	43.40%	63.88%	84.17%	88.34%	86.14%
	KNN [82]	77.32%	75.11%	80.35%	81.77%	65.45%	72.90%
	Region Growing [28]	81.60%	78.81%	84.10%	74.79%	63.00%	71.08%

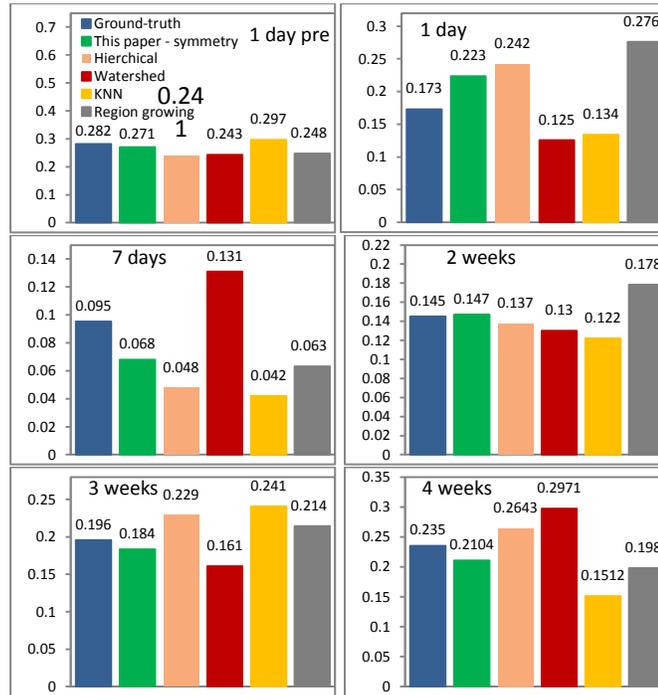


Fig 3.13. Comparison of ROI detection results: %ROI in volumetric measures.

3.5. Conclusions

This paper provides an *automated* symmetry-based ROI detection method for brain MR images. A symmetry-integrated image segmentation is used to ensure that the symmetry property of tissues is preserved in the segmentation results. Kurtosis and skewness are used on a symmetry affinity matrix to extract potential asymmetric segments. Asymmetry grouping using a 3D Relaxation algorithm is combined with kurtosis and skewness results to further refine the asymmetric regions. It allows the effective usage of 3D information contained in 2D slices of MRI sequences. Brain ROIs are finally extracted from asymmetric regions using an unsupervised classifier based on the Gaussian mixture model. Additionally, 3D asymmetry volume from 2D slices is proposed as a new feature for classification. Both qualitative and quantitative results on

three MRI datasets show that the computed ROIs closely approximate the ground-truth. By numerical comparison, the proposed method outperforms other existing ROI detection approaches [81-83, 86].

Chapter 4

Image Retrieval with Feature Selection and Relevance Feedback

4.1. Introduction

Content-based image retrieval (CBIR) [95] has been a significant topic of research in the last decade. In the CBIR context, an image is represented by a set of low-level visual features, which are generally not effective and efficient in representing the image contents, and they also have no direct correlation with high-level semantic information.

The gap between high-level information and low-level features is the fundamental difficulty that hinders the improvement of the image retrieval accuracy. Recently, a variety of solutions have been suggested that aim to bridge this semantic gap. Two of the most commonly used methods are online feature selection and user relevance feedback.

The feature selection [96] basically narrows the semantic gap by selecting the feature subset that best represents the query and discards redundant features. Image retrieval uses the selected feature subset to search the database such that the retrieved images are closer (resemble) to a given query.

The relevance feedback [97] narrows the semantic gap by making use of user provided judgments which are the labels (relevant or non-relevant) on the retrieved images for a query. The retrieval performance improves as the user provides more and more feedback information to the CBIR system. Query vector modification (QVM) [98] and feature relevance learning [99] are the two widely used methods to integrate user feedback information into the CBIR system.

Currently, the feature selection and relevance feedback are rarely used together to further narrow the semantic gap. The work in [102] applies feature selection as a form of feature weighting into the query vector modification (QVM) method for relevance feedback. However, it ignores the important classification or mutual information evaluation for feature selection. As a result, the work in [102] does not fully capture the key characteristics required for feature selection. In this paper, a measure of inconsistency from relevance feedback is integrated into feature selection, and combined with the Bayesian classifier to improve CBIR performance.

The feature selection procedure is composed of two steps: searching the combination of feature subsets within a feature space using specified search strategy, and evaluating the performance of the selected subset by a criterion. Existing evaluation criteria are classification performance, mutual information and entropy. In this paper, a new term called the measure of inconsistency from relevance feedback, is combined with the Bayesian classifier to build the overall criterion for feature selection. The combined criterion is able to select the optimal feature subset which leads to improve the image retrieval accuracy and better satisfies the user semantic requirements.

This paper makes the following contributions:

- (1) A new term called the measure of inconsistency, from relevance feedback, is combined into feature selection as a new criterion to further improve the image retrieval.
- (2) The semantic gap is further narrowed by combining the online feature selection and the user relevance feedback.

The outline of the chapter is as follows. Section 4.2 describes the technical approach of the new CBIR system in detail. Section 4.3 provides experimental results and analysis. Finally, conclusions are given in Section 4.4.

4.2. Technical Approach

The proposed CBIR system that integrates both online feature selection and the user relevance feedback is shown in Fig. 4.1. For a given query, the original features (color, texture and shape) are extracted from the query image, and the K-nearest neighbor (K-NN) algorithm with Euclidean metric searches the image database, and retrieves N top ranked images having features most closed to the query. The session with this query

is terminated when the user is satisfied with the retrievals, otherwise, the user provides relevance feedback by labeling the retrievals as relevant (positive feedback) and non-relevant (negative feedback). A measure of inconsistency is computed based on the user feedback and it is given as the input to the feature selection to select the feature subsets which will guide the K-NN search to obtain higher retrieval accuracy in the next CBIR iteration.

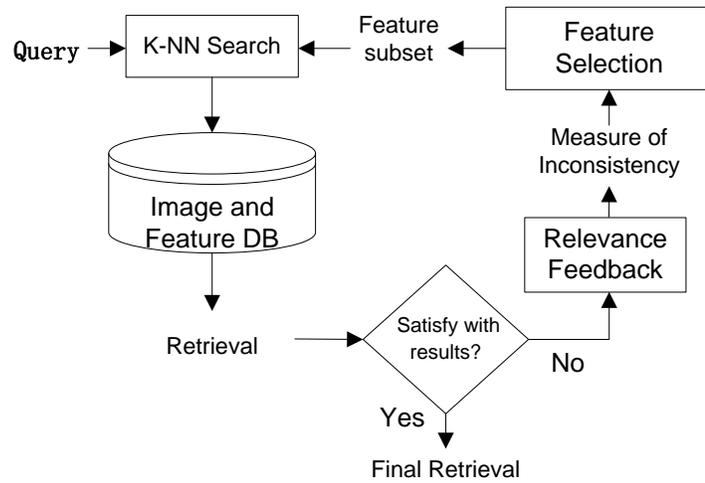


Fig. 4.1. The overall CBIR system diagram.

4.2.1. Measure of Inconsistency from Relevance Feedback

For each CBIR iteration, let $\mathcal{X} = \{x_1, \dots, x_N\}$ denotes the N retrieved images. The property $f(x_i)$ of the retrieved image x_i is expressed by its visual feature vector and its relevance feedback label: $f(x_i) = \{f_1(x_i), \dots, f_M(x_i), l_i\}$, where $i \in \{1, \dots, N\}$ denotes the i th retrieved image, and M is the dimension of feature, either be full dimension or dimension of the selected subset. The $l_i \in \{0, 1\}$ represents the feedback label of the retrieved image i , it can be either 1 for positive feedback or 0 for negative feedback. The

retrieved images χ are grouped into two clusters, as relevant (positive) and non-relevant (negative), according to user's feedback labels. The mean feature value is then computed for each of the two clusters, as given below.

$$\text{Mean positive feature vector: } \mathbf{Mf}_P = \frac{1}{N_P} \sum_{\substack{k=1 \\ (l_k=1)}}^{N_P} \{f_1(x_k), \dots, f_M(x_k)\} \quad (4.1)$$

$$\text{Mean negative feature vector: } \mathbf{Mf}_N = \frac{1}{N_N} \sum_{\substack{k=1 \\ (l_k=0)}}^{N_N} \{f_1(x_k), \dots, f_M(x_k)\} \quad (4.2)$$

where N_P and N_N is the number of positive and negative feedback images, respectively, satisfying $N_P + N_N = N$. The measure of inconsistency θ_{RF} is computed by the two mean vectors as shown below,

$$\theta_{RF} = \arccos\left(\frac{\mathbf{Mf}_P \bullet \mathbf{Mf}_N}{\sqrt{\|\mathbf{Mf}_P\|_2^2 \cdot \|\mathbf{Mf}_N\|_2^2}}\right) / 2\pi \quad (4.3)$$

which is the angle between the mean positive and negative feature vectors, and it is normalized into $\{0,1\}$. $\mathbf{Mf}_P \bullet \mathbf{Mf}_N$ is the dot product of the mean relevance and non-relevance vectors, and $\|\cdot\|_2^2$ is 2nd order norm operator. The larger the measure of inconsistency the better it is since we need the two mean positive and negative feature vectors to be as separated as possible. The inconsistency measure is further used as an evaluation criterion to guide the feature selection.

4.2.2. Feature Selection Combined with User Feedback

The feature selection block in Fig 4.1 starts with the original image features and

outputs the optimal feature subset. The realization of the feature selection block is indicated in Fig 4.2. The block (1) in Fig 4.2 refers to the feature space search strategy, namely the sequential forward selection [100], in which (a) features are sequentially selected from original features to build the feature subset, and (b) the feature dimension that is selected to the subset is uniquely decided by the feature performance evaluation criterion. The evaluation criterion, called the wrapper evaluation, is the most important element in feature selection system. In this paper, the new evaluation criterion is the combination of Bayesian classifier and the measure of inconsistency. The feature dimension having the highest classification results and measure of inconsistency, is selected and added into the current subset to build the new subset [101]. The wrapper evaluation criterion in this paper is shown below,

$$C_{wrapper} = \alpha \cdot C_{Bayesian} + \beta \cdot \theta_{RF} \quad (4.4)$$

where $C_{Bayesian}$ is the classification result of the Bayesian classifier, and θ_{RF} is measure of inconsistency introduced in Section 4.2.1. The weights are set to $\alpha = \beta = 0.5$. Traditionally $C_{wrapper} = C_{Bayesian}$. The equation (4.4) is the improvement of the tradition wrapper evaluation criterion by integrating the user feedback evaluation θ_{RF} . As a result, the best feature subset will be selected by criterion in equation (4.4), and the selected subset will have both highest classification and feedback inconsistency. Since user feedback is integrated into feature selection, user will provide a higher percentage of positive feedback in the next feedback iteration, based on images that are retrieved using the best feature subset selected by equation (4.4). Measure of inconsistency of a

candidate feature subset, with dimension M , is computed by equation (4.1)-(4.3). It is worth to note that the positive and negative feedback images are accumulated from all iterations to compute the measure of inconsistency.

The Bayesian classifier is extensively used in the wrapper evaluation criterion. The classifier estimates the label of an image by processing its feature vector, using the maximum a posteriori (MAP) probabilistic approach. From the comparison of the Bayesian classification results with the actual image class labels, an estimate of the correct classification rate (CCR) [101] is obtained as the feature subset evaluation $C_{Bayesian}$. After the search of block (1) in Fig 4.2, all feature dimensions are ranked according to the results of the evaluation criterion of equation (4.4). The feature ranking as well as the related performance evaluation are put into block (2) in Fig 4.2 to select the subset with the highest evaluation, as the final selected subset.

As shown in Fig 4.1, the feature selection provides more effective feature subset, which is input into the K-NN search for the next retrieval iteration. With the improved feature subset selected by the measure of inconsistency, the K-NN search ranks the images in a database that better represents the user feedback with higher accuracy.

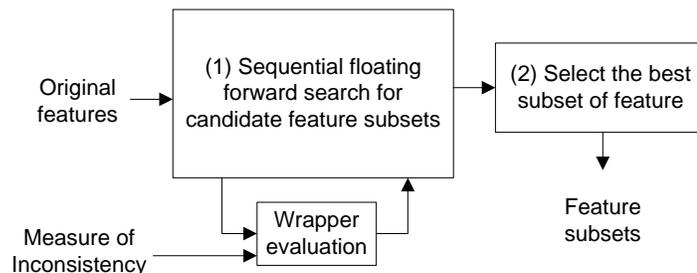


Fig 4.2. The feature selection diagram with user feedback.

4.3. Experiments

4.3.1. Datasets

In experiments, we run the CBIR system on two image databases, the first of which is the butterfly image database (<http://janzen.sas.upenn.edu/>) containing 29 highly similar classes with a total of 7600 images. The example images are shown in Fig 4.3(a). The second database has 210 natural images collected from Google Images, with 5 classes related to semantic concepts as snowy mountains, trees, falls, bridges and sand beaches that are quite similar, and example images are shown in Fig 4.3(b). As in Fig 4.3, the two databases are labeled DB#1 and DB#2, respectively.

We use features covering wide range of image properties. Totally 27 feature dimensions are extracted from the entire image, composed of following 4 sets of feature properties: (1) mean and standard deviation of the RGB components of color space, totally 6 dimensions. (2) The HSV components of color space, with the same distribution as RGB, so it also has 6 dimensions. (3) 8-dimension texture feature derived from the mean and standard deviation of the filtered image by Gabor filters at 4 orientations in steps of 45 degrees. (4) The 7-dimension shape feature derived from first 7 central geometric moments of the image. Totally 27 dimensions of features are extracted to build the image and feature database as shown in Fig 4.1. And they are also used as the original feature sets for feature subset selection. In the experiments, totally 20 images are retrieved at each iteration.

4.3.2. CBIR Combined with Feature Selection and Relevance Feedback

Fig 4.3 (a) and (b) provides example images. The two image databases indicate large overlapped properties which hinder the retrieval accuracy. For instance, in Fig 4.3(a), the images in class #13 and #25 are visually similar. In Fig 4.3(b), objects of ‘snowy mountain’ and ‘falls’ share the similar dominant properties. Both class #9 in Fig 4.3(a) and ‘bridge’ in Fig 4.3(b) show intra-class variations. The two databases are challenging cases that have overlapped properties among different classes, as well as high variations within class. By using our method, feature selection will search feature subset that best discriminate among classes and discard overlapped feature properties.

- ***Comparison of feedback precisions:*** The feedback precision is defined as the percentage of positive feedback in each feedback iteration. In Fig 4.3(c) and (d), the proposed method is compared to the query vector modification (QVM) scheme, in terms of feedback precision. QVM [98, 102] is one of the most widely used relevance feedback techniques. It modifies the query feature so that it will move closer to the relevant feature points and move away from non-relevant points. The final precision at each feedback iteration is computed by averaging precisions from different query sessions. Our method outperforms QVM in every feedback iteration as in Fig 4.3(c) and (d). Precisions after the 7th iteration are stabilized (almost unchanged).

- ***Comparison of retrieval precisions:*** The retrieval precision is defined as the percentage of the *true* retrieval in final retrieval results. For systems including the relevance feedback, this precision is the final (optimal) retrieval precision after the last feedback iteration. In Table 4.1, the proposed method is compared to three different

methods in terms of the retrieval precision. In order to ensure the comparability, all methods use K-NN search for retrieval and use the same feature data introduced in section 4.3.1. Additionally, for all the methods, the same queries are tested with the same number of repetitions, and the average precision is computed. *RF_ONLY* method (QVM) in Table 4.1 is the same method used in Fig 4.3(c) and (d).

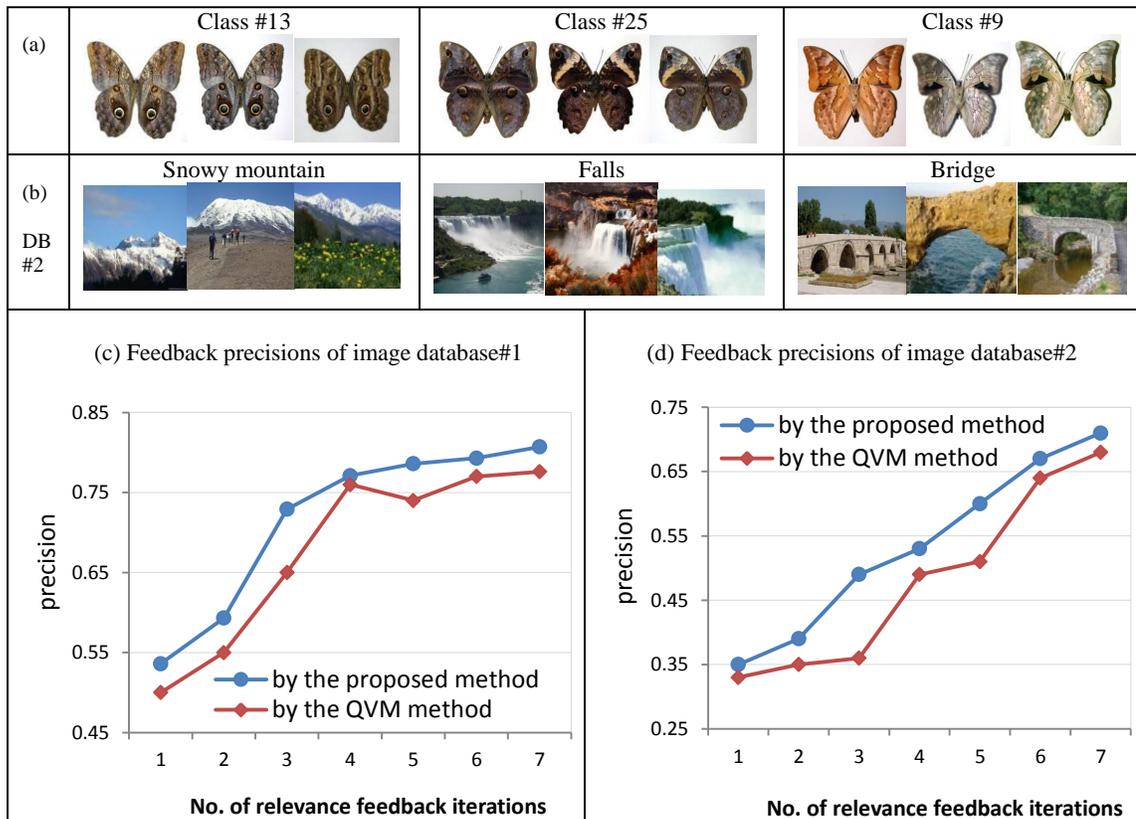


Fig 4.3. Example images in databases, and performance comparison of the proposed method.

In Table 4.1, the *RF_ONLY* method applies QVM at each feedback iteration, but uses the original features without feature selection. The *FS_ONLY* method uses the feature selection [101], which is the same selection scheme as used in our system, to select the best feature subset and input this subset into K-NN search for the retrieval, but no relevance feedback is performed. The *RT_ONLY* method only runs the K-NN search

for the retrieval by original feature data, without feature selection and relevance feedback. It is obvious that our method reaches higher precision, than those using only one of the relevance feedback and feature selection schemes.

Table 4.1. Comparison of retrieval precisions.

	DB#1	DB#2	Feature Selection	Relevance Feedback
<i>RF_FS:</i> This paper	80.7%	71.2%	YES	YES
<i>RF_ONLY:</i> QVM [8]	78.6%	69.0%	×	YES
<i>FS_ONLY:</i> [7]	76.1%	68.3%	YES	×
<i>RT_ONLY:</i> traditional retrieval	74.9%	66.5%	×	×

4.4. Conclusions

In this paper, we presented a new approach that combines relevance feedback and feature selection to improve the performance of a CBIR system. The approach uses a new criterion called the measure of inconsistency to guide the feature selection, in order to improve the image retrieval by integrating the user relevance feedback information. We performed experiments on different sizes of image databases to indicate the benefits of the proposed method. We showed the improvements in both feedback and retrieval precisions over the other current methods.

Chapter 5

Local Kernel Integrated Feature Selection for Image Classification

5.1. Introduction

Image classification [103-105] is commonly used for many practical applications such as object recognition [103], medical imaging [104] and image retrieval [105]. In image classification, an image is represented by a set of low-level global visual features, e.g. color, texture and shape, and the discriminative ability among the classes largely depends on the image features used. Two challenging problems commonly exist for image classification approaches.

Problem (1): There exists a significant redundancy among a large number of low-level features which prohibits the achievement of ideal classification.

Problem (2): The global features are inadequate to represent the important and discriminative local object information present in an image.

The objective of feature selection is to address the *problem (1)* by identifying the feature subset that is most predictive, and it is critical to minimize the feature redundancy and the classification error. Feature selection has been explored in many image processing, computer vision and pattern recognition tasks [106-109]. A survey of state-of-the-art feature selection methods is provided in [110].

The essence of *problem (2)* in image classification is usually characterized by the loss of important local image contents like regions and objects. Most objects of interest in an image cannot be effectively presented by using only the global image feature from the entire image, since they are either a small part of many objects in an image, or they are overlapped or distorted. Most of the current work extracts global features from the entire image. The popular solutions of *problem (2)* are to make use of the local image contents by partitioning an image into regions, and classifying these image regions instead of classifying an entire image [111], or computing the region-based image distance [112-113]. The local features help to improve the classification by means of highlighting important local objects (regions) and eliminating the effect of noisy background or unimportant regions.

In this paper, the local feature selection is realized by a novel approach, that integrates a local kernel into the Bayesian classifier, to guide the feature selection. The

proposed method contains the advantages of both feature selection and local features, and at the same time achieves improvement in classification accuracy. Furthermore, a multiple kernel learning is used to select an optimal classifier for the image classification.

The outline of this chapter is as follows. Section 5.2 gives an overview of related work and provides the contributions of this chapter. Section 5.3 describes the technical approach in detail. Section 5.4 provides experimental results and discussions. Finally, conclusions are given in Section 5.5.

5.2. Related Work and Contributions

5.2.1. Related Work

The aim of feature selection is to solve the *problem (1)* by decreasing the information redundancy in the raw feature data and improving the feature discrimination capability. The feature selection is usually described as a search problem in the feature space as follows [110]:

(1) *Feature space search*, requires a general strategy with which the feature space is explored. It is in relation to the portion of feature space that is explored with respect to their total size of the feature space. It also includes the mechanism by which possible feature subsets are proposed. For a feature space of dimension n , there are a total

$\sum_{m=1}^{n-1} n!/m!$ of combinations of feature subsets, with different dimensions and contents. An

efficient feature space search aims to search for optimal feature subsets in a minimized search space. The most commonly used search strategies are namely the exhaustive search, the sequential forward search (SFS) and the sequential forward floating search

algorithm (SFFS) [101]. Among the above, the SFFS search strategy, which is used in this paper, has been verified to provide the selected subset closer to the global optimal, with a much smaller search space.

(2) *Evaluation measure* is a function by which the candidate searched feature subsets are evaluated, allowing to compare different candidates, guiding the search process and selecting the optimal feature subset. Current evaluation functions accepted are classifiers, e.g. the Bayesian classifier [101] and SVM [108], which select an optimal feature subset which has the highest classification accuracy based on the dataset used. Other criterion functions include mutual information [107], etc. Since the objective of the proposed method is the classification of image, this paper uses the classifiers for evaluation function, within which the Bayesian classifier is chosen, for the reason of its requirement of small amount of training data and the independence assumption among the classes.

The other approach for improving classification is the use of local features, instead of global features from the entire image. The local feature methods are commonly divided into two categories: point-based features, of which the most frequently used is the SIFT operator [30], and the region-based features [111-113], which are normally extracted from the segmented regions. The point-based features can be made scale and rotation invariant, and they are helpful in object matching and tracking, and the region-based features, which are used in this paper, are more suitable for image classification since they contain color, shape, texture and other properties of objects. Within the region-based approaches, the work of [111] basically classify segmented regions and for each image it builds a region histogram, based on which the images in a

database are categorized. However, more effective applications of local feature are focused on region-based image distance metric [112-113], within which the earth mover distance (EMD) [112] and the integrated region matching (IRM) [113] are commonly used. The overall similarity distance between images incorporates properties of all the regions in images by a region matching scheme.

The combination of feature selection and local feature information, called the local feature selection, has not been thoroughly investigated. Current work [116-118] are listed in Table 5.1. In this paper, local feature selection is realized by a novel way of integration, in which a local kernel is integrated into the Bayesian classifier.

5.2.2. Contributions

The paper makes the following contributions:

- 1) It provides a new way to combine the feature selection and local feature information, by integrating local kernel into the distance metric of a Bayesian classifier, to guide the local feature selection. Thus, both *problem (1)* and *(2)* (see in Section 5.1) are solved.
- 2) A multiple kernel learning scheme is used to select the optimal inducer from multiple Bayesian classifiers.
- 3) Detailed experiments with three segmentation and three feature selection approaches show the improvement of image classification achieved by the proposed approach.

5.3. Technical Approach

In the system diagram of our approach as shown in Fig. 5.1, the region features are extracted from the segmented regions, and they are further used to compute the region-based image similarity distance. This distance is used as a local kernel to be integrated

into the Bayesian classifier to guide the feature selection and image classification. A local kernel learning is used to select the optimal classifier. The system outputs the optimal image classification result, generated by its selected feature subset. The proposed local kernel metric obtains higher classification accuracy compared to other methods.

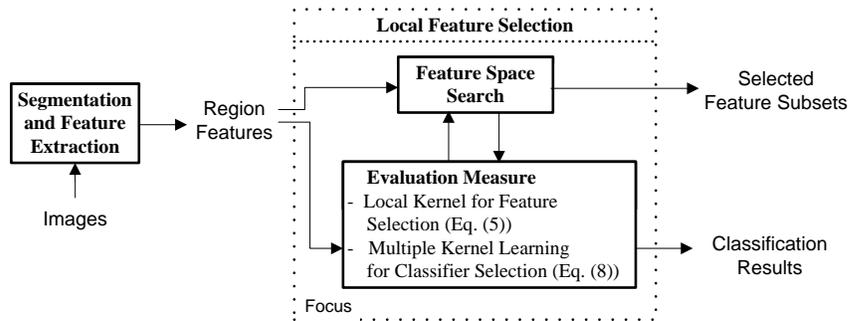


Fig 5.1. The overall system diagram for this paper.

Table 5.1. Summary of the related local feature selection methods.

Papers	Summary	Pros	Cons
Mohamad et al. [116]	Feature selection to detect local face regions in images.	Robust to face distortions and background regions.	Only facial features are used (low generality); gray scale images only; cannot detect the face boundary (only a bounding box).
Sun et al. [117]	Feature selection using the combined feature axis for region-based retrieval.	More relevant retrievals are achieved using feature selection based on local image regions.	Only consider relevant and irrelevant images, does not obtain multiple classes; the dimension of features is lower; only uses color features.
Jiang et al. [118]	Image feature representation by local regions for retrieval.	More efficient image representation than the global feature.	Cannot represent objects with distortions and noisy backgrounds; low dimension of features.
This paper	Image classification by local kernel.	Efficient similarity metric; high generality; plenty features.	The computational efficiency can be further improved.

5.3.1. Image Segmentation and Feature Extraction

- Image Segmentation:** As the first step of local feature representation, the images are segmented into regions. A good segmentation method may be able to provide regions close to the semantic objects in an image, and it is crucial for the classification performance. This paper uses 3 different segmentation approaches for the proposed method, and compares their performance in experiments. The first method used is the

region growing algorithm [34], a region-based image segmentation which examines the neighboring pixels of the ‘seed point’ and determines if the pixel should be added to the seed point by region homogeneity criteria. As more pixels agglomerated into the seed, a homogenous segmented region is generated. The second method used is the watershed segmentation [36], a segmentation method using the watershed transform to obtain the watershed lines of the image. Then the pixels draining from the line to a common minimum form a watershed basin, which represents a segment. The above two are both region-based segmentation approaches. The third one is called the normalized cut [39], a contour-based segmentation approach, realized by forming the region by minimizing the weights of edges. The above three methods are run for providing local regions for the proposed method, and their performances are shown in Fig. 5.3 and Table 5.2.

- **Region Feature Extraction:** We use features covering a wide range of image region properties. Totally 219 feature dimensions are extracted from segmented regions. The feature is composed of following properties.

- 1) The RGB components of color space, with local region’s mean and standard deviation for each component, so it has a total of 6 dimensions.

- 2) The HSV components of color space, with the same kinds of features as RGB, so it also has 6 dimensions.

- 3) The 8-dimension texture feature derived from the mean and standard deviation of the filtered image/region by Gabor filters at 4 orientations in steps of 45 degrees.

- 4) The 7-dimension shape feature derived from the first 7 central geometric moments of the image/region.

5) The 192 dimensional quantized color histogram for RGB components, with 64 dimensions for each component.

A 219 dimensional feature vector is assigned to each region as the raw feature data.

5.3.2. Local Feature Selection

The local feature selection block in Fig. 5.1 selects the best feature subset, whose corresponding classification result is the output as the image classification accuracy. The realization of the feature selection block is composed of feature space search strategy, and the evaluation measure.

- **The Feature Space Search:** This paper uses the sequential forward floating search (SFFS) [101] as the search strategy. SFFS is an iterative process starting with an empty feature subset ($X' = \phi$). At each iteration, one feature is chosen among the remaining m -dimensional feature space $X = \{x_i\}, i = 1, \dots, m$ and it is added into the subset. To determine which feature to add, it tests the performance of every addition of feature from the remaining features, by an evaluation measure $eva(\cdot)$, and select the one obtaining the highest performance for the new subset. The above process, called the forward search, is shown as follows [9].

$$\begin{aligned}
 X' &= X' \cup \{x_i \in X \mid \max(eva(X' \cup x_i)), i = 1, \dots, m\} \\
 X &= X - x_i
 \end{aligned}
 \tag{5.1}$$

A feature is incrementally added to the subset by the above process. At each iteration, a backward search is also performed by deleting a feature, after which the remaining subset reaches the highest improvement of performance evaluation, compared to the subset before deletion. The backward search is realized by the process below.

$$\begin{aligned} X' &= X' - \{x_i \in X \mid \max(\text{eva}(X' - x_i) - \text{eva}(X')), i = 1, \dots, m\} \\ X &= X + x_i \end{aligned} \quad (5.2)$$

In each iteration, equations (5.1) and (5.2) are run sequentially. Normally, the process terminates when no additional features could result in an improvement in accuracy or the feature subset already reaches a predefined size.

- Integrating Local Kernel with Bayesian Classifier:** The evaluation measure of feature subset can be different criteria, as introduced in Section 5.2.1. This paper uses the Bayesian classifier as the evaluation. The Bayesian classifier estimates the class label of a feature vector (extracted from region features in this paper) by finding the highest probability among the trained Gaussian model of each class. From the comparison of the Bayesian classification results with the actual image class labels, an estimate of the correct classification rate (CCR) is obtained as the feature subset evaluation. This procedure is repeated several times to get the repeated mean s -fold cross-validation correct classification rate (MCCR) [101], which is output as the final evaluation measure. The parameter s in s -fold means that the dataset is divided into s equal sets, and the Bayesian classifier uses 1 set as the testing data and the remaining $s-1$ sets as the training data. The classification is repeated s times for each of s subsets as the testing data, and the average CCR constitutes the MCCR. In each iteration, one feature is added to the subset by which the new subset has the highest MCCR. Value of s is set to 10 in this paper.

In this paper, a local kernel is integrated into the Bayesian classifier as the evaluation measure, to enforce the local features to guide the feature selection procedure. The local kernel is realized by the integrated region matching (IRM) [113], a region-based similarity measure between two images, attempting to overcome the

deficiencies of global image distance by representing images at the object level. Assume that images I_1 and I_2 are represented by two region sets $I_1 = \{r_1, r_2, \dots, r_m\}$ and $I_2 = \{r'_1, r'_2, \dots, r'_n\}$. The IRM distance between two images is the summation of all the weighted region distance,

$$dist_{IRM}(I_1, I_2) = \sum_{i,j} s_{i,j} d_{i,j} \quad (5.3)$$

where $d_{i,j}$ is the distance between regions r_i and r'_j of two images, which is the Euclidean distance used in this paper, and $s_{i,j}$ is the distance weights between them. The larger weights indicate the importance of the two regions in measuring the similarity. Mathematically, the distance weights are proportional to both the region importance values p_i and p'_j , for which a larger region has a higher region importance value. The equations are,

$$\begin{aligned} \sum_{j=1}^n s_{i,j} &= p_i, i=1, \dots, m \\ \sum_{i=1}^m s_{i,j} &= p'_j, j=1, \dots, n \end{aligned} \quad (5.4)$$

The above equations mean that the sum of the distance weights of a region is equal to its importance value, and the distance of a region can be measured to multiple regions of the other image. As a result, important regions contribute significantly in computing the distance. Examples of important regions with their values (p) are shown in Fig. 5.3.

The region-based distance metric is integrated into the Bayesian classifier as a local kernel, to guide the feature selection. In this aspect, the distance metric (see Equation

(5.3)) is integrated into the Gaussian kernel of the Bayesian classifier, by using the following equation,

$$f(x) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot V_dist_{IRM}^T(x, \mu) \cdot \Sigma^{-1} \cdot V_dist_{IRM}(x, \mu)\right) \quad (5.5)$$

where the IRM distance is used instead of the global image distance in Gaussian kernel.

The term $V_dist_{IRM}(x, \mu)$ is an IRM distance vector for k dimensional feature space,

$$V_dist_{IRM}(x, \mu) = \{dist_{IRM}(x_1, \mu_1), \dots, dist_{IRM}(x_k, \mu_k)\} \quad (5.6)$$

where $x_i, \mu_i, i=1, \dots, k$, are components of k -dimensional region vectors of testing image x and the cluster μ of the trained Gaussian model, respectively. The IRM distance is computed for each feature dimension of the data, and for a specific dimension i , the distance $dist_{IRM}(x_i, \mu_i)$ is computed by Equation (5.3). The testing image x is composed of segmented region feature vectors to compute IRM distance.

Now the remaining problem is how to assign region vectors to the training cluster μ ? In this paper, as shown Fig. 5.2, only the regions (black dots) within the ellipsoid, centered at the mean feature point of the training cluster, are picked as the regions for μ to compute the IRM distance. This method makes the IRM distance computationally fast and efficient. The effective radius R of the ellipsoid is computed by the distribution of the training region clusters. Let us assume that the regions in a cluster of the training data (as in Fig. 5.2) follow a Gaussian distribution and we take β as a significance level. For the given significance level, $100(1-\beta)\%$ of the regions in the cluster will fall inside the ellipsoid and follow a $\chi^2(\beta)$ distribution with p degrees of freedom. The radius of the

ellipsoid of a training cluster can be calculated as [17],

$$R(\text{cluster}_i) = \gamma \cdot \frac{n-1}{n-p} \cdot F_{p, n-p}(\beta) \quad (5.7)$$

where γ is the region filtering parameter, set to 0.2 in this paper, and p is the dimension of the feature data, and F is the feature space. However, all the region features in the training clusters are used to compute the k by k covariance matrix Σ . Integration of the region-based distance into the Gaussian kernel of the Bayesian classifier, allows the combination of feature selection and local feature to improve the image classification. The final image classification result is the local kernel integrated Bayesian classification accuracy obtained by the optimal selected feature subset.

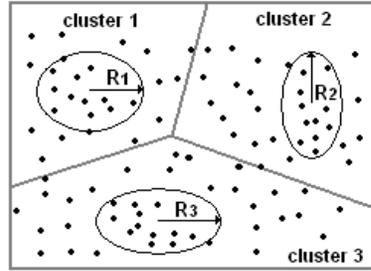


Fig. 5.2 Sampling of regions for each cluster, dot points: training sample of regions.

5.3.3. Multiple Kernel Learning for Classifier Selection

A multiple kernel learning is performed to learn the parameters of multiple Bayesian classifiers, each of which is individually trained in Section 5.3.2. For a testing image x , we employed a so-called kernel function $y(x)$ (see Equation (5.8)), which intuitively selects an optimal classifier for a specific testing data.

$$y(x) = \underset{c=1, \dots, C}{\operatorname{argmax}} [\beta_c \cdot (f_c(x) \cdot a_c + b_c)] \quad (5.8)$$

where C is the number of candidate Bayesian classifiers, and β_c , a_c and b_c are parameters with specific values learned for each classifier. The parameters satisfy the constraints,

$$\sum_{c=1}^C \beta_c = 1, \quad \sum_{c=1}^C a_c = 1, \quad \sum_{c=1}^C b_c = 1$$

$f_c(x)$ is the individual probability computed by the Bayesian classifier as shown in Equation (5.5). The weighted linear function of the posterior probability enables to put different weights on classifiers with different performance during the training session. The kernel function aims to select an optimal weighted linear posterior probability from one of the multiple classifiers. The three parameters are learned during training session, using the multi-objective optimization by NSGA-II [31] with the three constraints. In this respect, totally $3C$ parameters are learned with C multi-objectives.

5.4. Experiments

In experiments, we show the performance of our approach (local kernel integrated feature selection) with various image segmentation and feature selection approaches applied into diagram of Fig. 5.1. The three segmentation approaches used are:

(1) region growing [34], (2) watershed [36], (3) normalized cut [39]. And the three feature selection methods used are:

(1) sequential forward floating search (SFFS) evaluated by s-fold cross-validation Bayesian classifier [101],

(2) feature selection evaluated by max-dependency, max-relevance and min-redundancy (mRMR) [107],

(3) feature selection evaluated by entropy [114].

5.4.1. Datasets

The proposed method is applied to the following two image databases,

(1) Caltech-101 database [42]: It is composed of images of objects belonging to 101 categories, for 40 to 800 images per category. The size of each image is roughly 300 by 200 pixels. We use all the images from the database for the categories shown in Fig 5.3.

(2) Butterfly database (<http://janzen.sas.upenn.edu/>): It contains 30 classes with a total of 7600 images of butterflies. Some of images are shown in Fig. 5.3.

Database		Butterfly						Caltech-101					
Class		1		3		14		faces		cougar_face		dalmatian	
(a) Original Image													
(b) Region Growing [34]	Segm-entation												
	Regions with high p values	 0.03	 0.01	 0.03	 0.01	 0.03	 0.01	 0.01	 0.001	 0.054	 0.07	 0.22	 0.02
(c) Watershed [36]	Segm-entation												
	Regions with high p values	 0.042	 0.01	 0.01	 0.01	 0.03	 0.01	 0.01	 0.001	 0.030	 0.044	 0.083	 0.02
(d) Normalized Cut [39]	Segm-entation												
	Regions with high p values	 0.035	 0.02	 0.03	 0.03	 0.02	 0.01	 0.01	 0.001	 0.042	 0.047	 0.069	 0.02

Fig 5.3. Segmentations and important regions (with P values) for the two databases, (a) sample images with class names, (b) region growing segmentation, (c) watershed segmentation, (d) normalized cut segmentation.

5.4.2. Algorithms Compared and Parameters

The parameters are related to three segmentation methods used, as shown below,

(1) Region growing: thresholds for pixel aggregation and region merging [34], set to 0.03 and 35, respectively.

(2) Watershed [36]: threshold for region merging, set to 35.

(3) Normalized cut [39]: number of segments, set to 24.

Note that the above parameters are fixed for all images, and they are obtained by performing experiments. Another group of parameters are learned in Section 5.3.3.

5.4.3. Results of the Proposed Approach

We integrate local kernel in Bayesian classifier to three segmentation and three feature selection approaches. Results for three segmentation approaches and their significant regions for local kernel integration are shown in Fig. 5.3. The quantitative results for the different segmentation and feature selection methods with local kernel integration, are shown in Table 5.2. Fig. 5.3 shows segmentation results and some regions that are important, i.e., they contribute significantly in computing the region-based image distance for local kernel. Important regions include the human and dog faces (in Caltech-101 database) that are the most discriminative regions to classify the categories of ‘dalmatian’ from others. Large coherent patterns on the wings of the butterfly database are also obtained as significant regions. The final image classification results in Table 5.2 are obtained at the peak accuracy from the feature selection procedure.

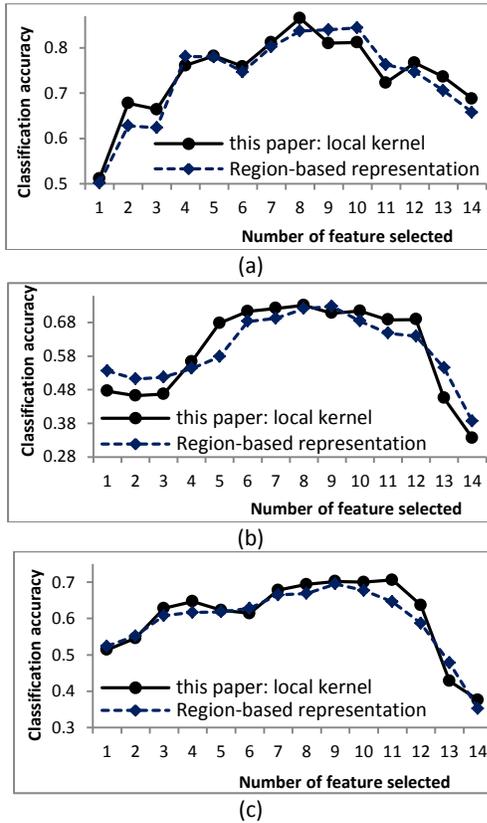


Fig. 5.4. Local kernel method compared with local feature selection method, for Caltech-101 database, with feature selection methods, (a) SFFS-Bayesian, (b) mRMR, (c) entropy.

5.4.4. Local Kernel for Feature Selection: Compare with Local Feature Selection

In this section, the proposed method using local kernel for feature selection, is compared to another local feature selection method [118] (introduced in Table 5.1). The method of [118] uses region-based image representation (global image feature represented by accumulation of weighted local region features, where the weights are obtained by the area of the regions) for feature selection. The region-based image representation in [118] is integrated into the ‘evaluation measure’ block in Fig. 5.1, instead of the local kernel, to run the feature selection. Fig. 5.4 shows the comparison

between the two methods in different feature selection iterations. The two approaches are run under three feature selection systems. For example, in Fig. 5.4 (a), our method by local kernel (peak accuracy of 86.5% at 8th feature), outperforms that of [118] (peak accuracy of 84.4% at 10th feature). The local kernel method has the higher performance under all the three feature selection systems.

5.4.5. Local Kernel for Feature Selection: Compare with Global Feature Selection

The benefits of the proposed method against the other three global feature selection methods are shown in Fig. 5.5 for the two datasets, where the classification results are displayed for incremental feature selection iterations. The three global feature selection methods compared are:

- (1) SFFS-Bayesian [101]: It is the most recently published feature selection method using the sequential forward floating search (SFFS) evaluated by Bayesian classifier.
- (2) mRMR [107]: It is an improvement of the current mutual information evaluated feature selection methods, by using the max-dependency, max-relevance and min-redundancy as the combinatorial evaluation criteria.
- (3) Entropy [34]: Feature selection using entropy as the evaluation measure.

All of the above methods use global features for classification. Note that the proposed method uses region growing as the segmentation scheme, and integrates the local kernel into feature selection by SFFS-Bayesian [101]. Fig. 5.5 only shows the first 14 iterations from total 219, since the classification accuracies of all four methods start to decrease after 14 iterations. The peak accuracy is regarded as the final image classification results, with corresponding optimal feature subset. It can be seen in Fig. 5.5 that our method

outperforms other global feature selection methods in most feature selection iterations, and reaches the highest peak classification accuracy. For example in Fig. 5.5(a), our method reaches the peak accuracy of 86.5% in the 8th dimension of selected subset, whereas the global SFFS-Bayesian only obtains 84.7% accuracy on 11th dimension, a larger subset compared to our method. The same cross validation Bayesian classifier is used in all the four methods to obtain the classification results.

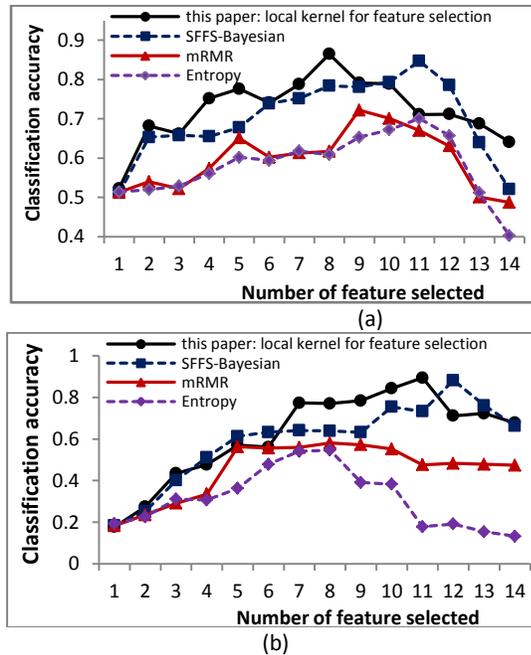
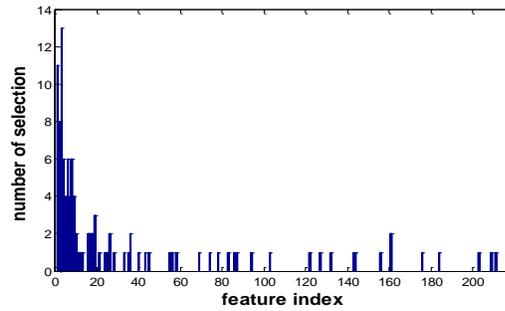


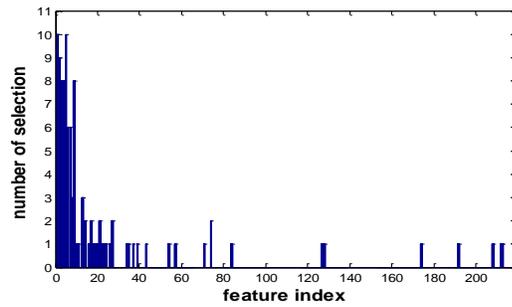
Fig 5.5. Comparison of classification results with feature selection iterations, (a) Caltech-101 database, (b) Butterfly database.

Note that the only difference between our method and the global SFFS-Bayesian method comes from the fact that our method uses the local feature, instead of the global feature used by SFFS-Bayesian. As a result, the classification improvement by our method only comes from the use of local feature. The reason of the classification improvement of our method comes from the region-based distance metric for local kernel, by which the important regions (see Fig. 5.3) have more distance weights and p values,

thus, contributing more to image distance measure. The important regions always represent distinct objects, which are useful to discriminate different image classes. For example, as the human face region shown in the ‘faces’ category in Fig. 5.3, it is assigned with larger weights and p values (0.009 at region growing) in image distance measure.



(a)



(b)

Fig 5.6. Statistics of selected features for the results shown in Table 5.2, for (a) Caltech-101 database, (b) Butterfly database.

5.4.6. Results of the Proposed Method with Various Segmentation and Feature Selection Methods

Table 5.2 shows a more detailed comparison of the final image classification results, among all combinations of the related methods, with the corresponding dimension and contents of selected feature subset. All the methods using local kernel use the Bayesian classifier for classification, but with different segmentation and feature selection procedures. The methods using local kernel, outperform those of the same feature

selection methods using global features. The region growing and watershed segmentations provide the highest performance, and both largely outperform the normalized cut segmentation. Note that the most selected features with top classification performance, are from the color features (first 12 features), like the 3rd feature (the blue component of region color) selected first by most of the methods in Caltech-101 database.

Fig. 5.6 shows a statistical result for the selected features in Table 5.2.

Table 5.2. Comparison of the final classification results, with corresponding feature subset dimensions and contents.

Approaches			Caltech-101		
Feature form	Feature selection	Segmentation	Classification results	Selected feature dimension	Contents of feature subset
Local feature with multiple kernels	SFFS-Bayesian	Region growing	86.7%	8	3, 1, 6, 2, 176, 122, 19, 33
		Watershed	86.4%	8	3, 12, 1, 17, 9, 2, 86, 144
		Normalized cut	84.9%	8	1, 26, 3, 8, 45, 209, 132, 2
	mRMR	Region growing	73.6%	8	3, 1, 4, 36, 18, 2, 69, 5
		Watershed	73.1%	9	1, 83, 19, 5, 3, 161, 40, 25, 6
		Normalized cut	72.2%	7	3, 1, 4, 36, 18, 2, 9
	Entropy	Region growing	70.9%	11	94, 3, 8, 103, 4, 1, 9, 2, 16, 5, 26
		Watershed	71.3%	11	3, 1, 6, 9, 2, 35, 55, 10, 184, 4, 17
		Normalized cut	70.3%	11	3, 17, 10, 43, 87, 4, 8, 1, 56, 7, 6
Global feature with multiple kernels	SFFS-Bayesian	NA	85.4%	11	3, 7, 6, 2, 58, 143, 161, 74, 28, 13, 203
	mRMR	NA	72.5%	9	3, 19, 1, 4, 8, 11, 9, 127, 24
	Entropy	NA	70.3%	11	3, 211, 4, 16, 21, 9, 7, 6, 156, 78, 5

Approaches			Butterfly		
Feature form	Feature selection	Segmentation	Classification results	Selected feature dimension	Contents of feature subset
Local feature with multiple kernels	SFFS-Bayesian	Region growing	89.5%	8	2, 1, 7, 192, 3, 16, 14, 5
		Watershed	89.5%	8	2, 1, 7, 192, 3, 16, 14, 5
		Normalized cut	89.6%	10	6, 2, 3, 1, 4, 39, 20, 128, 11, 57
	mRMR	Region growing	88.5%	11	19, 2, 5, 13, 10, 27, 4, 208, 3, 6, 37
		Watershed	60.4%	8	4, 8, 5, 3, 14, 18, 9, 1
		Normalized cut	58.6%	9	5, 4, 2, 17, 6, 9, 1, 74, 21
	Entropy	Region growing	58.8%	9	6, 5, 74, 35, 7, 1, 2, 84, 9
		Watershed	55.2%	8	5, 7, 43, 13, 9, 2, 26, 4
		Normalized cut	54.6%	8	2, 7, 6, 9, 24, 1, 71, 213
Global feature with multiple kernels	SFFS-Bayesian	NA	54.8%	6	9, 7, 1, 3, 2, 5
	mRMR	NA	88.4%	12	5, 9, 3, 21, 27, 13, 54, 1, 4, 127, 2
	Entropy	NA	58.0%	8	5, 8, 1, 3, 22, 34, 6, 4

5.4.7. Statistical Validation of the Results

The proposed method is evaluated with a 10-fold statistical validation on the two image databases. The results are shown in Table 5.2. For the proposed method, the mean accuracy of the classification results are 86.7% and 89.5%, respectively, with the standard deviation of 2.30% and 1.84%. All the other methods in Table 5.2 have standard deviation of 2.37%~5.90%, whose range is higher beyond those of the proposed method, except in only one case whose standard deviation is 2.02%, but its classification accuracy is only 71.3% (see the Local feature with Entropy and Watershed, on Catech-101 in Table 5.2).

5.4.8. Discussion of the Results

Based on the above experimental analysis, the proposed method makes the following contributions:

- (1) The proposed method has higher classification accuracy than any of the other methods (see Table 5.2),
- (2) The proposed method obtains lower standard deviation in statistical validation than any other methods, except in only one case (see Section 5.4.7),

The proposed method achieves the performance with the lowest number of feature subsets, except for only one case in each of the two dataset (see Table 5.2). But these two exceptional cases have much lower classification accuracy.

5.5. Conclusions

In this chapter, we presented a new approach for the local feature selection with multiple kernel learning. The proposed approach combines feature selection and local

feature information by integration of region-based image similarity metric into the Gaussian kernel of a Bayesian classifier, to guide the feature selection for an improved classification. We performed experiments on different standard image databases to indicate the benefits of the proposed method. We showed the improvement in image classification performance over the latest methods. Our future work will focus on integrating the proposed approach with image retrieval.

Chapter 6

Incremental Ensemble Learning for Classification

6.1. Introduction

Recently, with the rapid development of cheaper and better sensors, the volume of digital images has dramatically increased, and it is expected to grow exponentially. As a consequence, automated image classification has become one of the topics of significant interest in computer vision and pattern recognition. In this respect, the development of image classifier has been intensely researched [119-122]. Most of the current work pays significant attention to classifier learning, classification combination, image data mining, and image feature extraction, etc., all of which aim to develop a robust learning system to improve the image classification accuracy.

The performance of image classification is commonly hindered by the problems associated with feature redundancy, curse of dimensionality, insufficient training data, and high misclassification variance. All of these items belong to the long standing pattern recognition and data mining problems. The ensemble classification [123-126, 132, 142] is a feasible solution of the problems related to the design of a classifier. It has been shown to be very successful in creating more accurate classification results from a set of classifiers. Ensemble from multiple classifiers, referred individually as base classifiers, can offer complementary information about the data to be classified, and improve the effectiveness of the overall classification system. An ensemble classifier often has a better performance than any of the single learned classifiers in the ensemble. In the literature, the improvement by using ensemble mainly comes from the voting methods such as bagging and boosting [123], realized by selecting the majority of class labels from the classifiers in an ensemble. This leads to eliminate the effect of noisy misclassifications and decrease the classification variance of a single classifier. Furthermore, negative effect of insufficient training data is alleviated by an ensemble of multiple training sets.

The feature selection [96, 101, 133, 143] is the technique of selecting a subset of features which leads to a more effective classification than using the entire set of features. Guided by an evaluation criterion like the minimum classification error or the minimum mutual information, the reduced dimension subset has a better classification performance than the original entire feature set. The feature selection also solves the problems of curse of dimensionality and feature redundancy.

Thus, ensemble classification and feature selection are two robust tools to alleviate

the problems associated with classification performance. The combination of ensemble classification and feature selection [124-125, 126] has been a new research area. It is expected to obtain a higher performance. In this paper, multiple feature selection schemes train different classification models, which are ensembled to obtain improved classification accuracy.

The combination of ensemble classification and feature selection solves problems associated with image classification. However, an important aspect is often ignored by most of the previous work, where more attention is paid on building an efficient and robust classification model rather than handling the image data itself in an intelligent manner. The reason is that the performance of a classifier not only depends on the learned classification model, but also it is highly data dependent. As a result, the assignment and handling of image data, especially the training data, should be put on a high priority in the learning of an image classifier. In this paper, a discriminative assignment of the training data helps to train more effective classifiers. It is realized by putting the newly added training data into the nearest cluster of the training dataset. A parallel feature selection scheme is used to derive parallel feature subsets. These features are feedback in a closed loop to incrementally assign the training data. Finally, the optimal feature subsets, along with the incrementally assigned training data, are used to learn multiple classifiers to perform the ensemble classification.

The outline of the paper is as follows. Section 6.2 gives an overview of related work and provides the contributions of this paper. Section 6.3 describes the technical approach in detail. Section 6.4 provides the experimental results, discussions and

comparisons. Finally, conclusions are given in Section 6.5.

6.2. Related Work and Contributions

6.2.1. Related Work

Ensemble of learned models or classifiers [123-125, 132, 142] has been one of the main directions in machine learning and data mining. Ensemble allows us to achieve a higher final classification accuracy by means of combining classification labels from different single classifiers to obtain a final classification decision. Two essential questions exist in an ensemble approach:

- (1) How to generate the base classifiers of an ensemble?
- (2) How to integrate individual classifications of the base classifiers into one final classification decision?

With regard to question (1), this paper accepts the Bayesian classifier as the base classifier of an ensemble. The Bayesian classifier [125] is recognized as the optimal nonlinear probabilistic classifier with stable classification performance property, and the misclassification variance among single Bayesian classifiers is much lower than SVM [129] and EM [130] classifiers. The performance of a single Bayesian classifier can be improved by bias reduction ensemble techniques like bagging and boosting. Not that the feature selection performance in this paper is improved by taking advantage of the Bayesian performance for an optimal and robust feature subset evaluation criterion.

With regard to question (2), the bagging and boosting [123] are two of the most effective integration techniques for ensemble, by means of voting for a final decision from the majority of classification labels from ensemble classifiers. In this paper, an

advanced boosting technique called Arc-x4 [123], which is an improvement of Adaboost [123] algorithm, is used as the ensemble technique.

The combination of ensemble classification and feature selection, called the ensemble feature selection [124-126], is a relatively new area. Santana et al. [124] provide a comparative analysis of feature selection with different ensemble methods. In this paper, an ensemble of multiple Bayesian classifiers is used, and feature selection is integrated to select optimal feature sub-dimension set to learn each classifier models.

As stated previously, the performance of a classifier is highly data dependent. Very limited work has been done for designing the distribution of training data. Masud et al. [127] train a classifier on limited labeled training data. Cervantes [128] compute reduced training data to learn a support vector machine (SVM) classifier. How to learn a classifier under condition of limited or reduced training data, dominates this research area. In this paper, the multiple classifiers in the ensemble are able to use different training datasets that are incrementally distributed by the proposed approach. Thus, both existing and newly added training data are distributed incrementally to build multiple training sets used to learn the ensemble classifiers.

The incremental learning based on methods resulting in learned models that improves performance over time or iterative loops, have attracted attention in pattern recognition and machine learning [139-140]. Basically, the incremental scheme is based on an iteratively improved learning model, e.g., the classifier or the recognition machine, derived by the improvement from later steps of the system. In this paper, a new incremental learning scheme is realized by the following two steps,

- (1) The multiple sets of training data are discriminatively assigned for a more effective training session, and they are used for the parallel feature selection in (2);
- (2) The feature subsets obtained from the parallel feature selection, are feedback to the training data assignment in (1), to incrementally assign a more compact training data.

The above loop incrementally and iteratively learn a more discriminate training dataset, along with more effective feature subsets, both of which are used to obtain an improved classification accuracy.

6.2.2. Contributions

In the proposed method, the ensemble feature selection is performed using the incrementally distributed training data, to build a robust learning system of image classifiers. As a result, the paper makes the following contributions:

- (1) A new method called the incremental ensemble feature selection is proposed, which performs ensemble classification learned on the incrementally distributed training data;
- (2) Both existing and newly added training data can be discriminatively assigned to the current training sets, and this characteristic improves the performance of the classifiers;
- (3) The multiple training sets are incrementally assigned, in an iterative loop, by the feature sets from the parallel feature selection.
- (4) The proposed method outperforms other feature selection and ensemble learning methods, when run on publically available datasets. Also the hypothesis test indicates the statistical significance of the method.

6.3. Technical Approach

The overall system diagram of the proposed method is shown in Fig. 6.1. The

image database, composed of multiple image classes, is divided into the training and testing data. Then the training data are discriminatively assigned to build multiple training subsets which are further used in parallel feature selection to train multiple classifiers. During the training, each of the classifiers is also trained by one of the parallel feature selection schemes, and different feature subsets are selected for different classifiers. The feature subsets are feedback into the assignment of training sets, to incrementally assign more effective training data. After the training session, multiple classifiers are trained with different feature subsets. During the testing, the testing image data are classified by different classifiers, trained by incrementally assigned training sets, using multiple optimal feature subsets selected from the parallel feature selection scheme. The final classification result is derived from the ensemble of the classification decisions from multiple classifiers. In the following we describe the details of the approach.

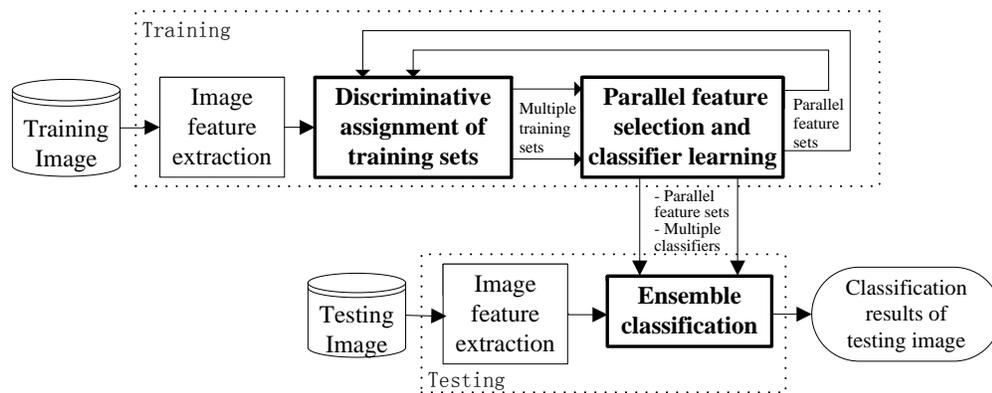


Fig 6.1. The overall system diagram.

Table 6.1. Symbols and their meaning used in this section.

Symbols	Meanings
Tr	The entire training set.
Tri, Tr_i	Distributed training set for individual classifier.
x	Individual training data.
λ	Individual testing data.
y	Class labels.
I	Individual classification inducer.
C	Classification model trained by inducer.
$C_y(\bullet)$	Center of the i th training set.
$Dist(\bullet, \bullet)$	Euclidean distance of two data.
eva	Feature subsets evaluation results for feature selection.
acc	Feature subsets accuracy.
div	Classification diversity
N	Number of multiple training sets/classifiers.

6.3.1. Image Feature Extraction

Before the learning of a classifier, a feature vector is extracted from each image. We use features covering a wide range of image properties. In total, 219 features are extracted from an image to build an image feature database. The feature data is composed of the following properties.

- 1) The RGB components of color space, with image's mean and standard deviation for each component, so it has a total of six dimensions.
- 2) The HSV components of color space, with the same kinds of features as RGB, so it also has six dimensions.
- 3) The 8-dimensional texture features from the mean and standard deviation of the filtered image by Gabor filters at 4 orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) in steps of 45 degrees.
- 4) The seven-dimensional shape feature derived from the first 7 central geometric moments of an image (Y component of an RGB image).
- 5) The 192 dimensional quantized color histogram for RGB components, with 64

dimensions for each component.

After the feature extraction, a 219 dimensional feature vector is assigned to each image as the original raw feature data. It is used for the parallel feature selection to select different feature subsets to build multiple classifier models.

6.3.2. Discriminative Assignment of Training Data

In order to realize the ensemble of classifiers, the entire training data is divided into multiple sets to learn different classification models. Traditionally, the training data is divided equally and randomly into multiple subsets, but this method is inefficient ignoring the inner distribution of training data. In this paper, a discriminative assignment of training data is obtained to build multiple training sets with more effective distribution.

Firstly the entire training data is divided (half by half) into two initial sets $Tr1$ and $Tr2$. For the first initial set $Tr1$, it is further divided into N subsets $\{Tr1_1, \dots, Tr1_N\}$ to provide the training subsets to learn N multiple classifiers. Then for each data $Tr2_d$ in the other training set $Tr2$, we first obtain the class label of $Tr2_d$ as y . Subsequently, we find the subset $Tr1_s$, whose cluster center of class y is closest to the data $Tr2_d$. The data $Tr2_d$ is added to the subset $Tr1_s$. Each data in $Tr2$ is iteratively added into one of the subsets of $Tr1$ using this incremental method. The key idea of this method is that the training data can be incrementally added into the subset with minimum distance to the cluster center, so the cluster distribution of training data is more compact. Note that the newly added training data can be regarded as new data of $Tr2_d$, and it can be added by the above method to the subset with the closest cluster distance. The above method also enforces a more effective training of the classification model. After the incremental

distribution, the N training data sets are input into parallel feature selection to learn N classifiers, as shown in Fig. 6.1. The pseudo code of the algorithm is given in Fig. 6.2.

Input: Entire training set Tr , integer N (number of multiple training subsets)

1. $Tr = Tr1 \cup Tr2$; (divide entire training data into two initial sets, half by half, randomly)
2. $Tr1 = \bigcup_{i=1, \dots, N} Tr1_i$; (divide training set $Tr1$ into N multiple subsets, as initial training sets to learn multiple classifiers)
7. For: each of the data $Tr2_d$ in $Tr2$ {
8. $L(Tr2_d)=y$; (y: class label of data $Tr2_d$)
9. $Tr1_s = \arg \min_{i \in \{1, N, \dots\}} \text{Dist}(Tr, d\mathcal{C}_y, T_k, \tilde{x})$;
(find the subset $Tr1_s$, whose cluster center of class y, is closest to the data $Tr2_d$)
10. $Tr1_s = Tr1_s \cup Tr2_d$; (add data $Tr2_d$ into subset $Tr1_s$)
11. Remove $Tr2_d$ from $Tr2$;
12. } While $Tr2$ is not empty;

Output: Incrementally distributed training sets :
 $\{Tr1_1, \dots, Tr1_N\}$.

Fig 6.2. The discriminative training data assignment algorithm.

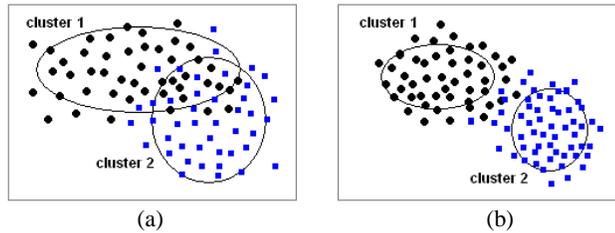


Fig. 6.3. Comparison of training data assignment, (a) random assignment, (b) discriminative assignment.

Fig. 6.3 shows comparison between the discriminative and random assignment of training data, where the discriminative assignment generates the classification model with more compact and discriminative boundary. It means that there is less overlap (over different classes) of the training data in the feature space. This reduces the classification errors. The problem of over-training may occur in some of the individual training sets (decrease the performance of single training data during testing). However, we find that

most of the discriminatively assigned training sets reach higher classification performance, compared to those of randomly assigned sets.

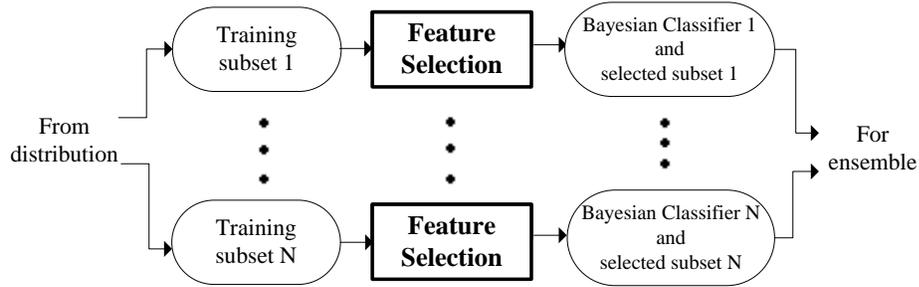


Fig. 6.4. The diagram for parallel feature selection and classifier learning.

6.3.3. Parallel Feature Selection and Classifier Learning

The parallel feature selection block in Fig. 6.1, as shown in detail in Fig. 6.4, starts from different training sets. Each feature selection aims to build an optimal selected feature subset with its trained classifier. The search strategy of feature selection used in this paper is called the sequential forward floating search (SFFS) [101]. In this algorithm, each feature selection scheme starts from an initial feature subset, and sequentially adds features from a total of 219 features introduced in Section 6.3.1, to generate a selected subset. At each iteration, which feature is added into the current subset, is decided by the feature evaluation strategy. The candidate feature with the highest evaluation score, is added to form a new subset. It stops when the adding of a new feature no longer improves the feature subset performance. The optimal feature subset with its learned Bayesian classifier are the output of each parallel feature selection scheme. As shown in Figs. 6.1 and 6.4, different distributed training sets generate a total of N different selected feature subsets, which lead to N different learned Bayesian classifiers that are output from the

parallel selection. Note that each of the N parallel feature selection schemes has a different initial feature subset to start. Thus, N different feature subsets are generated.

In this paper, Bayesian classifiers with cross-validation are used as the evaluation strategy to guide the feature subset selection. In terms of s -fold cross-validation classifier [101], for each feature evaluation, the training set of each of the parallel feature selection schemes is equally divided by s , where 1 set is used as testing data, and the remaining $s-1$ sets are used as training data. There are total of s such classification iterations for a single goodness evaluation of a feature, and the final evaluation score is obtained from the above s independent evaluation results. In this paper, each one of the s evaluations is expressed below [126],

$$eva_i = acc_i + \alpha \cdot div_i \quad (6.1)$$

where $i \in \{1, \dots, s\}$ belongs to one of the s -fold evaluation. The term acc_i is the Bayesian classification result on the i th validation set, and div_i is the classification diversity of the i th validation set, expressed as the standard deviation between i th classification accuracy and the mean accuracy of all s validation sets as, shown below:

$$div_i = \sqrt{E[(acc_i - u)^2]} \quad (6.2)$$

where,

$$u = \frac{1}{s} \sum_{k \in \{1, \dots, s\}} acc_k \quad (6.3)$$

is the mean classification accuracy of all s cross-validation classifiers. The parameter α

in equation (6.1) is the coefficient of the degree of the influence of diversity, set to 0.25 in this paper. The use of extra diversity term div in evaluation is based on the assumption that an effective ensemble should emphasize more on higher-accuracy classifiers that disagree on other predictions.

In this paper, the cross-validation evaluation combines s classifications by means of average accuracy, to generate the final evaluation score:

$$eva_{final} = \frac{1}{s} \sum_{p \in \{1, \dots, s\}} eva_p \quad (6.4)$$

The feature subset that has the highest evaluation by equation (6.4) is selected as the optimal feature subset. At the same time, parallel Bayesian classifiers are learned by using the optimal selected feature subset. As a result, each feature selection scheme (see Fig. 6.4) outputs two results: an optimal feature subset, and a Bayesian classifier learned by this feature subset. During testing in Fig. 6.1, the testing data is classified using the above different classification models, respectively, with the corresponding feature subsets. The final classification decision is obtained from classifications of multiple classifiers using the ensemble based classification. The advantage of parallel feature selection is that, the feature selection from more training sets will obtain better classification accuracy than single classification scheme only from one training set.

6.3.4. The Incremental Learning

Using the discriminatively assigned training sets (see section 6.3.2), the parallel feature selection provides multiple feature subsets (see Section 6.3.3). The incremental learning is realized by a feedback loop, from the parallel feature selection, to the

discriminative training data assignment (see Fig. 6.1).

Input: The individual training set Tr_i ; Classification inducer I ; Integer T (number of combined classifiers in the ensemble)

1. For $i = 2$ to T {
2. Tr_i : training data for the individual ensemble, with instance weight for a training data x set to $(1 + e(x)^4)$, where $e(x)$ is the number of misclassifications made on x by inducer I_1 to I_{i-1} .
3. $C_i = I_i(S')$ where C_i is the classifier model trained by training data S' , made by classifier I_i . }
4. $C_{ensemble}(\lambda) = \arg \max_{y \in Y} \sum_{i: C_i(\lambda)=y} 1$ (assign testing data λ to the most classified label y from all classifier models in the ensemble)

Output: ensemble classification label $C_{ensemble}$

Fig 6.5. The ensemble classification algorithm: Arc-x4.

The multiple feature subsets are feedback into the discriminative training assignment, to incrementally assign more effective training sets. In this aspect, the distance measure (see step 6 in Fig. 6.2) in each of the assignment blocks, uses the feature subset, provided by each of the parallel feature selection schemes. The selected feature set assign more discriminative training sets, which will help to improve the performance of the later steps of feature selection and classifier learning. The incrementally assigned training sets, are further used for the feature selection and classifier learning. At each iteration, the feature selection starts from the entire feature set.

In this incremental loop, the feature selection and training data assignment mutually improves the performance of each other. The loop terminates when the performance of the parallel feature selection (the average performance of parallel feature selection schemes) no longer improves from the previous iteration. In this situation, the

performance of feature selection is measured by the Bayesian evaluation (see Equation (6.4)), directly on the selected feature subsets. After the termination of the loop, the training session outputs the optimal selected feature subsets, along with the multiple classification models, trained by the incrementally distributed training sets. Both of them are used to classify a testing data using the ensemble classification as discussed next.

6.3.5. The Ensemble Classification

The ensemble of multiple learned classifiers (from section 6.3.4) to obtain the final classification of a testing data, is realized by dynamic voting of N results to obtain an accurate classification result. The Arc-x4 algorithm [123], as an improvement of the **ARCing** (**A**daptively **R**esample and **C**ombine) method, is used as the ensemble voting method in this paper. The pseudo code of the algorithm is shown in Fig. 6.5. In the ensemble classification, the number T in Fig. 6.5 equals to the number of learned classifiers by parallel feature selection. The instance λ is the feature vector of a testing image. A final classification label $C_{ensemble}$ of testing data is built by the label y , which is predicted by most of the cross-validated classifiers. This is recognized as the voting process. Like Adaboost [123], the Arc-x4 algorithm sequentially trains classifiers for a number of datasets T , but the training instances x are weighted using a simple scheme: the weight of an instance is proportional to the number of mistakes previous classifiers made to the fourth power, plus one. The Arc-x4 algorithm is applied directly as the ensemble method for integrating the results from cross-validated classifiers. By using the Arc-4x algorithm, the ensemble generates a classification decision which has a higher accuracy than any of the individual classifiers.

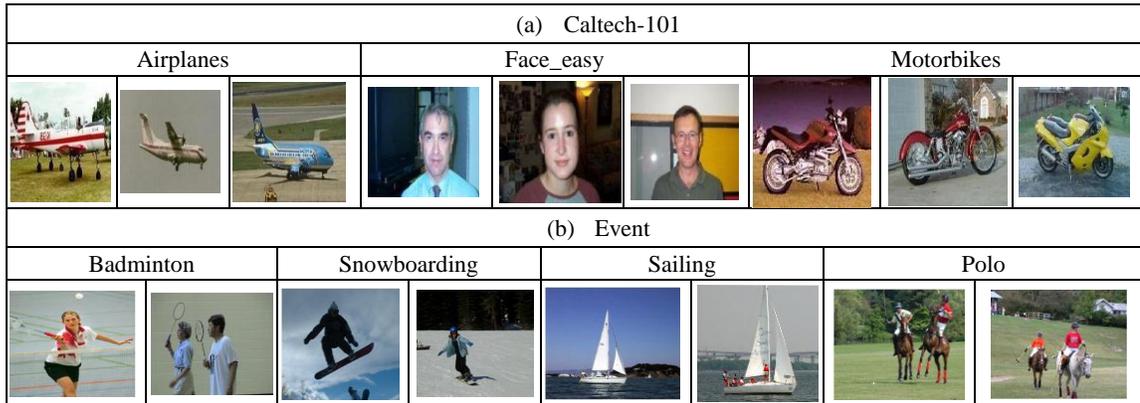


Fig 6.6. Example images of the two image databases.

6.4. Experimental Results

6.4.1. Datasets and Parameters

This paper uses two image datasets for experiments. The first one is called Caltech-101 [134], with images of objects belonging to 101 categories, and about 40 to 800 images per category, with 9185 images in total. Most categories have about 50 images. The size of each image is roughly 300 x 200 pixels. The use of Caltech-101 for image classification is also reported in [136-137]. In this paper, three categories are used, which have more than 450 images, as shown in Fig 6.6(a). The second image database is the event dataset [135] from Stanford University, containing 8 sports event categories with a database of 1586 images. Some of example images are shown in Fig 6.6(b). In both datasets, the training and testing data are divided into two equal subsets.

6.4.2. Incremental Assignment of Training Data

The following experiments show the advantages of using incremental learning method to assign the training data, via the feedback loop described in Section 6.3.4.

Fig. 6.7 shows the comparison among the classification by the following methods,

- (1) The incremental and discriminative training data assignments (the proposed method in Section 6.3).
- (2) The discriminative assignment only (use Section 6.3.2 only, no incremental loop of Section 6.3.4). The discriminative assignment is applied, for a more compact training data (see Fig. 6.3), but it does not use the incremental loop, and only runs the feature selection one time, before the ensemble classification.
- (3) The random training data distribution (no incremental and discriminative distribution). It assigns the training data into multiple training sets randomly.

The other parts of the three methods, including the parallel feature selection and the ensemble classification, remain the same as shown in Fig. 6.1, so the improvement of the proposed method in Fig. 6.7 comes from the incremental learning only. The x-axis of Fig. 6.7 is the number of ensemble classifiers used. Two conclusions can be derived from Fig. 6.7, (a) The discriminative assignment of training data improves the classification accuracy by obtaining a more compact training data distribution; (b) As compared to the discriminative assignment only, the incremental assignment of training data provides higher classification results under most of the values of the number of ensemble classifiers; (c) The classification results improve with the number of classifiers that are ensembled. The results is stabilized after a certain ensemble number. These observations show the benefits of using the proposed ensemble. For the proposed method in Fig. 6.7, the classification result by using only a single classifier is 93.06% (even with feature selection), compared to the peak value of 94.11% under 20 ensemble of classifiers. Note that the classification results are stable with 20 ensemble classifiers.

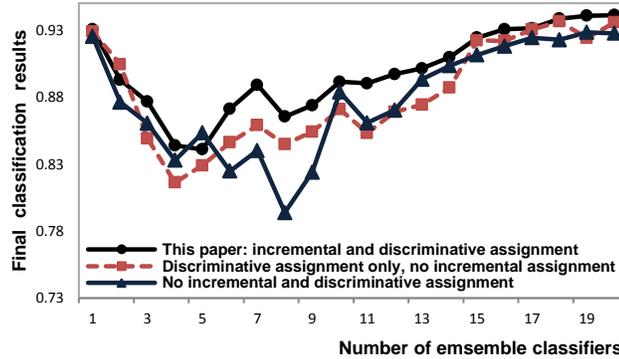


Fig 6.7. Comparison of classification results obtained from different number of ensemble classifiers learned from parallel feature selection, for the Caltech-101 database.

6.4.3. Incremental Ensemble Feature Selection and Classification

In this section we compare the final classification results on the testing data, using different methods. Tables 6.2 and 6.3 show the final classification accuracies of different methods for the two databases.

- **Method #1:** It is the proposed method, with both incremental and discriminative training data assignment, and with feature selection and ensemble classification. It has the optimal classification result of 94.11% for the Caltech-101 database (Table 6.2) under 20 ensemble classifiers (see Fig. 6.7), and 84.3% for the event database (Table 6.3) under 23 ensemble classifiers, respectively. These are the highest accuracy among all the methods. The other methods that are compared include:
- **Method #2:** It is the same as the proposed method, but with no incremental assignment of training data (discriminative assignment only, as displayed in Fig. 6.7). The classification result is 93.57% for the ensemble of 18 classifiers. The improvement of the proposed method compared to this method only comes from the

incremental assignment. Similarly, result of 83.86% is obtained for the event database, lower than that of the proposed method.

- **Method #3:** It is with no incremental and discriminative assignment of training data (only use random assignment, as displayed in Fig. 6.7).
- **Method #4:** The method performs no feature selection (uses the entire features for all procedures), and with the random assignment of training data. Note that the improvement in this method, compared to that of method #6, comes only from the ensemble classification.
- **Method #5:** The method uses no ensemble classification (picks up the best accuracy among the results from all the classifiers), and with the random assignment of training data. Note that the improvement in this method, compared to that of method (6), comes only from the feature selection.
- **Method #6:** Only trains a single classifier, using the entire training set and the entire set of features. It reaches the lowest performance among all the methods compared.
- **Method #7:** It is the same as the proposed method, but with no ensemble classification (picks up the best accuracy among the results of all the classifiers). Note that the improvement of the proposed method, compared to this method, comes only from the ensemble classification.
- **Method #8:** It is the same as the proposed method, but with no feature selection (using the entire set of features for all the procedures). The improvement of the proposed method, compared to this method, comes only from the feature selection.

The methods compared in Tables 6.2 and 6.3 follow the same system diagram in Fig. 6.1,

only with/without realizations of specific procedures above. All the parameters, including the number of individual classifiers, and the number of initial features, are kept the same for all the methods. Note that the proposed method is better than all the other methods.

Table 6.2. Comparison of different methods for Caltech-101 database [134].

Method#	Final accuracy of testing data	Use incremental training assignment?	Use discriminative training assignment?	Use feature selection? (if no: use the entire feature vector)	Use the Ensemble? (if no: pick up the best accuracy)
1	94.11%	YES	YES	YES	YES
2	93.57%	NO	YES	YES	YES
3	92.76%	NO	NO	YES	YES
4	84.87%	NO	NO	NO	YES
5	92.53%	NO	NO	YES	NO
6	83.54%	NO	NO	NO	NO
7	93.48%	YES	YES	YES	NO
8	86.76%	YES	YES	NO	YES

Table 6.3. Comparison of different methods for event database [135].

Method#	Final accuracy of testing data	incremental training assignment?	discriminative training assignment?	feature selection? (if no: use the entire feature vector)	Use the Ensemble? (if no: pick up the best accuracy)
1	84.37%	YES	YES	YES	YES
2	83.86%	NO	YES	YES	YES
3	83.69%	NO	NO	YES	YES
4	79.43%	NO	NO	NO	YES
5	81.12%	NO	NO	YES	NO
6	78.32%	NO	NO	NO	NO
7	83.80%	YES	YES	YES	NO
8	81.68%	YES	YES	NO	YES

6.4.4. Comparison with other Classifier Systems

In the following the proposed method will be compared with other classification methods with totally different system realizations.

6.4.4.1. Comparison with Non-Ensemble Classification Methods

In Table 6.4, the proposed method is compared to the other classification systems different from the system in Fig 6.1. All the methods use the same databases. Others methods compared are:

- (1) The support vector machine (SVM) [129] classifier;
- (2) The expectation maximum (EM) [130] classifier by Gaussian Mixture Model (GMM);
- (3) The naive Bayesian classifier [131].

Each classifier is used by the following two schemes:

- (a) *With feature selection*: The optimal feature subset is selected by [101] to classify the images;
- (b) *No feature selection*: The images are classified using the entire feature set without feature selection.

All of the methods apply feature selection and classification in a single session, without incremental assignment and ensemble. It is evident from Table 6.4 that, the proposed method outperforms all the other methods by a large amount, due to the incremental ensemble feature selection.

Table 6.4. Comparison of the different non-ensemble classification methods.

Image DB	Methods	Classification
Caltech-101	This chapter	94.11%
	SVM [129] (with feature selection)	71.43%
	SVM [129] (no feature selection)	65.56%
	EM+GMM [130] (with feature selection)	74.80%
	EM+GMM [130] (no feature selection)	60.09%
	Naive Bayes [131] (with feature selection)	82.96%
	Naive Bayes [131] (no feature selection)	67.54%
Event	This chapter	84.37%
	SVM [129] (with feature selection)	69.76%
	SVM [129] (no feature selection)	61.33%
	EM+GMM [130] (with feature selection)	70.90%
	EM+GMM [130] (no feature selection)	58.02%
	Naive Bayes [131] (with feature selection)	78.81%
	Naive Bayes [131] (no feature selection)	69.25%

6.4.4.2. Comparison with other Ensemble-based Classification Methods

Table 6.5 shows the comparison among different ensemble methods. The difference

among the listed methods is the ensemble method used, and all the other parts are the same as shown in Fig. 6.1. It is evident that the proposed method outperforms other current ensemble approaches by using the Arc-X4 ensemble technique.

Table 6.5. Comparison of different ensemble methods.

Image DB	Ensemble Methods	Classification
Caltech-101	This paper	94.11%
	Adaboost [138]	94.02%
	Bagging [138]	93.87%
Event	This paper	84.37%
	Adaboost [138]	84.14%
	Bagging [138]	83.82%

6.4.5. Incremental Learning with Additional Training Data

Section 6.3.2 introduced the algorithm for incremental assignment of training data. After training the parallel models, when additional training data are added, the same algorithm of Fig. 6.2 is executed to distribute the new data to the nearest cluster, and with additional incremental loops (section 6.3.4) to build a more effective training data. This procedure makes the proposed method an incremental learning system, where new training data can be added incrementally. Fig. 6.8 shows the performance of final classification accuracy of the Caltech-101 database, when additional training data is added from 4592 to 5100 to 5600 to 6100. The x-axis is the size of training images (added incrementally), and the y-axis is the classification results. The plot starts at the accuracy of 94.11% (see Table 6.2). It is the accuracy achieved by the original training set, which is composed of 4592 images, half of the total number of images in the database. It is clear that as more training data are added, the incremental distribution further improves the clusters of training sets, leading to a higher classification accuracy incrementally.

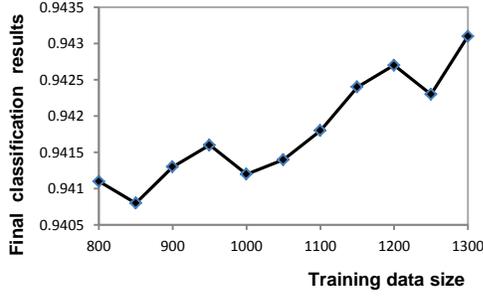


Fig. 6.8. Incremental learning with additional training data.

6.4.6. The Statistical Validation

The hypothesis test [141] is conducted on the results on two datasets to evaluate the statistical significance of the proposed method. Let the classification results of the multiple classifiers (before the ensemble) be the measurement samples of a random variable x . The number of individual classifiers is set to $N=32$ (by the experimental experiences) that has 31 degrees of freedom. To this end we define the test statistic,

$$q = \frac{\bar{x} - \hat{\mu}}{\sigma / \sqrt{N}} \quad (6.5)$$

where \bar{x} is the mean value of samples from multiple individual classifiers, and $\hat{\mu}$ is the hypothesis value for the test, and σ is the standard deviation of the samples, with degrees of freedom $N=31$. Recalling the central limit theorem [141], the probability density function of q under the assumption that *the values of multiple classification results do not differ significantly (they are statistically significant)*, is approximately $N(0,1)$. Let the confidence interval for a significance level ρ be the interval in which the random hypothesis $\hat{\mu}$ (or q) lies with probability $1-\rho$. Table 6.6 shows the hypothesis test results on the two datasets, on the proposed method.

Table 6.6. The hypothesis test of the proposed method.

Dataset	Mean \bar{x}	Standard deviation σ	Hypothesis value $\hat{\mu}$	Significance level ρ	Confidence interval of $\hat{\mu}$
Caltech-101	0.8633 (86.33%)	0.0437 (4.37%)	0.8700	0.05 (5%)	[0.8479, 0.8787]
Event	0.8127% (81.27%)	0.0281 (2.81%)	0.8200	0.05 (5%)	[0.8029, 0.8225]

Based on the criterion of the hypothesis test, in both datasets, the hypothesis value lies within the confidence intervals. As a result, the assumption that *the values of multiple classification results do not differ significantly*, is satisfied. So the classification results of the multiple classifiers of the proposed method are statistically significant. The final ensemble classification result is not derived only from the randomly high accuracies occasionally derived by the individual classifiers, and the performance of the individual classifiers with feature selection schemes is robust with stable accuracies.

6.4.7. Discussions of Results

(1) As shown in Figs. 6.7 and 6.8, and Tables 6.2 and 6.3, the incremental distribution of training data, both for original and newly added, is able to incrementally learn a robust classification system, with improved classification accuracy.

(2) The improvements of accuracy also comes from the parallel feature selection to build different optimal subsets, and the ensemble classification to make optimal decision from multiple classifiers. Tables 6.2 and 6.3 show the numerical improvements for each of the above two procedures.

(3) An incremental feedback loop is added from feature selection to the assignment of training data (see Fig. 6.1) to further refine the training data from information of feature selection, and this loop can be run incrementally.

6.5. Conclusions

This paper provided a new incremental learning of image classifiers, called the incremental ensemble feature selection. The training data is incrementally assigned into multiple training sets to learn different classifiers, which are finally combined to obtain the ensemble classification results. Experimental results on the data from the two challenging image datasets show that the classification accuracy of the proposed method outperforms other current classification schemes. In the future we will evaluate the approach on very large image databases.

Chapter 7

Conclusions and Future Work

In this dissertation, I proposed a high level symmetry feature and feature selection schemes, applied in numerous computer vision and pattern recognition tasks. The symmetry is clearly an important piece of perception puzzle on computer vision. The intense use of symmetry provides guidance on fields of image segmentation and ROI detection in MR images, for an automated and improved performance. Once the symmetry level of the image is defined, the symmetry integration on segmentation, statistical region filtering and region classification are completely enforced for a more effective manner. It always improves the performance of the tasks, on the images with

different symmetry levels (from highly symmetric to totally asymmetric). Meanwhile, the feature selection is used along with local kernel, ensemble learning and user relevance feedback, to provide these methods with more compact and effective feature sets. By understanding and analyzing the feature contents, the selected features with reduced dimensions, help to train the learning models that fundamentally improves the specific computer vision tasks.

In each chapter, thorough experiments are performed with a concrete comparison with state-of-the-art methods. Both visual and numerical comparisons are provided, along with statistical validation procedure to testify the robustness and statistical significance of the methods. The proposed methods outperform other methods, that indicate solid contributions of this dissertation. The use of symmetry and feature selection open a new window on computer vision tasks. In future work, I will integrate them into other tasks such as image enhancement, face recognition and video tracking.

Bibliography

- [1] J. S. Stahl, and S. Wang, "Global Optimal Grouping for Symmetric Closed Boundaries by Combining Boundary and Region Information," *IEEE TPAMI*, vol. 30(3), pp. 395-411, Mar. 2008.
- [2] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Full and partial symmetries of non-rigid shapes," *Intl. Journal of Computer Vision*, vol. 89(1), pp. 18-39, August 2010.
- [3] J. S. Stahl, and S. Wang, "Globally Optimal Grouping for Symmetric Boundaries," *IEEE CVPR*, 2006.
- [4] D. Shen, K. T. Cheung, and E. K. Teoh, "Symmetry Detection by Generalized Complex (GC) Moments: A Closed-Form Solution," *IEEE TPAMI*, vol. 21(5), pp. 466-476, May 1999.
- [5] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Symmetries of Non-rigid Shapes," *IEEE ICCV*, 2007.
- [6] P. J. Giblin, and B. Kimia, "On the Intrinsic Reconstruction of Shape from Its Symmetries," *IEEE TPAMI*, vol. 25(7), pp. 895-911, July 2003.
- [7] G. Marola, "A Technique for Finding the Symmetry Axis of Implicit Polynomial Curves under Perspective Projection," *IEEE TPAMI*, vol. 27(3), pp. 465-470, March 2005.

- [8] A. V. Tuzikov, O. Colliot, and I. Bloch, "Evaluation of the Symmetry Plane in 3D MR Brain Images," *Pattern Recognition Letters*, vol. 24(14), pp. 2219-2233, October 2003.
- [9] B. Combes, R. Hennessy, J. Waddington, N. Roberts, and S. Prima, "Automatic Symmetry Plane Estimation of Bilateral Objects in Points Clouds," *IEEE CVPR*, 2008.
- [10] V. S. N. Prasad, and B. Yegnanarayana, "Finding Axes of Symmetry from Potential Fields," *IEEE Trans. on Image Processing*, vol. 13(12), pp. 1559-1566, December 2004.
- [11] P. Cicconi, and M. Kunt, "Symmetry-Based Image Segmentation," *Proc. SPIE*, vol. 1977, no. 378, pp. 378-384, October 1993.
- [12] T. Liu, D. Geiger, and A. L. Yuille, "Segmenting by Seeking the Symmetry Axis," *Proc. Intl. Conf. on Pattern Recognition*, 1998.
- [13] R. Shor, and N. Kiryati, "Towards Segmentation from Multiple Cues: Symmetry and Color," *Proc. Intl. Workshop on Theoretical Foundations of Computer Vision: Multi-Image Analysis*, 2000.
- [14] A. Gupta, V. S. N. Prasad, and L. S. Davis, "Extracting Regions of Symmetry," *Proc. Intl. Conf. on Image Processing*, 2005.
- [15] T. Riklin-Raviv, N. Kiryati, and N. Sochen, "Segmentation by Level Sets and Symmetry," *IEEE CVPR*, 2006.
- [16] F. Jiao, D. Fu, and S. Bi, "Brain Image Segmentation Based on Bilateral Symmetry Information," *Intl. Conf. on Bioinformatics and Biomedical Engineering*, 2008.
- [17] S. Saha, and S. Bandyopadhyay, "MRI Brain Image Segmentation by Fuzzy Symmetry Based Genetic Clustering Technique," *IEEE Congress on Evolutionary Computation*, 2007.
- [18] F.P.G. Bergo, A.X. Falcao, C.L. Yasuda, and F. Cendes, "FCD Segmentation Using Texture Asymmetry of MR-T1 Images of the Brain," *IEEE Intl. Symposium on Biomedical Imaging*, 2008.
- [19] N. Ray, R. Greiner, and A. Murtha, "Using Symmetry to Detect Abnormalities in Brain MRI," *Computer Society of India Communications*, vol. 31, no. 19, pp 7-10, 2008.

- [20] Y. Liu, K.L. Schmidt, J.F. Cohn, and S. Mitra, "Facial Asymmetry Quantification for Expression Invariant Human Identification," *CVIU*, vol. 91, no. 1-2, pp. 138-159, July-Aug., 2003.
- [21] E. Saber, and A. Tekalp, " Frontal-View Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions," *Pattern Recognition Letters*, vol. 19(8), pp. 669-680, June 1998.
- [22] Q. B. Sun, W. M. Huang, and J. K. Wu, "Face Detection Based on Color and Local Symmetry Information," *Proc. IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 1998.
- [23] J. G. Wang, and E. Sung, "Frontal-View Face Detection and Facial Feature Extraction Using Color and Morphological Operations," *Pattern Recognition Letters*, vol. 20(10), pp. 1053-1068, October 1999.
- [24] W. H. Li, and L. Kleeman, "Real Time Object Tracking Using Reflectional Symmetry and Motion," *Proc. IEEE Intl. Conf. on Intelligent Robots and Systems*, 2006.
- [25] W. H. Li, A. Zhang, and L. Kleeman, "Fast Global Reflectional Symmetry Detection for Robotic Grasping and Visual Tracking," *Australasian Conference on Robotics and Automation*, 2005.
- [26] S. Thrun, and B. Wegbreit, "Shape from Symmetry," *ICCV*, 2005.
- [27] M. Park, S. Lee, P. Chen, S. Kashyap, A. A. Butt, and Y. Liu, "Performance Evaluation of State-of-the-Art Discrete Symmetry Detection Algorithms," *IEEE CVPR*, 2008.
- [28] H. Zabrodsky, S. Peleg, and D. Avnir, "Symmetry as a Continuous Feature," *TPAMI*, vol. 17(12), pp. 1154-1166, Dec. 1995.
- [29] G. Loy, and J. Eklundh, "Detecting Symmetry and Symmetric Constellations of Features," *Euro. Conf. on Computer Vision*, 2006.
- [30] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60(2), pp. Jan. 2004.
- [31] S. P. Kodali, R. Kudikala, and K. Deb, "Multi-Objective Optimization of Surface Grinding Process Using NSGA-II," *Intl. Conf. on Emerging Trends in Engineering and Technology*, Nagpur, India, 2008, pp. 763-767.
- [32] C. Xu, and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *ITIP*, vol. 7,

no. 3, pp. 359-369, Mar. 1998.

- [33] Y. Sun, B. Bhanu, and S. Bhanu, "Automatic Symmetry-integrated Brain Injury Detection in MRI Sequences," *CVPR Workshop on MMBIA*, 2009.
- [34] R. Adams, and L. Bischoff, "Seeded Region Growing," *IEEE TPAMI*, vol. 16(6), pp. 641-647, June 1994.
- [35] D. Comaniciu, and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE TPAMI*, vol. 24(5), pp. 603-619, May 2002.
- [36] S. Beucher, "The Watershed Transformation Applied to Image Segmentation," *Pfefferkorn Conf. on Signal and Image Processing in Microscopy and Microanalysis*, 1991.
- [37] A. R. Smith, "Color Gamut Transform Pairs," *ACM SIGGRAPH Computer Graphics*, vol. 12(3), pp. 12-19, August 1978.
- [38] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative Evaluation of Color Image Segmentation Results," *Pattern Recognition Letters*, vol. 19(8), pp. 741-747, June 1998.
- [39] J. Shi, and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE TPAMI*, vol. 22(8), pp. 888-905, Aug. 2000.
- [40] Y. Sun, and B. Bhanu, "Symmetry Integrated Region Based Image Segmentation," *IEEE CVPR*, 2009.
- [41] A. Hafiane, S. Chabrier, C. Rosenberger, and H. Laurent, "A New Supervised Evaluation Criterion for Region Based Segmentation Methods," *ACIVS*, 2007.
- [42] L. Fei-Fei, R. Fergus, and P. Perona, "One-Shot Learning of Object Categories," *TPAMI*, vol. 28, no. 4, pp. 594-611, April 2006.
- [43] X. Ren, C. Fowlkes, and J. Malik, "Figure/Ground Assignment in Natural Images," *Euro. Conf. on Computer Vision*, 2006.
- [44] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering Structural Regularity in 3D Geometry," *ACM SIGGRAPH*, 2008.
- [45] N. J. Mitra, A. Bronstein, and M. Bronstein, "Intrinsic Regularity Detection in 3D Geometry," *ECCV*, 2010.
- [46] N. J. Mitra, L. Guibas, and M. Pauly, "Symmetrization," *ACM SIGGRAPH*,

2007.

- [47] M. Ovsjanikov, J. Sun, and L. Guibas, "Global Intrinsic Symmetries of Shapes," *Computer Graphics Forum*, vol. 27, no. 5, pp. 1341-1348, July 2008.
- [48] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of Building Facades Using Procedural Shape Priors," *Proc. IEEE CVPR* 2010.
- [49] M. Cho, and K. Mu Lee, "Bilateral Symmetry Detection and Segmentation via Symmetry-Growing," *Proc. British Machine Vision Conference*, 2009.
- [50] J. Liu, and Y. Liu, "Curved Reflection Symmetry Detection with Self-validation," *Asian Conference on Computer Vision*, 2010.
- [51] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. V. Gool, "Computational Symmetry in Computer Vision and Computer Graphics," *Foundations and Trends in Computer Graphics and Vision*, vol. 5, no. 1-2, pp. 1-195, 2010.
- [52] Supplemental material to accompany. This paper will provide a link to this material.
- [53] S. Lee and Y. Liu, "Skewed Rotation Symmetry Group Detection," *IEEE TPAMI*, vol. 32(9). pp. 1659-1672, Sept. 2010.
- [54] S. Lee and Y. Liu, "Curved Glide-Reflection Symmetry Detection," *IEEE TPAMI*, vol. PP, no. 99. pp. 1-13, June. 2011.
- [55] Q. Guo, F. Guo and J. Shao, "Irregular Shape Symmetry Analysis: Theory and Application to Quantitative Galaxy Classification," *IEEE TPAMI*, vol. 32(10). pp. 1730-1743, Oct. 2010.
- [56] A. Hooda, M. Bronstein, A. Bronstein and R. P. Horaud, "Shape Palindromes: Analysis of Intrinsic Symmetries in 2D Articulated Shapes," *Intl. Conf. on Scale Space and Variational Methods in Computer Vision*, 2011.
- [57] A. M. Bruckstein and D. Shaked, "Skew symmetry detection via invariant signatures," *Pattern Recognition*, vol. 31(2), pp. 181-192, Feb. 1998.
- [58] J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, "Discovering Texture Regularity as a Higher-Order Correspondence Problem", *ECCV*, 2006.
- [59] M. Park, K. Brocklehurst, R. Collins, and Y. Liu, "Deformed Lattice Detection in Real-World Images using Mean-Shift Belief Propagation", *TPAMI*, vol. 31(10), pp. 1804 – 1816, Oct. 2009.

- [60] Y. Ran, Q. Zheng, R. Chellappa, and T. M. Strat, "Applications of a Simple Characterization of Human Gait in Surveillance", *IEEE Trans. SMC-B*, vol. 40(4), pp. 1009-1020, Aug. 2010.
- [61] S. Lee, Y. Liu, and R. Collins, "Shape Variation-Based Frieze Pattern for Robust Gait Recognition", *CVPR*, 2007.
- [62] P. M. Birgani, M. Ashtiyani and S. Asadi, "MRI Segmentation Using Fuzzy C-means Clustering Algorithm Basis Neural Network," *Intl. Conf. on Information & Communication Technologies*, 2008.
- [63] Y. Kabir, M. Dojat, B. Scherrer, C. Garbay and F. Forbes, "Multimodal MRI segmentation of Ischemic Stroke Lesions," *IEEE Intl. Conf. on Engineering in Medicine and Biology Society*, 2007.
- [64] K. V. Leemput, F. Maes, D. Vandermeulen, A. Colchester and P. Suetens, "Automated Segmentation of Multiple Sclerosis Lesions by Model Outlier Detection," *IEEE Trans. on Medical Imaging*, Vol. 20, No. 8, pp. 677-688, August 2001.
- [65] M. B. Cuadra, C. Pollo, A. Bardera, O. Cuisenaire, J. Villemure and J. Thiran, "Atlas-Based Segmentation of Pathological MR Brain Images Using a Model of Lesion Growth," *IEEE Trans. on Medical Imaging*, Vol. 23, No. 10, pp. 1301-1314, October 2004.
- [66] G. Manana, E. Romero and F. Gonzalez, "A Grid Computing Approach to Subtraction Radiography," *Intl. Conf. on Image Processing*, 2006.
- [67] Z. Cao, X. Liu, B. Peng and Y. Moon, "DSA image Registration Based on Multiscale Gabor Filters and Mutual Information," *Intl. Conf. on Information Acquisition*, 2005.
- [68] F.P.G. Bergo, A.X. Falcão, C.L. Yasuda and F. Cendes, "FCD Segmentation using Texture Asymmetry of MR-T1 Images of the Brain," *IEEE International Symposium on Biomedical Imaging*, 2008.
- [69] N. Ray, B. N. Saha and M. Brown, "Locating Brain Tumor from MR Imagery Using Symmetry," *Asilomar Conf. on Signals, Systems, and Computers*, 2007.
- [70] H. Khotanlou, O. Colliot, J. Atif and I. Bloch, "3D Brain Tumor Segmentation in MRI Using Fuzzy Classification, Symmetry Analysis and Spatially Constrained Deformable Models," *Fuzzy Sets and Systems*, vol. 160, no. 10, pp. 1457-1473, May 2009.

- [71] S. Klein, M. Staring, K. Murphy, M. A. Viergever and J. Pluim, “elastix: A Toolbox for Intensity-Based Medical Image Registration,” *IEEE Trans. on Medical Imaging*, vol. 29, no. 1, pp. 196 - 205, 2010.
- [72] C. Tresp, M. Jager, M. Moser, J. Hiltner and M. Fathi, “A New Method for Image Segmentation Based on Fuzzy Knowledge,” *IEEE International Joint Symposia on Intelligence and Systems*, 1996.
- [73] Y. Sun and B. Bhanu, “Symmetry-integrated Region Based Image Segmentation,” *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2009.
- [74] B. Bhanu and O. D. Faugeras, “Segmentation of Images Having Unimodal Distributions,” *IEEE Trans. on PAMI*, Vol. 4, No. 4, pp. 408-419.
- [75] B. Zitova and J. Flusser, “Image Registration Methods: A Survey,” *Image and Vision Computing*, Vol. 21, no. 11, pp. 977-1000, Oct. 2003.
- [76] Q. Du and I. Kopriva, “Automated Target Detection and Discrimination Using Constrained Kurtosis Maximization,” *Geoscience and Remote Sensing Letters*, Vol. 5, No. 1, pp. 38-42, Jan. 2008.
- [77] J. A. Bilmes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixed and Hidden Markov Models,” *U.C. Berkeley Technical Report*, TR-97-02, April 1998.
- [78] B. Bhanu and B. Parvin, “Segmentation of Natural Scenes”, *Pattern Recognition*, Vol. 20, No. 5, pp. 487-496.
- [79] C. Sadasivan, L. Cesar, J. Seong, A. K. Wakhloo and B. B. Lieber, “Treatment of Rabbit Elastase-Induced Aneurysm Models by Flow Diverters: Development of Quantifiable Indexes of Device Performance Using Digital Subtraction Angiography,” *IEEE Trans. on Medical Imaging*, Vol. 28, no. 7, pp. 1117 – 1125, June 2009.
- [80] N. Otsu, “A Threshold Selection Method From Gray-Level Histograms,” *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 9, No. 1, pp. 62-66, Jan. 1979.
- [81] R. Ratan, S. Sharma, and S. K. Sharma, “Brain Tumor Detection based on Multi-parameter MRI Image Analysis”, *ICGST-GVIP Journal*, Vol. 9, No. 3, pp. 9-17, June 2009.
- [82] P. Anbeek, K. L. Vincken and M. A. Viergever, “Automated MS-Lesion

Segmentation by K-Nearest Neighbor Classification”, *MICCAI Workshop on MS Lesion Segmentation*, 2008.

- [83] S. A. Hojjatoleslami and F. Kruggel, “Segmentation of Large Brain Lesions”, *IEEE Trans. on Medical Imaging*, Vol. 20, No. 7, pp. 666-669, July 2001.
- [84] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha and A. Yuille, “Efficient Multilevel Brain Tumor Segmentation with Integrated Bayesian Model Classification”, *IEEE Trans. on Medical Imaging*, Vol. 27, No. 5, pp. 629-640, May 2008.
- [85] “Clinical Applications of SPECT/CT: New Hybrid Nuclear Medicine Imaging System”, *IAEA TECDOC Series No. 1597*, 2008.
- [86] A. Bianchi, Personal communication on the implementation of hierarchical algorithm.
- [87] I. D. Dinov, "Expectation Maximization and Mixture Modeling Tutorial", *California Digital Library, Statistics Online Computational Resource*, December 9, 2008.
- [88] H. G. Feichtinger and T. Strohmer, "Gabor Analysis and Algorithms", *Birkhäuser*, ISBN 0817639594, 1998.
- [89] J. P. Thirion, S. Prima, G. Subsol and N. Roberts, “Statistical Analysis of Normal and Abnormal Dissymmetry in Volumetric Medical Images”, *Medical Image Analysis*, No. 4, pp. 111-121, 2000.
- [90] C. Xu and J. L. Prince, “Snakes, Shapes and Gradient Vector Flow”, *IEEE Trans. on Image Processing*, Vol. 7, No. 3, pp. 359-369, March 1998.
- [91] X. Li, T. Qiu, S. Lebonvallet and S. Ruan, “An Automatic Method of Brain Tumor Segmentation From MRI Volume Based on the Symmetry of Brain and Level Set Method”, *Proc. SPIE*, Vol. 7546, No. 5, Feb. 2010.
- [92] V. Chen and S. Ruan., “Graph Cut Segmentation Technique for MRI Brain Tumor Extraction”, *Intl. Conf. on Image Processing Tools and Applications*, 2010.
- [93] V. P. Rathi, G. Pushpa and S. Palani, “Detection and Characterization of Brain Tumor using Segmentation Based on HSOM, Wavelet Packet Feature Spaces and ANN”, *Intl. Conf. on Electronics Computer Technology*, 2011.

- [94] A. Mayer, G. Zimmerman-Moreno, R. Shadmi, A. Batikoff and H. Greenspan, "A Supervised Framework for the Registration and Segmentation of White Matter Fiber Tracts", *IEEE. Trans on Medical Imaging*, Vol. 30, No. 1, pp. 131-145, Jan. 2011.
- [95] R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti, and J. E. Fritts, "Localized content-based image retrieval," *IEEE PAMI*, vol. 30, no. 11, pp. 1902–1912, Nov. 2008.
- [96] H. Liu, and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE PAMI*, vol. 17, no. 4, pp. 491-502, April 2005.
- [97] M. R. Azimi-Sadjadi, J. Salazar, and S. Srinivasan, "An adaptable image retrieval system with relevance feedback using kernel machines and selective sampling," *IEEE PAMI*, vol. 18, no. 7, pp. 1645-1659, July 2009.
- [98] M. R. Widyanto, and T. Maftukhah, "Fuzzy relevance feedback in image retrieval for color feature using query vector modification method," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.14, No.1, pp. 34-38, Jan. 2010.
- [99] B. Ko, J. Peng, and H. Byun, "Region-based image retrieval using probabilistic feature relevance learning," *Journal of Pattern Analysis & Applications*, Vol.4, No.2, pp. 174-184, June 2001.
- [100] B. Ko, J. Peng, and H. Byun, "Sequential forward selection approach to the non-unique oligonucleotide probe selection problem," *Proceedings of the Third IAPR International Conference on Pattern Recognition in Bioinformatics*, 2008.
- [101] D. Ververidis, and C. Kotropoulos, "Fast and accurate sequential floating forward feature selection with the bayes classifier applied to speech emotion recognition," *Elsevier Signal Processing*, Vol. 88, No. 12, 2008.
- [102] C. Tusk, K. Koperski, S. Aksoy, and G. Marchisio, "Automated feature selection through relevance feedback," *Geoscience and Remote Sensing Symposium*, 2003.
- [103] M. Toews, T. Arbel. Detection, Localization, and Sex Classification of Faces from Arbitrary Viewpoints and Under Occlusion. *PAMI*, vol. 31, no. 9, pp. 1567-1581, Sept. 2009.
- [104] L. I. Kuncheva, J. J. Rodriguez, C. Plumpton, D. E. J. Linden, S. J. Johnston. Random Subspace Ensembles for fMRI Classification. *TMI*, vol. 29, no. 2, pp.

531-542, Feb. 2010.

- [105] L. Jing, N. Allinson, T. Dacheng, L. Xuelong. Multitraining Support Vector Machine for Image Retrieval. *IEEE Trans. on Image Processing*, vol. 15, no. 11, Oct. 2006.
- [106] H. Wei, S. Billings. Feature Subset Selection and Ranking for Data Dimensionality Reduction. *PAMI*, vol. 29, no. 1, 2007.
- [107] H. Peng, F. Long, C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE TPAMI*, Vol. 27, No. 8, Aug 2005.
- [108] J. Brank, M. Grobelnik, N. Milic-Frayling, D. Mladenic. Feature Selection Using Linear Support Vector Machines. *Microsoft Research Technical Report*, June 2002.
- [109] P. Dollar, Z. Tu, H. Tao, S. Belongie. Feature Mining for Image Classification. *CVPR*, 2007.
- [110] L. C. Molina, L. Belanche. Feature selection algorithms: a survey and experimental evaluation. *ICDM*, 2002.
- [111] Authors.
- [112] Y. Rubner, C. Tomasi, L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *Intl. Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
- [113] J. Li, J. Z. Wang, G. Wiederhold. IRM: Integrated Region Matching for Image Retrieval. *ACM Multimedia*, 2000.
- [114] N. Abe, M. Kudo. Entropy Criterion for Classifier- Independent Feature Selection. *Intl. Conf. on Knowledge -Based and Intelligent Information & Engin. Systems*, 2005.
- [115] D. Kim, S. Yu. A new region filtering and region weighting approach to relevance feedback in content-based image retrieval. *Journal of Systems and Software*, vol. 81, no. 9, pp. 1525-1538, Sept. 2008.
- [116] D. Mohamad. Multi Local feature Selection Using Genetic Algorithm For Face Identification. *Intl. Journal of Image Processing*, vol. 1, no. 2, pp. 1-28, July 2007.

- [117] F. Li, Q. Dai, W. Xu. Improved Similarity-Based Online Feature Selection in Region-Based Image Retrieval. *Intl. Conf. On Multimedia and Expo*, 2006.
- [118] W. Jiang, G. Er, Q. Dai, L. Gu. Similarity-based Online Feature Selection in Content-based Image Retrieval. *IEEE Trans. on Image Processing*, vol. 15, no. 3, Mar. 2006.
- [119] M. Pham and T. Cham. Online Learning Asymmetric Boosted Classifiers for Object Detection. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2007.
- [120] C. Monteleoni and M. Kaariainen. Practical Online Active Learning for Classification. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2007.
- [121] E. Lughofer, J. E. Smith, M. A. Tahir, P. Caleb-Solly, C. Eitzinger, D. Sannen and M. Nuttin. Human–Machine Interaction Issues in Quality Control Based on Online Image Classification. *IEEE Trans. on Systems, Man and Cybernetics*, vol. 39, no. 5, pp. 960-971, Sept. 2009.
- [122] G. Qi, X. Hua, Y. Rui, J. Tang and H. Zhang. Two- Dimensional Multilabel Active Learning with an Efficient Online Adaptation Model for Image Classification . *IEEE PAMI*, vol. 31, no. 10, pp.1880-1897, Oct. 2009.
- [123] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, vol. 36, no. 1/2, pp. 105-139, July/Aug 1999.
- [124] L. E. A. Santana, D. F. Oliveira, A. M. P. Canuto and M. C. P. Souto. A Comparative Analysis of Feature Selection Methods for Ensembles with Different Combination methods. *Intl. Joint Conf. on Neural Networks*, 2007.
- [125] A. Tsymbal, S. Puuronen and D. Patterson. Feature Selection for Ensembles of Simple Bayesian Classifiers. *Intl. Symposium on Methodologies for Intelligent Systems*, 2002.
- [126] A. Tsymbal and S. Puuronen. Ensemble Feature Selection with the Simple Bayesian Classification in Medical Diagnostics. *IEEE Symposium on Computer-based Medical Systems*, 2002.
- [127] M. M. Masud, J. Gao, L. Khan, J. Han and B. Thuraisingham. A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data. *Int. Conf. on Data Mining*, 2008.

- [128] J. Cervantes, X. Li and W. Yu. Support Vector classification for large data sets by reducing training data with change of classes. *IEEE Intl. Conf. on Systems, Man and Cybernetics*, 2008.
- [129] C. J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, June 1998.
- [130] F. Dellaert. The Expectation Maximization Algorithm. *Technical Report number GIT-GVU-02-20*, Feb. 2002.
- [131] L. I. Kuncheva and Z. S. J. Hoare. Error-Dependency Relationships for the Naïve Bayes Classifier with Binary Features. *IEEE PAMI*, vol. 30, no. 4, pp. 735-740, April 2008.
- [132] D. Hernandez-Lobato, G. Martinez-Muo and A. Suarez. Statistical Instance-Based Pruning in Ensembles of Independent Classifiers. *IEEE PAMI*, vol. 31, no. 2, pp. 364-369, Feb. 2009.
- [133] L. Wang. Feature Selection with Kernel Class Separability. *IEEE PAMI*, vol. 30, no. 9, pp. 1534-1546, Sept. 2008.
- [134] F. Li, R. Fergus, and P. Perona. One-Shot Learning of Object Categories. *IEEE PAMI*, vol. 28, no. 4, pp. 594-611, April 2006.
- [135] J. Li and F. Li. What, where and who? Classifying event by scene and object recognition. *IEEE Intl. Conf. in Computer Vision*, 2007.
- [136] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. *Intl. Conf. on Computer Vision*, 2005.
- [137] L. Fei-Fei, R. Fergus, and P. Perona. Learning Generative Visual Models From Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *CVPR Workshop on Generative-Model Based Vision*, 2004.
- [138] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, vol. 36, no. 1-2, pp. 105-139, 1999.
- [139] M. Cheng, B. Fang, Y. Y. Tang, T. Zhang and J. Wen. Incremental Embedding and Learning in the Local Discriminant Subspace With Application to Face Recognition. *IEEE Trans. On Systems, Man, and Cybernetics*, vol. 40, no. 5, pp. 580-591, Sept. 2010.
- [140] Z. Chen, L. Huang and Y. L. Murphey. Incremental Learning for Text Document

Classification. Intl. Joint Conf. on Neural Networks, 2007.

- [141] S. Theodoridis and K. Koutroumbas. Pattern Recognition (4th Edition). ISBN: 978-1-59749-272-0. Academic Press.
-
- [142] B. Verma and A. Rahman. Cluster-Oriented Ensemble Classifier: Impact of Multicluster Characterization on Ensemble Classifier Learning. IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 4, pp. 605-618, April 2012.
- [143] F. Yang and K.Z. Mao. Robust Feature Selection for Microarray Data Based on Multicriterion Fusion. IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 8, no. 4, pp. 1080-1092, July-Aug. 2011.

Appendix

The Appendix consists of 5 figures. Figs. A1 and A2 show the pictorial results for images ‘Vase’, ‘Bear’, ‘Woman_2’, ‘Butterfly’ and ‘Fresco’. The quantitative results for these images are shown in Table 2.8. Fig. A3 shows the image segmentation results by the proposed method, on the image from the Caltech-101 database. Fig. A4 and Fig. A5 show images that are used to obtain the results in Table 2.10.

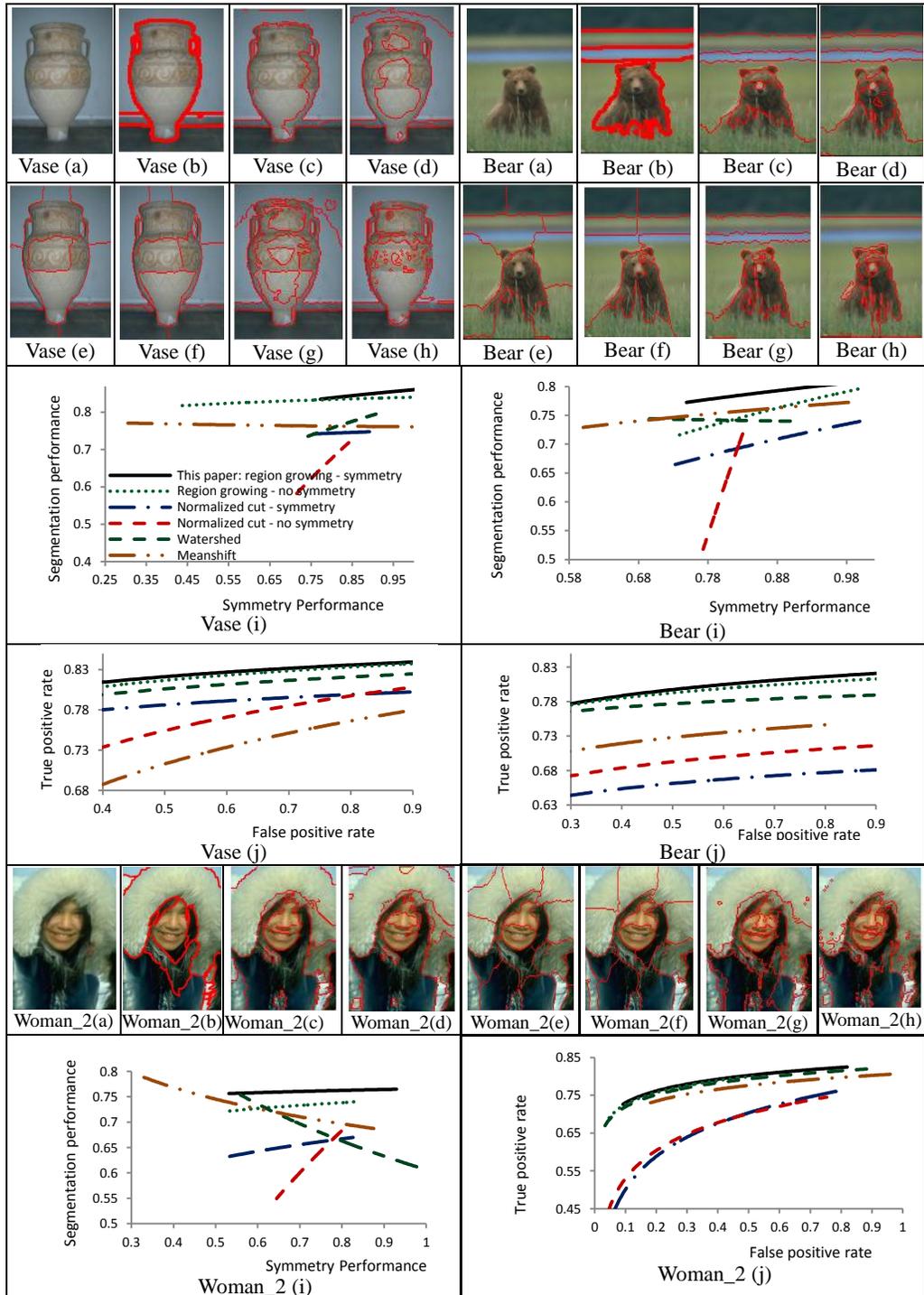


Fig. A1: Comparison of results on UCB database [43]: ‘Vase’, ‘Bear’ and ‘Woman_2’, (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing - no symmetry, (e) normalized cut - symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves.

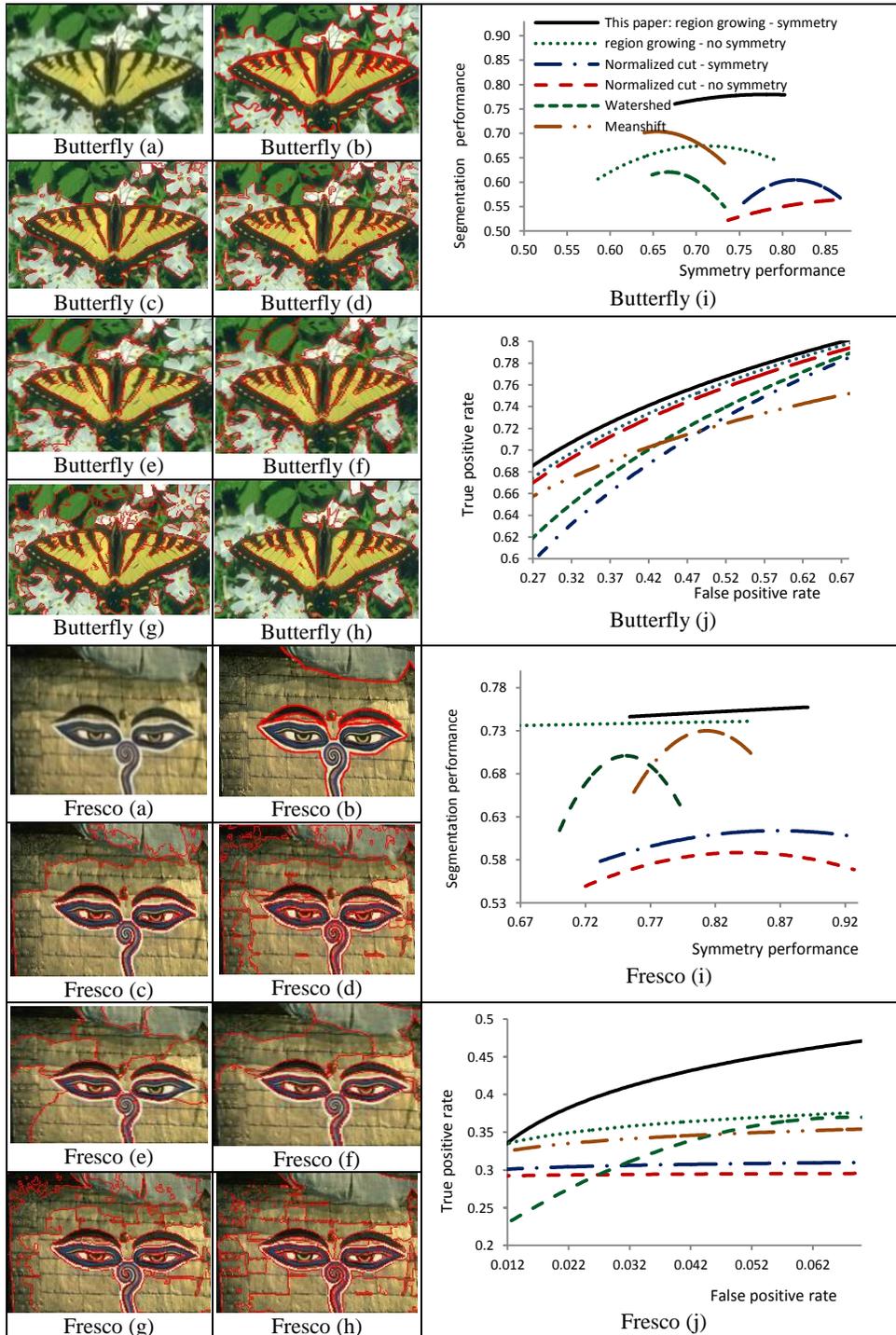


Fig. A2: Comparison of results on UCB database [43]: ‘Butterfly’ and ‘Fresco’. (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) normalized cut - symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves.

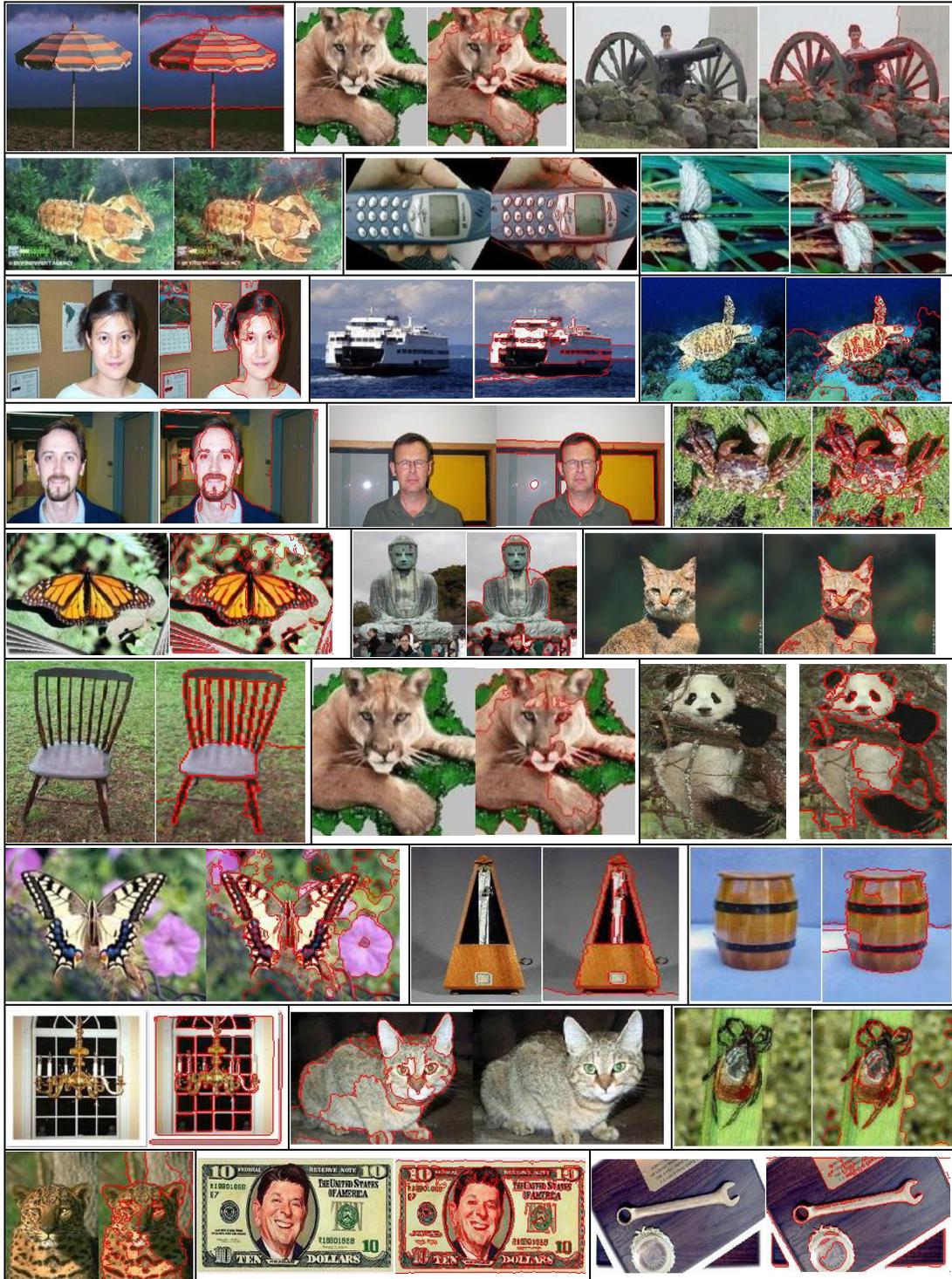


Fig. A3: Examples of symmetry-integrated segmentation results using images from the Caltech-101 database [42].

Category	Image	Category	Image	Category	Image
anchor	 24	butterfly	 10	chandelier	 66
ant	 23		 13	Cougar _face	 14
	 31		 14		 61
barrel	 2		 15		 62
binocular	 17		 24	crab	 3
brain	 65		 73		 35
buddha	 61		 86		 64
	 23	Ceiling _fan	 90		 73
	 31	cellphone	 4	crayfish	 38

Fig. A4. A list of 93 images selected from the Caltech-101 image database. The number shown below an image corresponds to the image numbers in a category in the Caltech-101 database. We provide these numbers so that our results can be replicated. These 93 images are used for the statistical validation of the proposed method, in Section 2.4.7 and Table 2.10.

Category	Image	Category	Image	Category	Image		
dalmatian	 51	Faces	 11	Faces _easy	 4		
	 55		 54		 45		
dragonfly	 28		 115		 54		
	 38		 150		 78		
Electric _guitar	 16		 159		 115		
lamp	 9		 397		 157		
	 26		 400		 164		
	 59		 435		 213		
lobster	 32		octopus		 34	platypus	 26

Fig A4 Continued.

Category	Image	Category	Image	Category	Image		
laptop	 55	menorah	 20	scissors	 4		
	 77		 27		 5		
Leopards	 44		 73		 20		
	 123		 74		 34		
mandolin	 36		pagoda		 4	tick	 4
	 37				 38		 18
	 42	panda	 5	 23			
 24	 27		 49				
metronome	 25	umbrella	 27	scorpion	 20		
	 26		 70	wild_cat	 24		

Fig A4 Continued.

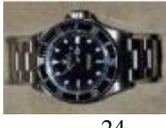
Category	Image	Category	Image	Category	Image
watch	 13	watch	 84	Windsor _chair	 21
	 24		 2		 30
	 27	 12	 50		

Fig A4 Continued.



Fig. A5: A list of 15 images selected from the UCB image database. These 15 images are used for the statistical validation of the proposed method, in Section 2.4.7 and Table 2.10.