

FRAMEWORK FOR CREATING LARGE-SCALE CONTENT-BASED IMAGE
RETRIEVAL SYSTEM (CBIR) FOR SOLAR DATA ANALYSIS

by

Juan Banda

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April, 2011

©COPYRIGHT

by

Juan Banda

2011

All Rights Reserved

APPROVAL

of a dissertation submitted by

Juan Banda

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citation, bibliographic style, and consistency and is ready for submission to The Graduate School.

Dr. Rafal Angryk

Approved for the Department of Computer Science

Dr. John Paxton

Approved for The Graduate School

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this dissertation in partial fulfillment of the requirements for a doctoral degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. I further agree that copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for extensive copying or reproduction of this dissertation should be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, Michigan 48106, to whom I have granted "the exclusive right to reproduce and distribute my dissertation in and from microform along with the non-exclusive right to reproduce and distribute my abstract in any format in whole or in part."

Juan Banda

April 2011

DEDICATION

This work is dedicated to my family, thank you for all the patience, support and encouragement over the many years my academic student career has lasted. A special mention goes to my mother, Dr. Ana Lilia Orozco, for the great example of motivation and determination. Without such an example, I would not have accomplished this goal.

ACKNOWLEDGEMENTS

My thanks and appreciation to Dr. Rafal Anrgyk for persevering with me as my advisor through out the time it took me to complete this research and write the dissertation. Without his invaluable and knowledgeable advice my research and this dissertation would not be what they are today. I would also like to thank the people that provided the valuable input in any way, shape, or form for this dissertation to be completed: Dr. Petrus Martens, Michael Schuh, Richard McAllister and Jamie Garrison.

Thanks to all the people that contributed my academic formation over the years: Dr. Alan Chmura, Dr. Tom Brown and Dr. Regina Aragon at Eastern New Mexico University and Dr. Martha Canales Leyva at Universidad Autonoma de Chihuahua. I would also like to include all my fellow classmates and office partners that helped me over the years. The institutions that I have attended: Montana State University, Eastern New Mexico University, and Universidad Autonoma de Chihuahua have been a great part of my academic formation and I will be eternally grateful to them.

Parisa Naghiz, Elrend Sandorf, Kathy Reiss, Stefan Klepp, Francisco Paez, Ece Topuzulu, Eric Johnston, Jill Foster and Eduardo Chairez are also acknowledged for their invaluable friendship, encouragement, and advise over the years. I would also like to express gratitude to the Office of International programs and all the fabulous people I have met through them over my years at Montana State. All the people that accompanied me over the years in my intramural teams, camping/hiking trips, road trips, and movie nights have a special mention in this section for making my time as a graduate student an extraordinary one. Nina (a.k.a Lisbeth Salander) for the daily company.

ACKNOWLEDGEMENTS – CONTINUED

Special thanks to the members of my Ph.D committee for their time and input on my dissertation work: Dr. Rafal Angryk, Dr. Petrus Martens, Dr. Brendan Mumey and Dr. Rocky Ross. The Department of Computer Science and its wonderful staff have my gratitude for the education and support provided.

Lastly, I would like to thank the people that greatly influenced my life during this academic journey, but are no longer with us: my grandparents and C.P. Francisco Velarde.

TABLE OF CONTENTS

1. INTRODUCTION	1
Document Organization	3
Framework Outline and Contributions	3
Contributions to Computer Science	10
Creation of a CBIR Building Framework	10
Creation of a Composite Multi-Dimensional Indexing Technique.....	11
Contributions to Solar Physics.....	12
Creation of a CBIR for Solar Dynamics Observatory	12
2. BACKGROUND INFORMATION	13
Content-Based Image Retrieval	13
Image Processing in Astrophysics	22
3. BENCHMARK DATASETS	25
Introduction.....	25
Solar Dataset	26
Dataset Creation.....	26
Dataset Class Balancing.....	27
Dataset Labeling	28
INDECS Dataset	30
ImageCLEFMED Datasets	30
PASCAL VOC Datasets	32
4. FEATURES FOR IMAGE RETRIEVAL	34
Introduction.....	34
Textural Image Parameters	35
Color Based Image Parameters.....	39
Shape/Region Based Image Parameters.....	41
Image Parameter Extraction.....	44
Grid-based Segmentation.....	44
Computational Costs	45
5. CLASSIFICATION ALGORITHMS FOR EVALUATION	48
Introduction	48
Naïve Bayes Classifier.....	49
C 4.5 Classifier.....	50
Support Vector Machines Classifier	51

TABLE OF CONTENTS – CONTINUED

Classification Measures Utilized for Evaluation	53
Initial Classification Evaluation Results	55
6. ATTRIBUTE EVALUATION	58
Introduction.....	58
Unsupervised Methods.....	59
Correlation Analysis	60
Supervised Methods.....	61
Chi Squared Method	61
Gain Ratio Method.....	61
Info Gain Method.....	61
Evaluation Results for the Solar Dataset	61
Unsupervised Attribute Evaluation.....	61
Supervised Attribute Evaluation	65
Classification Results for both Supervised and Unsupervised Attribute Evaluations	66
7. DISSIMILARITY EVALUATION MEASURES.....	81
Introduction.....	81
Euclidean Distance.....	82
Standardized Euclidean Distance.....	83
Mahalanobis Distance.....	83
City block Distance.....	83
Chebychev Distance.....	84
Cosine Distance	84
Correlation Distance	84
Spearman Distance.....	85
Hausdorff Distance	85
Jensen–Shannon Divergence	85
χ^2 Distance.....	86
Kullback–Leibler Divergence.....	87
Fractional Minkowski-based Metrics	87
Evaluation Results for the Solar Dataset	88
Dissimilarity Matrix Calculations.....	90
Transformation of Dissimilarity Matrices via Multidimensional Scaling	93
Fractional Minkowski Metrics.....	95
Component Thresholding.....	98
Quantitative Evaluation via Comparative Analysis of Classifiers.....	99
Quantitative Evaluation via Comparative Analysis of Classifiers for the Fractional Minkowski Metrics	104

TABLE OF CONTENTS – CONTINUED

8. DIMENSIONALITY REDUCTION	121
Introduction.....	121
Linear Dimensionality Reduction Methods	122
Principal Component Analysis	122
Singular Value Decompositions	123
Locality Preserving Projections	124
Factor Analysis	124
Non-linear Dimensionality Reduction Methods	125
Kernel PCA.....	125
Isomap.....	125
Laplacian Eigenmaps	125
Locally-Linear Embedding	126
Dimensionality Thresholding.....	128
Thresholding via PCA and SVD Variance Percentage.....	128
Mapping Functions	129
Evaluation Results	130
9. INDEXING AND RETRIEVAL	145
Introduction.....	145
Multi-Dimensional Indexing.....	146
R-Trees.....	147
R+-Trees	148
R*-Trees.....	148
kd-Trees	148
Single-Dimensional Indexing for Multi-Dimensional Data	151
iDistance	152
Pyramid-Tree	154
Measures for Evaluation of Retrieval	155
Evaluation Results	156
Multi-Dimensional Indexing: kd-tree Experiments	158
Single-Dimensional Indexing for Multi-Dimensional Data:	
Pyramid-tree Experiments	159
Single-Dimensional Indexing for Multi-Dimensional Data:	
iDistance Experiments	161
CBIR System - Retrieval Demonstration.....	169
Solar Images.....	169
ImageCLEFMed07 Images.....	177
PASCAL 2008 Images.....	183

TABLE OF CONTENTS – CONTINUED

10. GENERAL CONCLUSIONS.....	188
Closing Remarks.....	188
Ideas for Future Work.....	190
REFERENCES CITED.....	193
APPENDICES	211
APPENDIX A: Publications and Permissions.....	212
APPENDIX B: Source Code	214

LIST OF TABLES

Table	Page
3.1. Characteristics of the solar benchmark data set	26
3.2. Characteristics of the INDECS dataset generated from the INDECS database	29
3.3. Characteristics of the ImageCLEFMED05 dataset.....	31
3.4. Characteristics of the ImageCLEFMED07 dataset.....	31
3.5. Characteristics of the PASCAL2006 dataset	33
3.6. Characteristics of the PASCAL2006 dataset	33
4.1. List of extracted image parameters	47
5.1. Classifiers and their function calls used for this dissertation.....	55
5.2. Precision values for the classes of the solar dataset (closer to 1 is better)	56
5.3. Recall values for the classes of the solar dataset (closer to 1 is better)	56
5.4. F-Measure values for the classes of the solar dataset (closer to 1 is better)	57
6.1. Supervised attribute ranking results.....	66
6.2. Description of attribute evaluation Experiments	67
6.3. Classification results Obtained for our attribute evaluation experiments	67
6.4. Average precision values for our attribute evaluation experiments	68
6.5. Average recall values for our attribute evaluation experiments	69

LIST OF TABLES - CONTINUED

Table	Page
6.6. Average F-Measure values for our attribute evaluation experiments	69
7.1. Dissimilarity measure labels and ranges used in our classification accuracy figures.....	89
7.2. Scaled image plot ranges and colors corresponding to the classes of the solar dataset	90
7.3. Top 10 classification accuracy results for 10 component thresholded dimensionality reduction experiments. D indicates dissimilarity measure and P indicates image parameter number from table 4.1	107
7.4. Top 10 classification accuracy results for tangent thresholded dimensionality reduction experiments. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C= indicates number of components.....	107
7.5. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on ImageCLEFMED05 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components.....	110
7.6. Scaled image plot ranges and colors corresponding to the classes of the ImageCLEFMED05 dataset	110
7.7. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on ImageCLEFMED07 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components.....	112
7.8. Scaled image plot ranges and colors corresponding to the classes of the ImageCLEFMED07 dataset	112

LIST OF TABLES - CONTINUED

Table	Page
7.9. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on INDECS dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components.....	114
7.10. Scaled image plot ranges and colors corresponding to the classes of the INDECS dataset.....	114
7.11. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on PASCAL2006 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components.....	117
7.12. Scaled image plot ranges and colors corresponding to the classes of the PASCAL2006 dataset.....	117
7.13. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on PASCAL2008 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components.....	119
7.14. Scaled image plot ranges and colors corresponding to the classes of the PASCAL2008 dataset.....	119
8.1. Number of dimensions selected for each dataset.....	128
8.2. Top 10 classification results for our solar dataset for all three classifiers.....	140
8.3. Precision comparison between the SVM original and the PCA 6 experiments on the solar dataset	140
8.4. Recall comparison between the SVM original and the PCA 6 experiments on the solar dataset.....	141

LIST OF TABLES - CONTINUED

Table	Page
8.5. F-Measure comparison between the SVM original and the PCA 6 experiments on the solar dataset	141
8.6. Top 10 classification results for the ImageCLEFMED05 dataset for all 3 classifiers	143
8.7. Top 10 classification results for the ImageCLEFMED07 Dataset for all 3 classifiers	143
8.8. Top 10 classification results for the INDECS Dataset for all 3 classifiers	143
8.9. Top 10 classification results for the PASCAL2006 dataset for all 3 classifiers	144
8.10. Top 10 classification results for the PASCAL2008 dataset for all 3 classifiers	144
9.1. Top three average precision results for all datasets and dissimilarity measures	163
9.2. Experimental set-up for the different dissimilarity measures and iDistance based on the best dimensionality reduction experiments of chapter eight	164
9.3. Solar dataset classes and image ranges	170
9.4. ImageCLEFMed07 dataset classes and image ranges	178
9.5. PASCAL2008 dataset classes and image ranges	183

LIST OF FIGURES

Figure	Page
1.1. CBIR Building Framework Layout	8
1.2. Indexing Module Layout.....	9
1.3. Dissimilarity Measures Module	9
2.1. Traditional layout of a CBIR system	14
2.2. Example of Google Image Search ‘color matching’ feature	19
2.3. CBIR in Airlines.net [62] portal that allows users to find similar airplane images	20
2.4. CLUE (Cluster-based Image Retrieval) System [63] query example.....	20
2.5. Query retrieval examples in FIRE [64]	21
3.1. Sample Solar Dataset images.....	27
3.2. Sample INDECS Dataset images.....	29
3.3. Sample images from the ImageCLEFmed 2005 and 2007 datasets	31
3.4. Sample images from the PASCAL 2006 and 2008 datasets	32
4.1. Two Highly similar images according to the SIFT descriptors. (a) belongs to the Emerging Flux Class and (b) Filament class. As is clearly evident both images are in fact very different.	42
4.2. Regions of interest in this sample image from the (a) Active Region,. (b) Flare, and (c) Filament Activation classes	44
4.3. Example of our 128 by 128 pixel grid and image parameter extraction from one of the resulting cells.....	45

LIST OF FIGURES - CONTINUED

Figure	Page
4.4. Image Parameter Extraction Times for 1,600 Images	46
5.1. Examples of 3 SVM Hyperplanes in two dimensions. H3 doesn't separate the 2 classes (black and white dots). H1 does but with a small margin. H2 achieves separation with the maximum margin. [121]	52
5.2. Example of a confusion matrix	53
5.3. Optimal example of a ROC Curve evaluation. The closer the curve is to the Ideal Test, the better the classification	54
5.4. Initial classification accuracy for our six datasets, experiments using 10-fold cross-validation	55
5.5. ROC Curves for the 8 classes of the solar dataset, experiments using 10-fold cross-validation	57
6.1. Shows the one-to-one comparison between Query image (ql) cells against Images from the data set cells (Xn). 1,1 will only be compared to 1,1 and so on	60
6.2. Average correlation map for the Active Region class in the one image as a query against the same class scenario (intra-class correlation)	62
6.3. Average correlation map of the Active Region class in the one image as a query against other classes scenario (inter-class correlation)	63
6.4. Multi Dimensional Scaling 2D-plot of correlation map in Fig. 6.2	64
6.5. Multi Dimensional Scaling 2D-plot of correlation map in Fig. 6.3	65
6.6. ROC Curves for the best performing Solar experiment (Exp 3), experiments using 10-fold cross-validation	70

LIST OF FIGURES - CONTINUED

Figure	Page
6.7. Inter-class Correlation Maps for the ImageCLEFMED05 Dataset	71
6.8. Intra-class Correlation Maps for the ImageCLEFMED05 Dataset	72
6.9. Inter-class Correlation Maps for the ImageCLEFMED07 Dataset	73
6.10. Intra-class Correlation Maps for the ImageCLEFMED07 Dataset	74
6.11. Inter-class Correlation Maps for the INDECS Dataset	75
6.12. Intra-class Correlation Maps for the INDECS Dataset	76
6.13. Inter-class Correlation Maps for the PASCAL2006 Dataset	77
6.14. Intra-class Correlation Maps for the PASCAL2006 Dataset	78
6.15. Inter-class Correlation Maps for the PASCAL2008 Dataset	79
6.16. Intra-class Correlation Maps for the PASCAL2008 Dataset	81
7.1. Average time in minutes required to calculate one 1,600 by 1,600 dissimilarity matrix	91
7.2. Scaled Image plot of dissimilarity matrix for (a) Correlation (D7) measure with image parameter mean, (b) JSD (D10) measure with image parameter mean, (c) Chebychev (D5) measure with image parameter relative smoothness	92
7.3. MDS map for the correlation measure with image parameter mean (top 2 and top 3 components)	94

LIST OF FIGURES - CONTINUED

Figure	Page
7.4. Top 3 MDS components plot for (a) Correlation (D7) measure with image parameter mean, (b) JSD (D10) measure with image parameter mean, (c) Chebychev (D5) measure with image parameter relative smoothness	95
7.5. Plots for Fractional Minkowski metric with $p=0.25$ (D14) and the fractal dimension image parameter: (a) Scaled Image Plot of the dissimilarity matrix (b) Top 3 MDS components plot.....	96
7.6. Plots for Fractional Minkowski metric with $p=0.95$ (D18) and the fractal dimension image parameter: (a) Scaled Image Plot of the dissimilarity matrix (b) Top 3 MDS components plot.....	97
7.7. Exponential curve fitting for: (a) correlation measure (D7) with image parameter mean, (b) JSD measure (D10) with image parameter mean. The x-axis indicates the number of components (1-63) and the y-axis indicates the magnitude of these components	99
7.8. Percentage of correctly classified instances for the 10 component threshold: a) for the Naïve Bayes classifier, b) for the decision-tree (C4.5) classifier, and c) for the SVM classifier, for the original data (1-10) and our 180 experiments. Experiments using 10-fold cross-validation	100
7.9. Number of components to use indicated by the tangent thresholding method.....	101
7.10. Percentage of correctly classified instances for the tangent-based component threshold: a) for the Naïve Bayes classifier, b) for the decision-tree (C4.5) classifier, and c) for the SVM classifier, for the original data (1-10) and our 180 experiments. Experiments using 10-fold cross-validation	102

LIST OF FIGURES – CONTINUED

Figure	Page
7.11. Percentage of classification accuracy for the 10-component Minkowski based metrics with $p = .25$ to $p = 8$, ordered by image parameter and using the same ordering from table 7.1. Experiments using 10-fold cross-validation	104
7.12. Percentage of classification accuracy for the Tangent thresholded Minkowski based fractional metrics with $p = .25$ to $p = 8$, ordered by image parameter and using the same ordering from table 7.1. Experiments using 10-fold cross-validation	105
7.13. (a) 3D-MDS plot, (b) 2D-MDS plot, (c) Scaled Image Plot, for the ImageCLEFMED05 Dataset combination of Spearman (D8) measure with the mean (P2) image parameter	111
7.14. (a) 3D-MDS plot, (b) 2D-MDS plot, (c) Scaled Image Plot, for the ImageCLEFMED07 Dataset combination of City-block (D4) measure with the mean (P2) image parameter	113
7.15. (a) 3D-MDS plot, (b) 2D-MDS plot, (c) Scaled Image Plot, for the INDECS Dataset combination of KLD A-B (D12) measure with the fractal dimension (P8) image parameter	115
7.16. (a) 3D-MDS plot, (b) 2D-MDS plot, (c) Scaled Image Plot, for the PASCAL2006 Dataset combination of KLD A-B (D12) measure with the fractal dimension (P8) image parameter	118
7.17. (a) 3D-MDS plot, (b) 2D-MDS plot, (c) Scaled Image Plot, for the PASCAL2008 Dataset combination of KLD A-B (D12) measure with the entropy (P1) image parameter.....	120
8.1. Classification accuracy of All Dimensionality Reduction Methods for all 8 Experiments per Method for all Datasets. Experiments use 67% of the data for training and 33% for testing	131

LIST OF FIGURES - CONTINUED

Figure	Page
8.2. Classification accuracy for all Factor Analysis dimensions for the Solar dataset. Experiments use 67% of the data for training and 33% for testing	134
8.3. Classification accuracy for all KernelPCA dimensions for the Solar dataset. Experiments use 67% of the data for training and 33% for testing	135
8.4. Classification accuracy for all Laplacian dimensions for the Solar dataset. Experiments use 67% of the data for training and 33% for testing	136
8.5. Average Classification Accuracy per Number of Targeted Dimensions for the Solar Dataset Experiments use 67% of the data for training and 33% for testing	139
8.6. PCA 6 experiment SVM ROC Curves for the 8 classes of the solar dataset. Experiments use 67% of the data for training and 33% for testing	142
9.1. 2D R-tree example showing the MBR's and their tree organization, adapted from [162]. Dotted lines indicate nodes in the tree and solid lines represent the MBR's of actual data objects that have references to them placed in tree's leaves	147
9.3. B+ tree example, the tree links the keys 1-7 to data values d1-d7. The linked list (red) allows rapid in-order traversal. Image from [168]	150
9.4. Mapping of points via iDistance to a B+-tree. Figure from [52]	152
9.5. Sample breakdown of data points into pyramids. Figure adapted from [240]	153
9.6. 2D and 3D MDS plots that show the cluster formed by our image parameter, this indicates very close similarity between all images of our dataset	157

LIST OF FIGURES - CONTINUED

Figure	Page
9.7. kd-tree precision results for each of our 1,600 queries on our solar dataset. Index built on original 640-dimensional feature vector	158
9.8. Average retrieval precision values for the solar dataset and the pyramid-tree and iDistance indexing algorithms. Indexes built on dimensionally reduced feature vectors and the original feature vector	160
9.9. Average precision values for the dissimilarity measures implemented on the iDistance algorithm for the 640-dimensional original feature vectors	162
9.10. Average retrieval precision values for the experimental analysis of the dissimilarity measures outline in table 9.2.	165
9.11. Average 10-NN distance for all experiments and all datasets tested compared with the top original non-reduced datasets. Distance and datasets described in table 9.2.....	166
9.12. Average page reads for all experiments and all datasets tested compared with the original non-reduced datasets. Distance and datasets described in table 9.2	167
9.13. Average retrieval precision per number of generated dimensions for the solar dataset using the euclidean distance.....	168
9.14. Retrieval results for the solar dataset – Active Region class. The query image (#87) is in the top, and the retrieved 10-NN images are in the bottom rows	170
9.15. Retrieval results for the solar dataset – Coronal Jet class. The query image (#386) is in the top, and the retrieved 10-NN images are in the bottom rows	171

LIST OF FIGURES - CONTINUED

Figure	Page
9.16. Retrieval results for the solar dataset – Emerging Flux class. The query image (#440) is in the top, and the retrieved 10-NN images are in the bottom rows	172
9.17. Retrieval results for the solar dataset – Filament class. The query image (#741) is in the top, and the retrieved 10-NN images are in the bottom rows	173
9.18. Retrieval results for the solar dataset – Filament Activation class. The query image (#920) is in the top, and the retrieved 10-NN images are in the bottom rows	174
9.19. (a) Retrieval results for the solar dataset – Filament Eruption class. The query image (#1054) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the solar dataset – Flare class. The query image (#1323) is in the top, and the retrieved 10-NN images are in the bottom rows.....	175
9.20. Retrieval results for the solar dataset – Oscillation class. The query image (#1513) is in the top, and the retrieved 10-NN images are in the bottom rows.....	176
9.21. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Chest class. The query image (#48) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Hand class. The query image (#328) is in the top, and the retrieved 10-NN images are in the bottom rows	179

LIST OF FIGURES - CONTINUED

Figure	Page
9.22. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Cranium class. The query image (#552) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Side Chest class. The query image (#710) is in the top, and the retrieved 10-NN images are in the bottom rows	180
9.23. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Trachea class. The query image (#813) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Breast class. The query image (#1128) is in the top, and the retrieved 10-NN images are in the bottom rows	181
9.24. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Lungs class. The query image (#813) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Leg Bones class. The query image (#1128) is in the top, and the retrieved 10-NN images are in the bottom rows	182
9.25. (a) Retrieval results for the PASCAL2008 dataset – Car class. The query image (#8) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Bicycle class. The query image (#240) is in the top, and the retrieved 10-NN images are in the bottom rows	184
9.26. (a) Retrieval results for the PASCAL2008 dataset – Bottle class. The query image (#453) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Boat class. The query image (#737) is in the top, and the retrieved 10-NN images are in the bottom rows	185

LIST OF FIGURES - CONTINUED

Figure	Page
9.27. (a) Retrieval results for the PASCAL2008 dataset – Dog class. The query image (#986) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – TV Monitor class. The query image (#1193) is in the top, and the retrieved 10-NN images are in the bottom rows	186
9.28. (a) Retrieval results for the PASCAL2008 dataset – Person class. The query image (#1324) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Cat class. The query image (#1517) is in the top, and the retrieved 10-NN images are in the bottom rows	187

ABSTRACT

With the launch of NASA's Solar Dynamics Observatory mission, a whole new age of high-quality solar image analysis was started. With the generation of over 1.5 Terabytes of solar images, per day, that are ten times higher resolution than high-definition television, the task of analyzing them by scientists by hand is simply impossible. The storage of all these images becomes a second problem of importance due to the fact that there is only one full copy of this repository in the world, therefore an alternate and compressed representation of these images is of vital importance. Current automated image processing approaches in solar physics are entirely dedicated to analyze individual types of solar phenomena and do not allow researchers to conveniently query the whole Solar Dynamics Observatory repository for similar images of their interests.

We developed a Content-based Image Retrieval system that can automatically analyze and retrieve multiple different types of solar phenomena, this will fundamentally change the way researchers look for solar images in a similar way as Google changed the way people searched the internet. During the development of our system, we created a framework that would allow researchers to tweak and develop their own content-based image retrieval systems for different domain-specific applications with great ease and a deeper understanding of the representation of domain-specific image data. This framework incorporates many different aspects of image processing and information retrieval such as: image parameter extraction for reduced representation of solar images, image parameter evaluation for validation of image parameters used, evaluation of multiple dissimilarity measures for more accurate data analysis, analyses of dimensionality reduction methods to help reduce storage and processing costs, and indexing and retrieval algorithms for faster and more efficient search. The capabilities of this framework have never been available together as an open source and comprehensive software package.

With these unique capabilities, we achieved a higher level of knowledge of our solar data and validated each of our steps into the creation of our solar content-based image retrieval system with an exhaustive evaluation. The contributions of our framework will allow researchers to tweak and develop new content-based image retrieval systems for other domains (e.g astronomy, medical field) and will allow the migration of astrophysics research from the individual analysis of solar phenomenon into larger-scale data analyses.

CHAPTER 1

INTRODUCTION

With the recent launch of NASA's Solar Dynamics Observatory mission, we have been presented with the unique opportunity of having a massive repository of high-quality solar images to analyze and categorize. The mission's Atmospheric Imaging Assembly (AIA), generates a set of eight 4096 pixels x 4096 pixels images every 10 seconds. Which leads to a data transmission rate of approximately 700 Gigabytes per day (the entire mission is expected to be sending about 1.5 Terabytes of data per day), for a minimum of 5 years.

In this document, we propose the creation of a framework that will aid us in constructing a large-scale content-based image retrieval (CBIR) system for unlabeled solar phenomena, which is a novel idea in the field of solar physics - while still being an important computer science challenge [1, 2, 3, 4, 5]. Many solar physicists have been working on the recognition and categorization of individual types of phenomena with the use of custom-tailored algorithms for individual phenomena. Also, some domain-specific CBIR systems (e.g. medical and military images) have been implemented on a smaller scale, but have proven to be constrained by their specificity [6]. Since there are no 'out of the box' solutions that will work in this domain, the building of a CBIR system for large scale solar phenomena recognition and classification is a non-trivial task.

In order to create such a system, a new framework which provides a testing environment that would allow different image parameters and large scale image processing techniques to be fully tested for different domain-specific datasets is of vital

importance. The same principle can be applied to: 1) unsupervised and supervised attribute evaluation methods, 2) dimensionality reduction techniques, and 3) high-dimensional data indexing algorithms. This will allow our framework to be configurable enough to help in the construction of large-scale CBIR systems for different domains.

The novelty of this framework is the capability of combining different aspects of large-scale data processing in order to create a better suited retrieval mechanism for different domain-specific data sources. With such a framework in place, researchers will be able to try to combine different combinations of methods to determine the optimal configuration/combination of methods. This is due to the fact that many retrieval techniques, when dealing with high-dimensional spaces, tend to produce different results depending on the domain of the data they are being tested on (e.g. sparse data behaves differently than dense data). To our knowledge, there is nothing available that allows researchers to perform the combination and testing of such tasks presented in this image retrieval framework.

Such a framework will allow, and hopefully encourage, researchers to improve it in order to create an even more helpful tool that would allow the construction of CBIR systems in a faster and more efficient manner -- allowing them to focus on the domain-specific performance issues they encounter, rather than on the work needed to build a testing and deployment framework. A framework like this could also be very beneficial for researchers and students investigating the foundations for a CBIR system in both academia and industry, since the framework will allow them to quickly test

predetermined image parameters, image processing techniques, etc., in their own domain-specific dataset without the need to start from scratch.

Document Organization

This document is organized into the following sections:

- Background Information
- Benchmark Datasets
- Features for Image Retrieval
- Classification Algorithms for Evaluation
- Image Filtering for Solar Images
- Attribute Evaluation
- Dissimilarity Evaluation Measures
- Dimensionality Reduction
- Indexing Algorithms
- General Conclusions

Framework Outline and Contributions

In order to build our framework, presented in figure 1.1, we will start like other researchers [6, 7, 8] have done in the past, by carefully selecting the image parameters we wish to extract and feature in our framework. Since our domain-specific images are black and white and do not contain perfectly outlined shapes, based on [6, 9, 10] we decided to primarily implement texture-based image parameters. The reason is that these types of

parameters have been shown to be effective for similar images in the area of medical CBIR systems. In [6] the authors provide a very thorough and explicit investigation of many image parameters and provide charts that will aid future researchers in selecting which image parameters will work better for their particular images.

Immediately after commencing work on selecting the image parameters (*Feature Extraction Module* in figure 1.1) we noticed that, while there are some ‘out of the box’ freely available tools to determine the usefulness of image parameters (e.g. attribute evaluation in WEKA [11]), there is a need to be able to perform other types of attribute evaluation on the image parameters to better determine their usefulness and gain the trust of researchers from the solar physics community, where correlation-based tests are more popular than entropy-based evaluation. This is where we decided to start work on our framework; we wanted to be able to extract and test as many image parameters as we desired while being able to compare them in a consistent manner. In this step (*Attribute Evaluation Module* in figure 1.1), our framework is constructed to allow us to perform unsupervised attribute evaluation (i.e. correlation analysis), and it also allows us to output WEKA files, ready for the supervised attribute evaluation methods provided in WEKA. This is achieved in an automated way, after copying the image dataset in a folders and configuring a few initial parameters.

In order to enable testing, and the determination of the usefulness of the attribute evaluation methods, our framework allows users to automatically (or manually) specify which image parameters to remove in order to experiment on them. The framework will output WEKA files that will allow users to perform classification on them, for

comparative purposes only, and verify what kind of effect the selected parameters have on the results when compared against all parameters (*Comparative Evaluation* sections in figure 1.1, right-hand side). Here the users can run any classifier they want (available in WEKA or compatible software) and with any settings they desire. The classification task, in this stage, is merely for comparative analysis of the interaction between the image parameters in the classifiers, and to determine if they are worth keeping or can be safely ignored for future tasks.

Paying special attention to dissimilarity measures, since they can outline certain peculiarities of different datasets, we decided to incorporate 17 different measures (listed in figure 1.3) that are highlighted in the literature as the most relevant ones in different areas of image processing and retrieval. We cover from the traditional Minkowski based measures (i.e. Euclidean distance [12], Manhattan distance [12], etc) to more ‘modern’ and newly used measures to compare histograms (JSD [13], KLD [14], Hausdorff [15]). With the addition of these measures, we allow future users of our framework to test different measures that will allow them to gain insights on their domain-specific datasets when performing correlation analysis. These measures will also be of great use for our multi-dimensional indexing component of the framework, and will be discussed again later in this section.

One of the most important stages in building a CBIR system (and on our framework) is the dimensionality reduction stage (*Dimensionality Reduction Module* in figure 1.1). Since most image parameter vectors are very high in dimensionality, they will become very problematic when it is time to store, query, and retrieve them depending on

the volume of the dataset. We have incorporated in our framework eight different dimensionality reduction techniques that will allow the researcher to determine which of them produces the best (or closest to the best) results in their domain-specific dataset. Since most datasets are different, we incorporated different linear and non-linear dimensionality reduction techniques that take advantage of different properties of the dataset in order to produce an optimal low-dimensional representation of the original data. These techniques vary from the classical Principal Component Analysis [16] to more modern methods like Laplacian Eigen maps [17], and Locally-Linear Embedding [18], with the only common factor that all of these methods employed in our framework have a mapping function to map new unseen data into the reduced dimensional spaces. Having a mapping function is a fundamentally important issue since we want to be able to process new/unseen data from the user's queries.

In the last section of our framework, (*Indexing Module* in figure 1.1, outlined in detail in figure 1.2) we tackle the problem of indexing multi-dimensional data by providing the user with seven different indexing mechanisms to try on their datasets. With an innovative and configurable set-up, we will allow users to combine all the previously mentioned similarity measures and dimensionality reduction methods alongside several of the indexing mechanisms, in order to produce domain-specific indexing structures. There have been many differently named indexing mechanisms that allow users to use different similarity measures and dimensionality reduction techniques in order to achieve efficient retrieval, but only for a few domain specific problems [19, 20, 21]. Our framework has more of a 'plug and play' feel to it, since the users are not

restricted in any combination of measures (figure 1.3) or dimensionality reduction techniques that they want to investigate. The user can also add the ones he desires using our framework to combine and evaluate them.

As outlined in this chapter (and on figure 1.1), the proposed framework will allow users to efficiently test image parameters that will be later used on their domain-specific CBIR system. This framework provides a solid foundation of evaluation methods that are currently not available in any other freely available software. By combining Matlab scripts and toolboxes, with open source indexing algorithm implementation and the WEKA's API, we provide users the ability to quickly interconnect these software packages and focus more on the dataset-specific problems they encounter rather than the implementation side of things. Going a step further, for researchers with high-dimensional datasets we provide the capability to test several dimensionality reduction methods at once. Since retrieval is one of the most important parts for the success of CBIR systems, we also include a module (***Indexing Module*** in figure 1.1) that allows users to fine-tune widely accepted single-dimensional indexing algorithms for multi-dimensional data to better fit their needs by allowing them to change a wide variety of parameters. This module also contains testing facilities that will allow them to easily verify how their dataset is performing under the specified combination of indexing method and parameters, giving the users great flexibility when testing their newly created indexes.

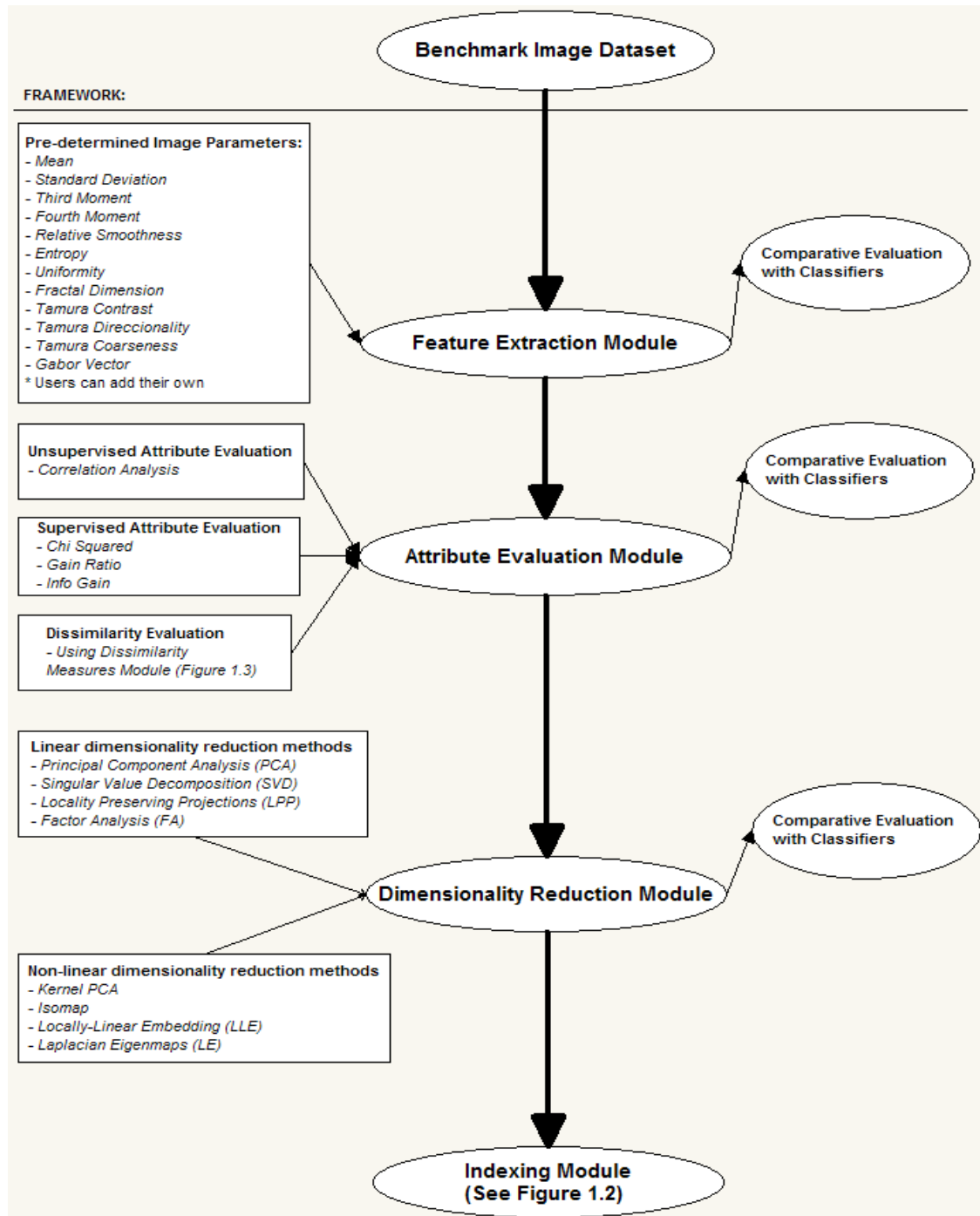


Figure 1.1. CBIR Building Framework Layout

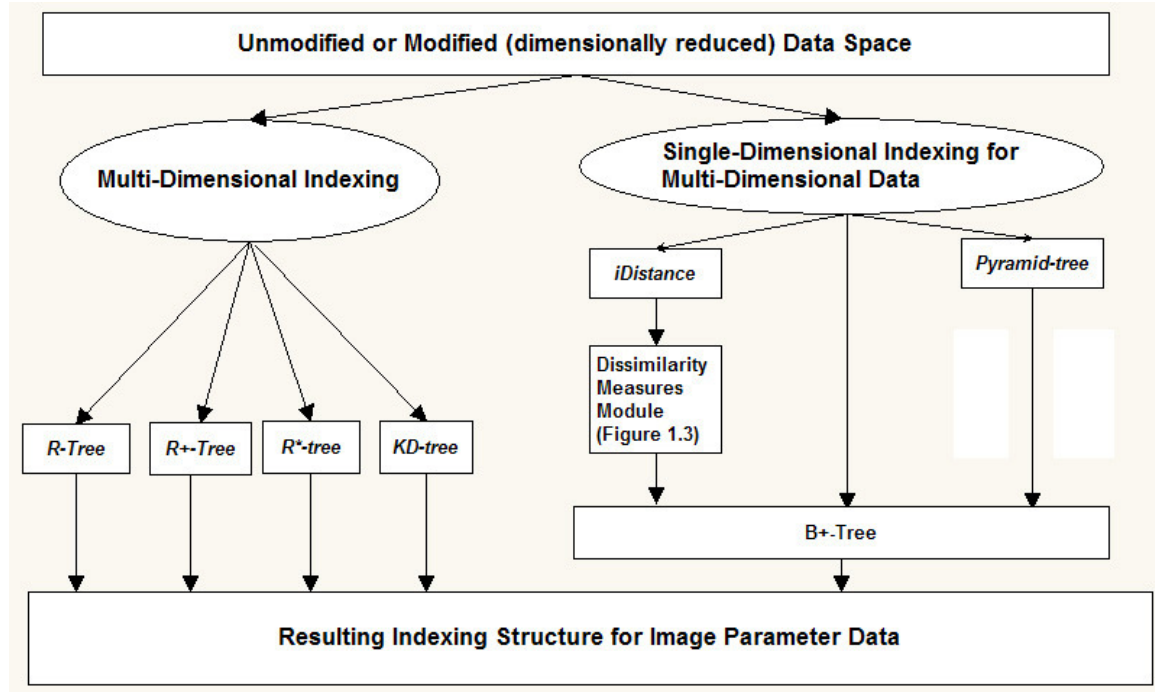


Figure 1.2. Indexing Module Layout

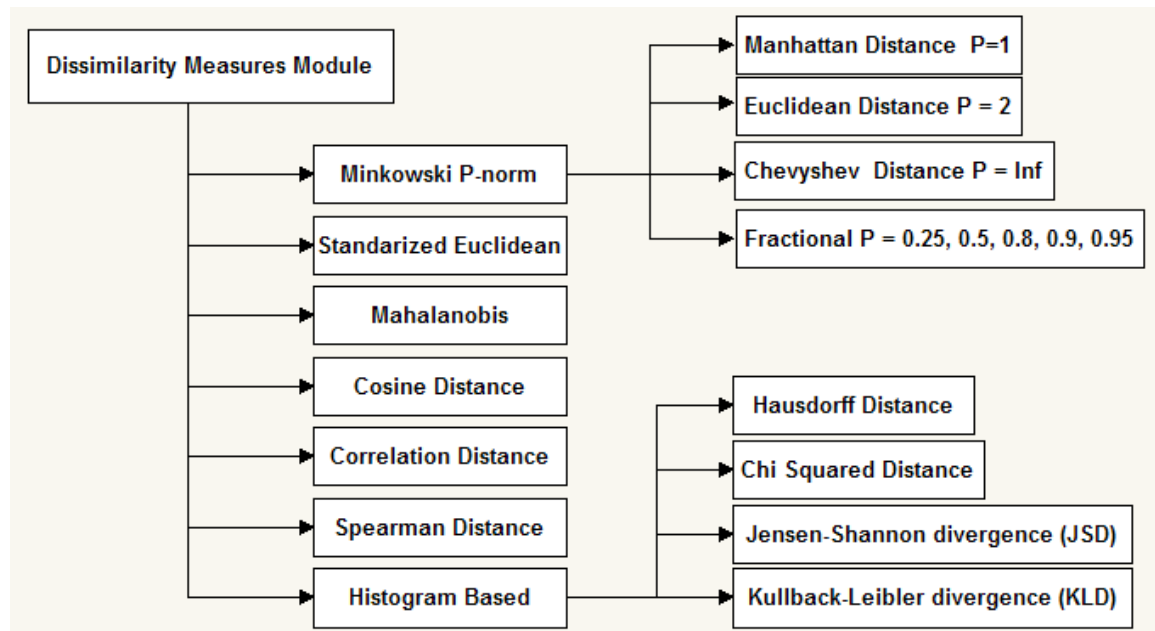


Figure 1.3. Dissimilarity Measures Module

Contributions to Computer Science

Creation of a CBIR Building Framework: With our novel application came the realization that while most stages and techniques needed to develop a CBIR are available via countless research papers, survey papers, algorithm implementations, and open source CBIR systems, there is the lack of a common framework that would allow a researcher to fully focus on the peculiarity of their domain-specific image dataset rather than building a testing environment, gathering code from multiple sources, and implementing everything to be able to test new image parameters on their specific image dataset. With such a big missing component, and based on our own experience, we theorize that many researchers in this area are spending most of their time building their testing framework and adding features to it, rather than testing the image parameters they desire and focusing on how to better describe their images in terms of building an image parameter vector.

We believe this is a considerable contribution since it will allow future researchers to spend their time gathering a deeper understanding of their data, this in turn will allow their CBIR system to be better optimized for their domain-specific task. With all of the components in this framework, our work in creating a CBIR system for the Solar Dynamics Observatory has been greatly aided by being able to test new image parameters, dimensionality reduction methods, etc. seamlessly and allowing us to fine tune our system for better performance.

Creation of a Composite Multi-Dimensional Indexing Technique: An important addition to our CBIR building framework is the ability to allow the user to use one of several indexing mechanisms for their image parameter vector. With high-dimensionality comes high complexity in terms of data retrieval and many of the current single-dimensional indexing for high-dimensional data methods rely on a combination of some kind of dimensionality reduction method and translating the new dimensional space into 1-D segments to be efficiently indexed by a B+-tree. In our framework, the user will be able to select one of our eight state-of-the-art dimensionality reduction methods and combine (and test) them with eight different high-dimensional data indexing mechanisms. This will allow users to test and select the most optimal retrieval mechanism for their needs. Since most retrieval mechanisms are highly domain-dependent, here users can test several indexing mechanism combinations and pick the best, or they can implement their own which can be added to this framework.

Contributions to Solar Physics

Creation of a CBIR for Solar Dynamics Observatory: The need to quickly identify phenomena in the growing number of solar images provided by the Solar Dynamics Observatory is of great importance since the timely hand-labeling is simply impossible with the volume of data generated. Individual algorithms work on identifying particular solar phenomena in SDO images, but they are not fully integrated to allow researches to query their entire repositories, let alone in a combination of them. In our novel approach, our goal is to have at least ten different solar phenomena classes in our CBIR system

(currently we have eight), allowing scientists to quickly retrieve images similar to the ones they find interesting, and search in a repository that contains more than one type of solar phenomenon.

CHAPTER 2

BACKGROUND INFORMATION

Content-based Image Retrieval

Content-Based Image Retrieval (CBIR) can be dated back to 1992 [22], and the term is used to describe the process of retrieving images from a specific collection based on descriptive image features or parameters. CBIR in broad terms is the application of computer vision techniques to the image retrieval problem (the problem of searching for digital images in large databases). The term “Content-based” implies that the search will analyze the contents of the images, rather than typical text metadata such as keywords, tags, and descriptions generally associated with the image. The image content is analyzed by using: descriptors, histograms, colors, shapes, textures, etc., that can be extracted or generated from the image. One of the main applications of CBIR spawned from the need of producing relevant results for image search engines, without relying on deficient metadata that produced a high number of irrelevant results. This can easily be addressed by humans generating the metadata, but this solution is infeasible with high volumes of image data generated every day; Flickr indicates in [23], their users upload a total of 4,320,000 new images per day. In our particular domain of application, the SDO mission will be generating approximately 69,000 images per day, making the task of hand labeling them impossible.

Some authors [1] in one of the main survey papers on the subject, believe that current CBIR systems are described as any piece of technology that helps to organize

digital image archives by their visual content. One of the most important aspects of CBIR is that we have researchers from different fields (computer vision, machine learning, information retrieval, human-computer interaction, database systems, Web and data mining, information theory, statistics, and psychology) contributing and becoming part of this growing community [24].

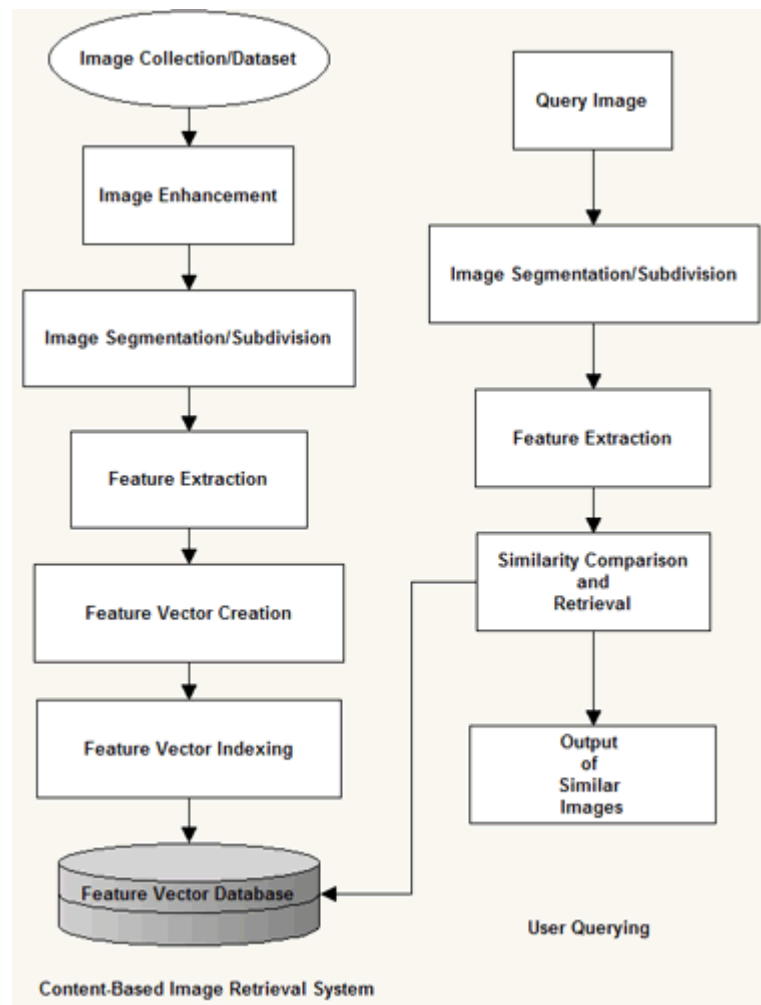


Figure 2.1. Traditional layout of a CBIR system

There are many very comprehensive surveys on CBIR [25, 26, 27, 28], but their main shortcoming is that they refer to advances and the state of the field pre-2000. Not

until [1] has there been an update on work done in CBIR with emphasis on image retrieval. For our domain specific problem of solar images, we referred to surveys of high-dimensional indexing of multimedia data [29] and applications of CBIR to medicine [30] since they seem to fit the constraints of our own CBIR system. In the next pages we will provide a brief discussion of the evolution of the field over time while indicating the essential needs for building a successful and efficient CBIR system.

In terms of image segmentation/subdivision (figure 2.1), we find that significant progress had been made in the field of image segmentation since the early beginnings, for example in [31] and [32], where snake- and region growing ideas were combined within a principled framework and spectral graph partitioning was employed for segments and shape matching needs. The use of graphs to represent spatial relationships between objects was geared towards medical imaging in [33]. A method for automatic feature selection was proposed in [34] and in [32] the topic of visual content description was nicely concluded showing the advantages and problems of image segmentation, along with other alternatives to it that still characterizes image structure well enough for retrieval. While there are many approaches for segmentation, in our particular domain-specific problem we will utilize a simpler and more effective (for our solar images) approach as describe in chapter 4. We point out the following very useful survey papers on the subject [35, 36, 37] for and in-depth look into this rich field.

The most important and crucial part of a CBIR system involves the selection of image parameters for feature extraction (figure 2.1). The most important thing here is to be able to determine what image parameters to extract in order to achieve meaningful and

relevant retrieval. The use of color histograms for image indexing was first introduced in [38], later it was popularized in CBIR systems such as QBIC [39], Pictoseek [40], and VisualSEEK [41]. Later improvements were made by taking specular reflection and shape into consideration in [42], and on [43] color correlograms were proposed as enhancements to histograms, taking into consideration the spatial distribution of the colors. In order to improve color layout, Daubechies' wavelet transforms were used in the WBIIS system [44]. Analysis of color image parameters is one of the main steps for researchers working on new CBIR systems, with emphasis on exploiting color spaces (YCbCr, LUV, etc) that seem to coincide better with regular human vision than the basic RGB color space -- even for print CMYK proves to be more effective. In recent years, research on color features has focused more on the construction of image signatures out of colors. A set of color and texture descriptors tested for inclusion in the MPEG-7 standard, and well suited to natural images and video, is described in [45] as well as chapter 4. Textural image parameters capture the granularity and repetitive patterns of surfaces within in a picture, and their role in domain-specific image retrieval, such as satellite imagery and medical imaging, is particularly interesting for our own solar image domain. These textural image parameters have been analyzed for many years in the fields of image processing and computer vision, and a thesaurus for texture-based image retrieval, has been proposed in [46] involving the application of banks of Gabor filters to the images, and calculating the image parameters from the filtered outputs. There are many textural image parameters but for the sake of our discussion, the ones we have pre-selected for our CBIR system are defined and described in chapter 4.

As a main difference between other works talking about the structures of CBIR systems, we have added the ‘Feature Vector Creation’ module in our figure 2.1. The reason we added this section was to introduce the analysis of said image parameters before building a full system. In this step, we propose the utilization of attribute evaluation methods in order to determine the usefulness of the selected image parameters. A more in-depth discussion of this comes in chapter 6. We also propose the analysis of said image parameters with not only the Euclidean distance, as other researchers have shown [6, 12, 7] there are other distance metrics that can find underlying relationships between our image parameters that are not seen by the traditional Euclidean distance. We have an ample discussion (and very interesting results) for this proposed dissimilarity evaluation in chapter 7.

Once our image parameters are extracted and evaluated, the very important question as to how they could be indexed and matched against each other for retrieval arises. The resulting feature vectors are of very high dimensional nature, and indexing of high-dimensional data is one of the biggest challenges of CBIR systems. Numerous approaches have been proposed: R-Trees [47], R*-Trees [48], KD-Trees [49], iDistance [50], Pyramid-Tree [240], to name a few. However, none of these methods seem to address or be presented using more than 30-40 dimensions [52, 30]. In LDR [53] we first see an analysis that presents results on real datasets and 120 dimensions. High-dimensional indexing remains one of the most researched areas in CBIR, and there are constantly new algorithms being developed. In some earlier approaches, particularly applied to CBIR systems, we have: [34], where similarity measures are grouped for

feature-based matching, structural feature matching in hierarchically ordered sets of image features [54], object-silhouette-based matching [55] and learning-based approaches for similarity matching [56, 57]. In our approach, we incorporate dissimilarity measures with iDistance, the most effective single-dimensional indexing algorithms for high-dimensional data we found through our literature review. As we will show in several chapters of this dissertation, the use of different similarity measures will help us find hidden relationships between the domain-dependent image parameters and find surprising relationships that are not easily discovered using the traditional Euclidean distance similarity measure.

Wide usage of CBIR systems is still not very common, and while there are a few very used systems such as Yahoo! Image Search, and Google Image Search, these systems rely heavily on metadata (such as file name, surrounding text, etc) and have very poor performance when it comes to complicated searches. In recent years these search engines have mimicked the incorporation of content-based features. For example, when we query for a certain type of car, lets say a Porsche, the search allows us to look for a specific color. However, this is still based on some of the metadata as you can find when hovering over each of the returned images (see figure 2.2).

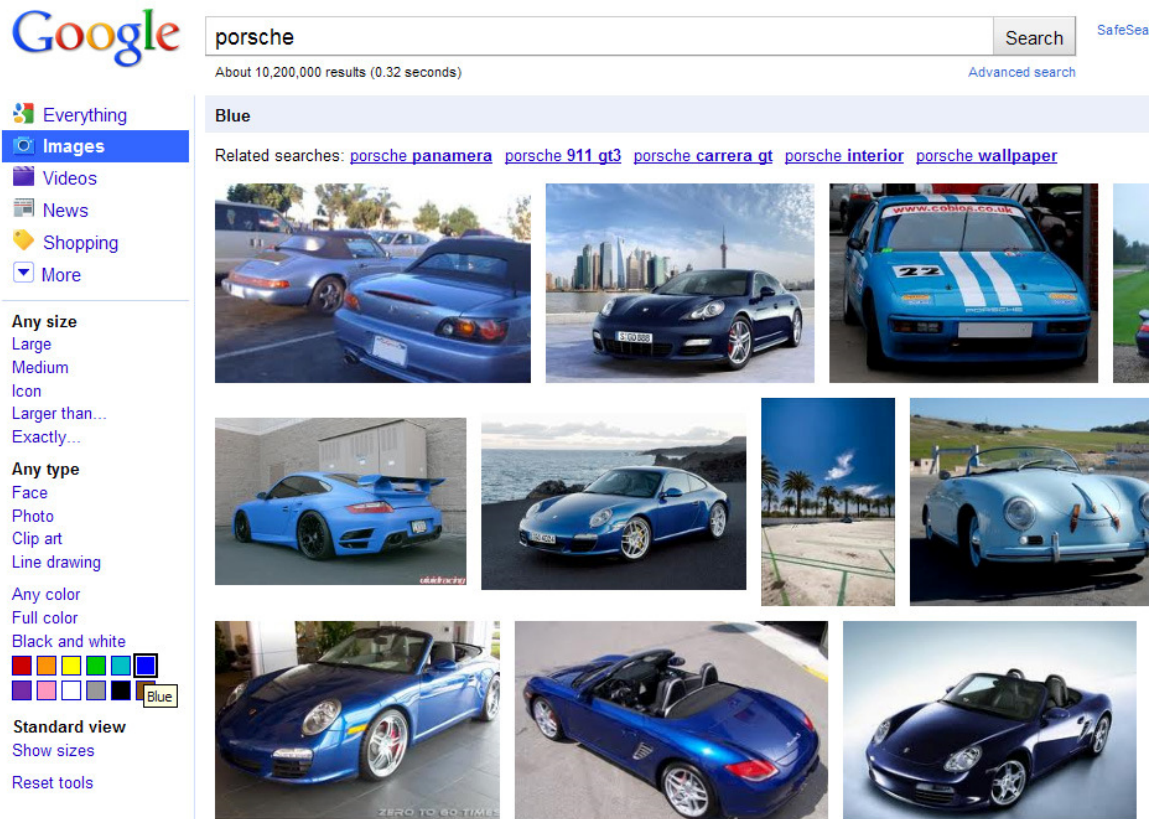


Figure 2.2. Example of Google Image Search 'color matching' feature

Another very visible application of CBIR systems is used on the website *Airliners.net*. In this vast source of information and pictures of airplanes, we have the similar search for pre-indexed images (as seen in Figure 2.3). The CBIR system behind the image retrieval of this site has been developed by Wang in [58], which has also spawned other systems such as the CLUE project (seen in figure 2.4). There are also more general uses for CBIR systems such as family album management, astronomy, and remote sensing; [59, 60, 61]. One of the main drawbacks of these systems is that they rely on retrieval of whole images when the actual object is represented in a very small section of the image. This will lead to very random results, as seen in (figure 2.4), where all the

images contain a dominant amount of blue color making them easily retrievable, but they do not seem to match our query image at all.

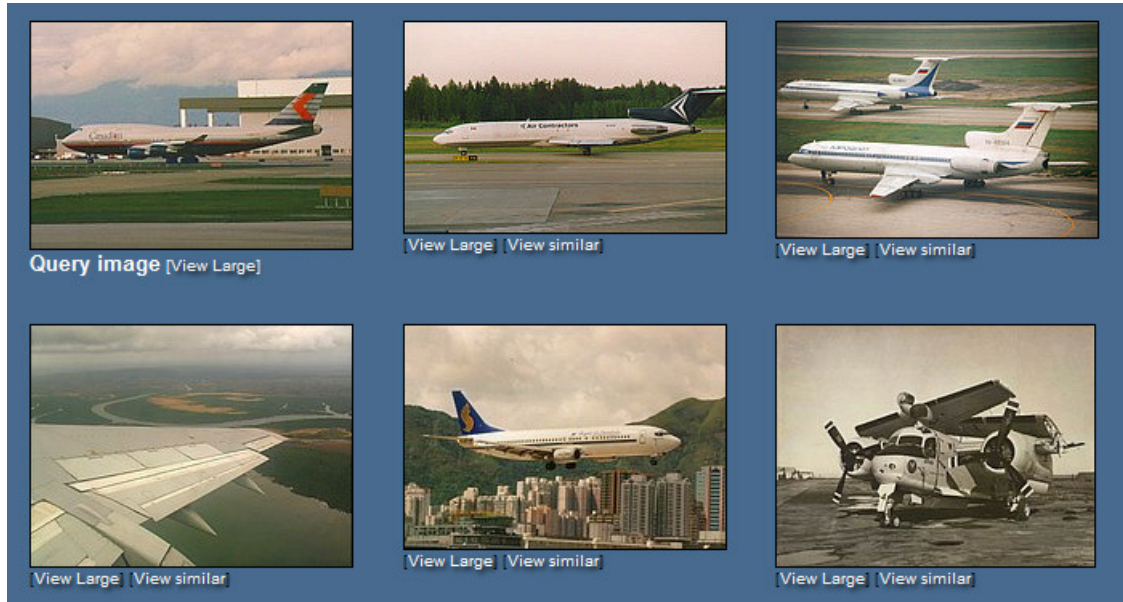


Figure 2.3. CBIR in Airlines.net [62] portal that allows users to find similar airplane images

Unified Feature Matching + Spectral Clustering

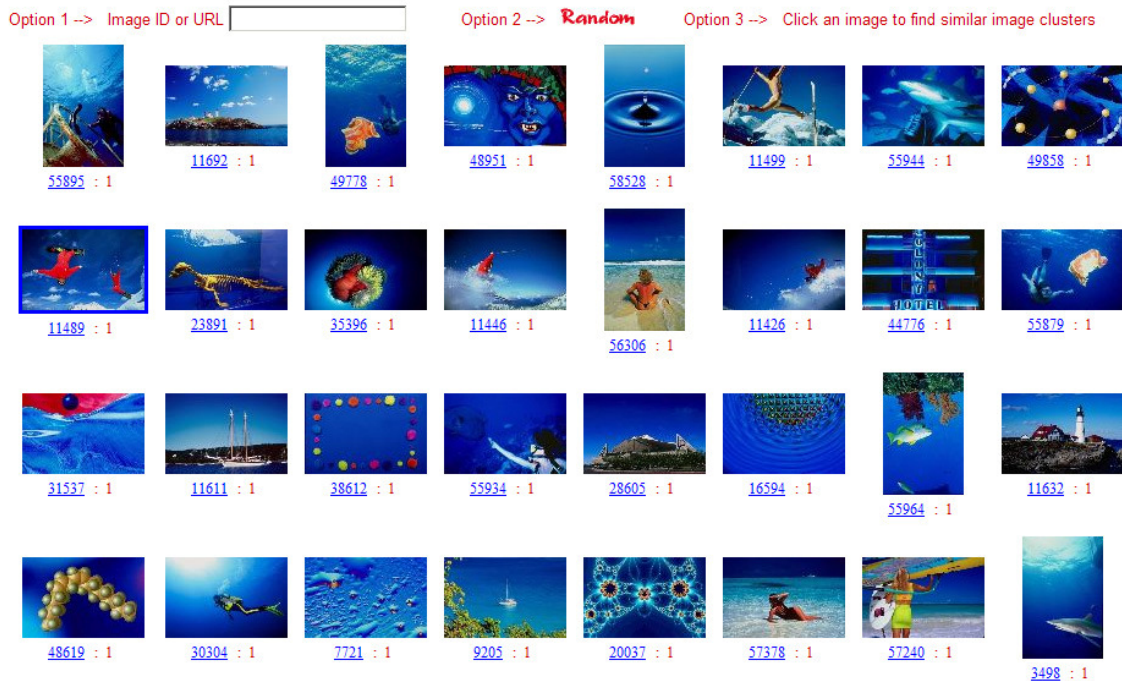


Figure 2.4 CLUE (Cluster-based Image Retrieval) System [63] query example

In terms of CBIR systems that have spawned from research projects, and are now freely available for other researchers to customize, we have the Flexible Image Retrieval Engine (FIRE) [64] created by Thomas Deselaers and a group of researchers at the Human Language Technology and Pattern Recognition Group of RWTH Aachen University. This system to our knowledge is one of the very few that is fully available to the public and relies on content-based features for image retrieval tasks. While the interface and relevance feedback mechanisms are already provided and implemented, this system is somewhat complicated to extend and implement new image parameters on it. A sample query retrieval results is provided on image figure 2.5.

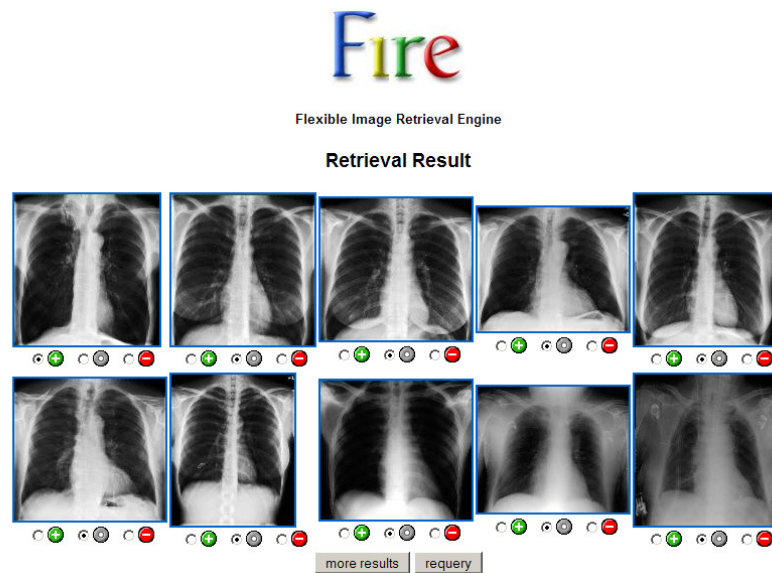


Figure 2.5 Query retrieval example in FIRE [64].

As we have learned during our literature review process of CBIR systems, very few researchers actually discuss their parameter selection in detail such as in [6]. And even in this discussion, there are several key types of dissimilarity evaluation that are not addressed in the extent as we present in the framework outlined in this document. There

is also almost no real guidance into how to identify valuable image attributes and observe the trade-offs from discarding them or combining them via different metrics. The proposed framework in this dissertation aims to cover those deficiencies and create a useful tool for researchers and educators embarking on the task of building a CBIR system for any type of domains.

Image Processing in Astrophysics

Since the first space-based observations of the sun in EUV and X-rays with Skylab the 1970's, NASA has been gathering massive amounts of solar images. With the launch of the Japanese/US/UK Yohkoh satellite in 1991, the influx of solar images has been uninterrupted. The ESA/NASA Solar and Heliospheric Observatory (SoHO) launched in 1996, and the Transition Region and Coronal Explorer (TRACE) in 1998, added more types of solar images: full-Sun Extreme Ultra-violet (EUV) images, visible wavelength images, and magnetograms. The most recent, NASA's Solar Dynamics Observatory launched on February 11 2010, is currently producing around 1.5 terabytes of data per day (it takes 4096 pixels x 4096 pixels images of the entire Sun, in 8 wavelengths every 10 seconds) leading to a total of over 69,000 images per day. With such advances in the acquisition of solar images, researchers have been faced with the problem of identifying solar phenomena in them, and with the current transmission rate, the hand labeling of them is virtually impossible.

In recent years, multiple attempts were made to automatically identify specific types of phenomena from solar images. Zharkova et al. [254] using Neural Networks,

Bayesian interference, and shape correlation, have detected phenomena including sunspots, flares and, coronal mass ejections. Zharkova and Schetinn [65] have trained a neural network to identify filaments within solar images. Wavelet analysis was used by Delouille [66] along with the CLARA clustering algorithm to segment mosaics of the sun. Irbah et al. [67] have also used wavelet transforms to remove image defects (parasite spots) and noise without reducing image resolution for feature extraction. Bojar and Nieniewski, [68] modeled the spectrum of the discrete Fourier transform of solar images and discussed various quality measures. However, we are not aware of any single technique reported effective in finding a variety of phenomena and no experiments have been performed on the size of repository even comparable to the dataset we will have.

Automatically detecting individual phenomena in solar images has become a popular topic in recent years. Zharkova et al. [254] discuss several methods for identifying features in solar images including Artificial Neural Networks, Bayesian interference, and shape correlation analyzing five different phenomena: sunspots, inference, plage, coronal mass ejections, and flares. Automatic identification of flares, on the SDO mission, will be performed by an algorithm originating from Christe et al. [69] which works well for noisy and background-affected light curves. This approach will allow detection of simultaneous flares in different active regions. Filament detection for the SDO mission will be provided by the “Advanced Automated Filament Detection and Characterization Code” [70]. The algorithm goes beyond the typical filament identification, in that it determines its spine and orientation angle, finds its magnetic chirality (sense of twist), and tracks it from image to image for as long as they are visible

on the solar disk. Additionally, if a filament is broken up into two or more pieces it can correctly identify them as a single entity. As for the coronal jet detection and parameter determination algorithms, they will work on data cubes covering a box enclosing the corresponding bright point and extending forward in time. SDO methods for determining the coronal jet parameters are described in detail in [71]. Oscillations on the SDO pipeline are detected using algorithms presented on [72] and [73] that consist of Wavelet transform analysis. In order to detect active regions the SDO pipeline use the Spatial Possibilistic Clustering Algorithm (SPoCA) that was developed by Barra [74], and produces a segmentation of EUV solar images into classes corresponding to active region, coronal hole and quiet sun.

As we can clearly see, the majority of currently popular approaches deal with the recognition of individual phenomena and a few of them have demanding computational costs, and it was not until recently that Lamb [9] discussed creating an example based Image Retrieval System for the TRACE repository. This is the only attempt we are aware of that involves trying to find a variety of phenomena, with expectation of building a large-scale CBIR system for solar physicists.

CHAPTER 3

BENCHMARK DATASETS

Introduction

CBIR systems have been shown to be very domain specific and require a benchmark dataset in order to be built. This was a particular problem for our application (solar images) as we found that since there was no widely available multi-class dataset, we had to create our own. Introduced in [75] our dataset covers several different solar phenomena in equally balanced representations. A detailed description of this dataset is found in section 3.2. For comparison purposes of our framework with other datasets we also analyzed five other domain specific (and very different) datasets. We utilized two medical datasets from the ImageCLEFMed image annotation and classification competition [76], two general purpose datasets from the PASCAL Visual Object Classes (VOC) challenge [77], and the INDECS Database [78]. Each of these datasets provides different domain images that will allow us to validate our domain specific experiments with our own solar dataset and will allow us to discover the usefulness of the framework presented in this work in the process of building CBIR systems for other domains. Selecting very different types of datasets will clearly allow us to show the domain-dependency of image parameters and the need to perform each section of this framework on the intended dataset in order to gain insight valuable insights about it and aid in the construction of a CBIR system for that particular domain. The six datasets will be explained in detail in individual sections in this chapter.

Solar Dataset

Dataset Creation

The Solar dataset was created using the Heliophysics Events Knowledgebase (HEK) portal [79] to find the event dates. Then we retrieved the actual images using the TRACE Data Analysis Center’s catalog search [80]. The search extends into the TRACE repository as well as other repositories. To make all images consistent, we filtered out all results that did not come from the TRACE mission. Table 3.1 provides an overview of our dataset. All images selected have an extent label of “Partial Sun”. All events retrieved were queried within the 99-11-03 to 08-11-04 date ranges. As one can see from the table 3.1, for the 16 events searchable by HEK (at the time of creation in 2008) we had to reduce the number of events to 8 due to the limited number of available occurrences of the remaining types of phenomena and poor quality of images available in some of the data catalogs (more on this in the next section). All images are sized 1024 x 1024 pixels.

Table 3.1. Characteristics of the solar benchmark dataset

Class #	Event Name	# of images retrieved	Wavelength
1	Active Region	200	1600
2	Coronal Jet	200	171
3	Emerging Flux	200	1600
4	Filament	200	171
5	Filament Activation	200	171
6	Filament Eruption	200	171
7	Flare	200	171
8	Oscillation	200	171

The full solar benchmark dataset contains 1,600 images distributed in 200 images per event class. The benchmark dataset both in its original and pre-processed format is available to the public via Montana State University’s Server [81].

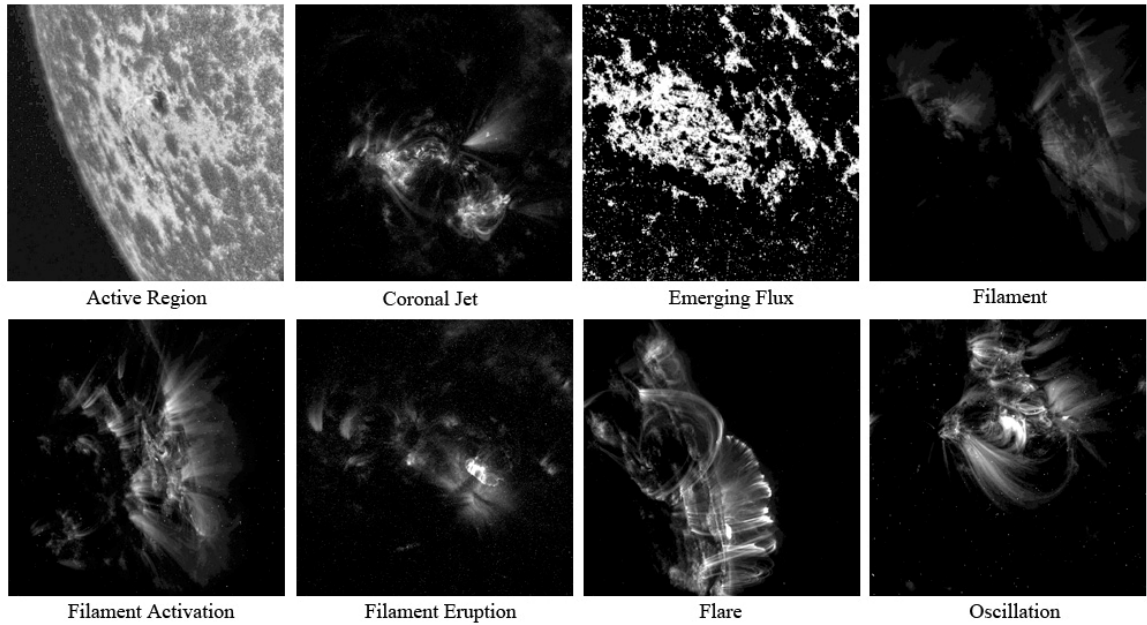


Figure 3.1. Sample Solar Dataset images

Dataset Class Balancing

In the process of creating our dataset to analyze parameters for image recognition we stumbled upon several problems when trying to balance the number of images per class. First, we found that several phenomena do not occur as often as others, making it harder to balance the number of images between individual classes. The second issue is that several phenomena can be sighted and reported in different wavelengths. For our benchmark we selected the wavelengths that contained the largest number of hand labeled results provided by HEK contributors. The current selection of classes was solely based on the number of images available, and the wavelengths in which the images were available. We wanted to be sure that our benchmark dataset has equally frequent classes in order to be sure that our results are not skewed towards the most frequent phenomena. For the events that occurred only a few times, but during a prolonged period of time, we

have selected a few different images within that time range to complete our goal of 200 images per event class.

Dataset Labeling

All reported events in the HEK repository have a label for the location of the phenomena in the HEK VOEvent XML [79] data, allowing the researcher to theoretically locate the exact position where the phenomena took place in the sun and the image. However, we encountered that this labeling is very inconsistent over the course of the phenomena duration. Due to this in our initial works [82, 83, 750] our classification accuracy was lower (73% as the maximum). We are currently working with human experts in labeling each image one by one to provide more accurate labels. As this process is not completed, we will utilize one label per image rather than labeling only the windows where the phenomena occurs for the experiments presented in this dissertation.

INDECS Dataset [78]

INDECS is an acronym which stands for INDoor Environment under Changing conditionS [78]; this database consists of several sets of pictures taken in five rooms of different functionality under various illumination and weather conditions at different periods of time. Each room was observed from many viewpoints and angles. Moreover, the normal activity in the rooms was recorded: people appear in the rooms, pieces of furniture are moved over time. The dataset can be obtained freely on the web at [84].

Table 3.2. Characteristics of the INDECS dataset generated from the INDECS database

Class #	Class Label	Number of images	Resolution
1	Corridor – Cloudy	200	1024x1024
2	Corridor – Night	200	1024x1024
3	Kitchen – Cloudy	200	1024x1024
4	Kitchen – Night	200	1024x1024
5	Kitchen – Sunny	200	1024x1024
6	Two-persons Office - Cloudy	200	1024x1024
7	Two-persons Office - Night	200	1024x1024
8	Two-persons Office - Sunny	200	1024x1024

After selecting a subset of this dataset (see table 3.2), we resize all the images to 1024x1024 pixels, and balance the classes in a way that we can provide a comparison with the solar and other datasets. This database features very different images than the ones found in the solar dataset and allow us to clearly demonstrate that CBIR research is very domain specific and is greatly dependent in the image parameters that the user selects in the early stages.

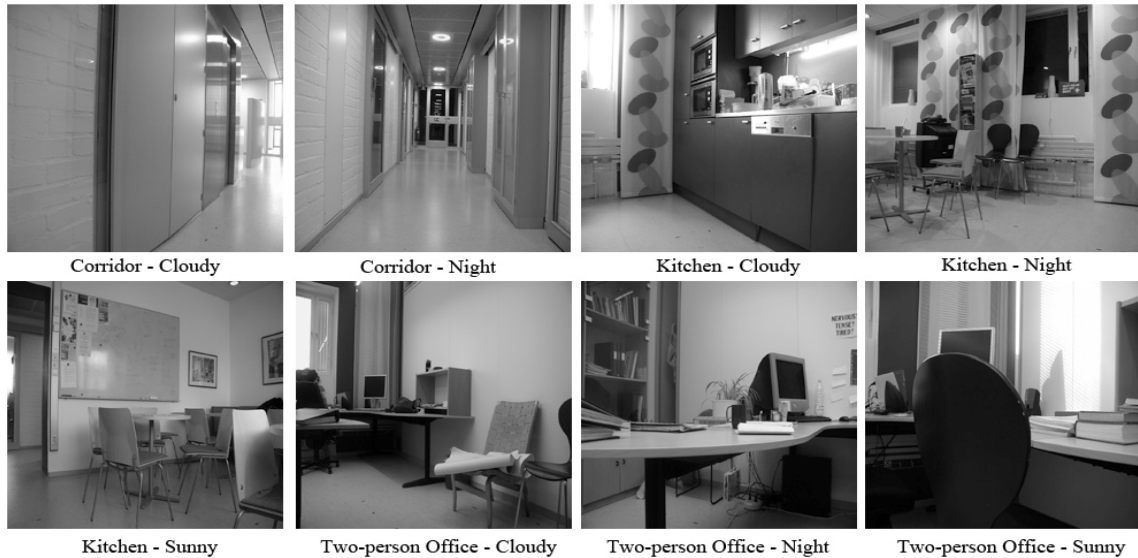


Figure 3.2. Sample INDECS Dataset images

ImageCLEFMed Datasets [76]

ImageCLEF is the cross-language image retrieval track which is run as part of the Cross Language Evaluation Forum (CLEF) [85]. ImageCLEF has participation from both academic and commercial research groups worldwide from communities including: Cross-Language Information Retrieval (CLIR), Content-Based Image Retrieval (CBIR) and user interaction. In their medical retrieval task, we find several datasets available: 2005, 2006, 2007, and 2010 (that features an augmented version of the 2008 and 2009 datasets). Currently, we are using the 2005 and 2007 datasets. The 2005 dataset consists of 10,000 radio graphs that can be fitted in a 512 x 512 pixels bounding box. From these 10,000 images, we have that 9,000 of them are categorized in 57 different categories; the remaining 1000 images are to be used for testing since they are uncategorized. For the 2006 and 2007 datasets the number of categories doubled to 116 and the number of images increased by one thousand each year.

This dataset compiles anonymous radiographs, which have been arbitrarily selected from routine at the Department of Diagnostic Radiology, Aachen University of Technology (RWTH), Aachen, Germany. The images in this dataset present different ages, genders, view positions and pathologies. This makes some of them very different from each other, but in some cases very similar to some of our Solar dataset. We randomly selected 8 different classes from the 2005 and 2007 datasets creating two balanced datasets with 1,600 images each. In tables 3.3 and 3.4 we describe the class distribution of our 2005 and 2007 dataset in detail. The resolution is variable but at least one side is 512 pixels.

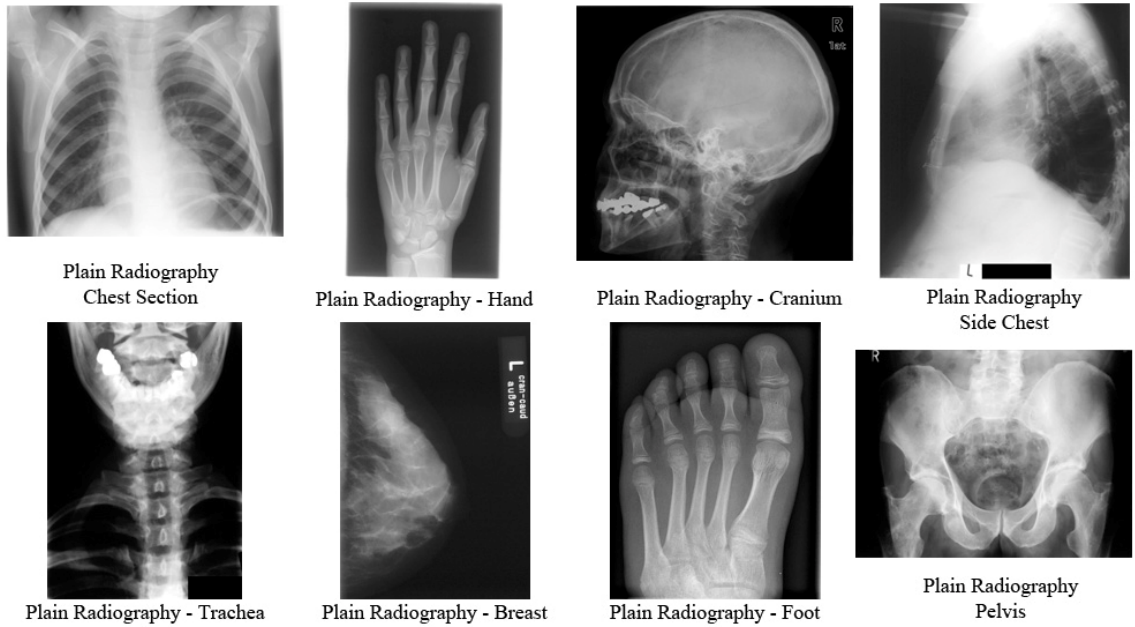


Figure 3.3. Sample images from the ImageCLEFMed 2005 and 2007 datasets

Table 3.3. Characteristics of the ImageCLEFMed05 dataset

Class #	Class Label	Number of Images	Resolution
1	Plain Radiography - Frontal Cranium	200	* Variable
2	Plain Radiography – Trachea	200	* Variable
3	Plain Radiography – Spine	200	* Variable
4	Plain Radiography – Hand	200	* Variable
5	Plain Radiography – Chest	200	* Variable
6	Plain Radiography – Pelvis	200	* Variable
7	Plain Radiography – Feet	200	* Variable
8	Plain Radiography – Cranium	200	* Variable

Table 3.4. Characteristics of the ImageCLEFMed07 dataset

Class #	Class Label	Number of Images	Resolution
1	Plain Radiography - Chest	200	* Variable
2	Plain Radiography - Hand	200	* Variable
3	Plain Radiography - Cranium	200	* Variable
4	Plain Radiography - Side Chest	200	* Variable
5	Plain Radiography - Trachea	200	* Variable
6	Plain Radiography - Breast	200	* Variable
7	Plain Radiography - Lungs	200	* Variable
8	Plain Radiography - Leg Bones	200	* Variable

PASCAL VOC Datasets [77]

The PASCAL Visual Object Classes (VOC) challenge is a popular benchmark in visual object category recognition and detection, providing the vision and machine learning communities with a standard dataset of images, annotation, and standard evaluation procedures. This dataset has been gaining popularity in the community since its conception in 2005. The main challenges of the PASCAL challenge are classification and detection of several different classes of objects within their images, we selected this dataset for their classification challenge and to observe how our domain-specific image parameters behave.

The PASCAL 2006 dataset consists of 10 classes, from which we selected 8 classes at random for our comparative analysis. For the PASCAL 2008 dataset of 20 classes, we again selected 8 different classes. For both datasets we selected 200 random images from each class.

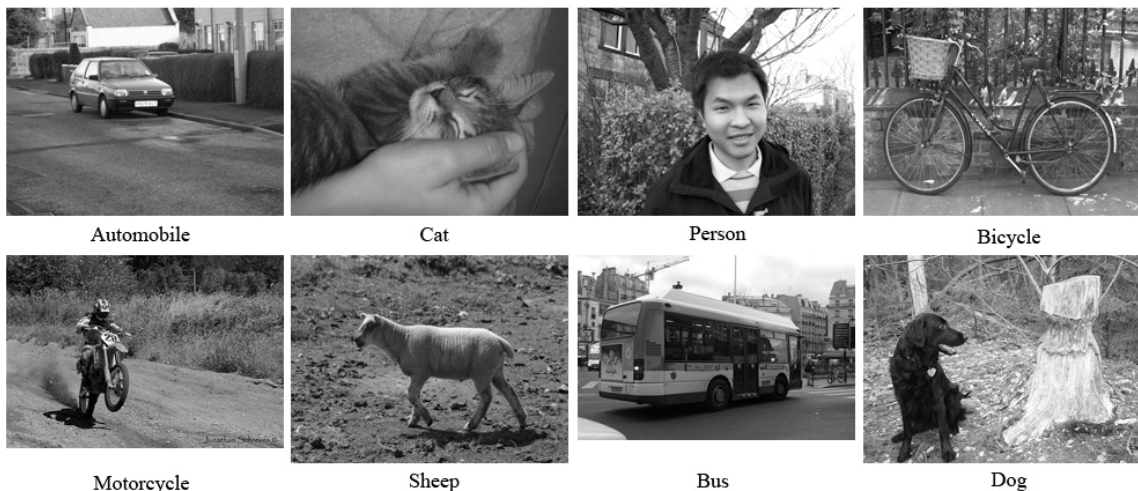


Figure 3.4. Sample images from the PASCAL 2006 and 2008 datasets

The images presented in this dataset provide a very tough challenge for any researchers analyzing them due to the fact that they are very dissimilar and they provide multiple labels per image in some instances. Researchers have shown good classification results when analyzing certain classes individually [172, 173, 174], but once they start considering more than 4 classes at the same time [175, 176, 177] classification accuracy tends to drop to less than 70%, showing the difficulties of selecting image parameters that work well for all types of images in this dataset. Descriptions of the two subsets of these datasets we created are presented in tables 3.5 and 3.6.

Table 3.5. Characteristics of the PASCAL2006 dataset

Class #	Class Label	Number of Images	Resolution
1	Car	200	* Variable
2	Bicycle	200	* Variable
3	Cat	200	* Variable
4	Cow	200	* Variable
5	Dog	200	* Variable
6	Horse	200	* Variable
7	Person	200	* Variable
8	Sheep	200	* Variable

Table 3.6. Characteristics of the PASCAL2008 dataset

Class #	Class Label	Number of Images	Resolution
1	Car	200	* Variable
2	Bicycle	200	* Variable
3	Bottle	200	* Variable
4	Boat	200	* Variable
5	Dog	200	* Variable
6	Tv Monitor	200	* Variable
7	Person	200	* Variable
8	Cat	200	* Variable

CHAPTER 4

FEATURES FOR IMAGE RETRIEVAL

Introduction

Based on our literature review, we decided that we would use some of the most popular image parameters used in different fields such as medical images, text recognition, natural scene images and traffic images [6, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97], as a common denominator the usefulness of all these image parameters have shown to be very domain dependent. Based on the fact that our framework was developed in parallel with the SDO CBIR system, we perform experiments and quantitative analysis with the 10 most useful parameters we determined for our domain-specific dataset. However, the framework capabilities allow researchers to quickly implement and test parameters that are most appealing for them. The framework allows the usage of software packages that generate image parameters like WND-CHARM [98] as a starting point for CBIR development and provides extensive facilities to test them and verify their applicability to the target domain of the users CBIR system. The default parameters of the framework are mainly textured based and shown to be very useful for grayscale images that do not contain very sharp edges or well defined objects. In this chapter we will describe them in detail in section 4.2; we also present some general purpose image parameters that have wide popularity in the community and are very useful in other domains in the Color Based Image Descriptors section (4.3) and the Shape/Region Based Image Descriptors on section (4.4.). Since new image parameters

are being published almost every day by image analysis experts, testing and implementing every single one of them would be a nearly impossible task, but the framework easily allows the user to test and implement their own.

Textural Image Parameters

Based on our literature review [6, 86, 87, 88, 89, 90, 91] and previous work [82, 83, 750], we decided to include in our framework and use for our solar CBIR system the following texture based image parameters:

Mean [99]: This image parameter is the traditional statistical mean or average of the pixel values of the input image or image segment.

$$m = \frac{1}{L} \sum_{i=0}^{L-1} z_i \quad (4.1)$$

Where L is the number of pixels in our image/segment, and z_i is the i -th pixel value.

Standard Deviation [99]: This image parameter shows how much variation there is from the mean of the pixel values. A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data are spread out over a large range of values.

$$\sigma = \sqrt{\frac{1}{L} \sum_{i=0}^{L-1} (z_i - m)^2} \quad (4.2)$$

Where L is the number of pixels in our image/segment, z_i is the i -th pixel value, and m is the mean.

Third Moment (skewness) [99]: This image parameter measures the lopsidedness of the histogram distribution of input image or image segment. Any symmetric distribution will have a value of zero.

$$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i) \quad (4.3)$$

Where L is the number of pixels in our image/segment, z_i is the i -th pixel value, m is the mean, and $p(z_i)$ is the grayscale histogram representation of z at i .

Fourth Moment (Kurtosis) [99]: This image parameter measures of whether the histogram distribution of the input image or image segment is tall and skinny or short and squat when compared to the normal distribution of the same variance. This image parameter is always positive.

$$\mu_4 = \sum_{i=0}^{L-1} (z_i - m)^4 p(z_i) \quad (4.4)$$

Where L is the number of pixels in our image/segment, z_i is the i -th pixel value, m is the mean, and $p(z_i)$ is the grayscale histogram representation of z at i .

Relative Smoothness (RS) [99]: This image parameter is a measure of gray level contrast of the pixel values of the input image or image segment.

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (4.5)$$

Where σ^2 is the square of the standard deviation of z .

While Relative Smoothness and Standard Deviation are functionally dependent, please note the dependence is quadratic. This will result in the values of Standard Deviation to be more spread out than the ones for Relative Smoothness in certain ranges.

Following [178, 179, 180], we decided to keep both parameters, hoping that this spreading of the data will separate some of our classes more effectively.

Entropy [99]: This image parameter is a statistical measure of randomness that characterizes the histogram the input image, or image segment.

$$E = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (4.6)$$

Where L is the number of pixels in our image/segment, z_i is the i -th pixel value, m is the mean, and $p(z_i)$ is the grayscale histogram representation of z at i .

Uniformity [99]: This image parameter is a measure of the uniformity of the histogram of the input image or image segment.

$$U = \sum_{i=0}^{L-1} p^2(z_i) \quad (4.7)$$

Where L is the number of pixels in our image/segment, z_i is the i -th pixel value, m is the mean, and $p(z_i)$ is the grayscale histogram representation of z at i .

Fractal Dimension [100]: This image parameter measures how rough a surface is by providing a statistical quantity that gives an indication how fine grained a fractal structure is and how it completely fills a space. If the fractal dimension has a value of one, it means the texture is entirely distorted. This image parameter is scale and linear transformation invariant. The fractal dimension is calculated using the box counting method in our framework.

$$D_0 = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log \frac{1}{\epsilon}} \quad (4.8)$$

Where $N(\varepsilon)$ is the number of boxes of side length ε required to cover the z .

The *Tamura Textures* Introduced in [101], the authors propose six texture features corresponding to human visual perception: coarseness, contrast, directionality, line-likeness, regularity, and roughness. From experiments used to test the significance of these features with respect to human perception, it was concluded [6] that the first three features are more useful than the rest.

Tamura Coarseness: This image parameter gives information about the size of the texture elements. The higher the coarseness value is, the rougher the texture is.

Tamura Contrast: This image parameter stands for picture quality; this parameter is influenced by the following four factors: 1) dynamic range of gray-levels, 2) how polarized the distribution of black and white levels is on the gray-level histogram, 3) sharpness of edges, and 4) period of repeating patterns.

Tamura Directionality: This image parameter does not rely on the orientation itself, rather on the presence of orientation in the texture. For example, two textures that differ only in the orientation are considered to have the same directionality.

While this list is not 100% exhaustive, like we mentioned before, trying everything is out of the question. Using a tool like WND-CHARM [98] can give the user access to many other image parameters and they can be easily incorporated to be used in conjunction with the analysis facilities of our framework.

A final list of the image parameters utilized for our experimental evaluations in through the chapters of this dissertation will be provided in the section 4.5.

Color Based Image Parameters

For datasets that feature color images we include some discussion about the most popular image parameters currently used by researchers in the field of image retrieval and that have shown to produce good results in current CBIR systems. Color is one of the most widely used visual features in image and video retrieval since color features are relatively robust to the viewing angle, translation and rotation of the regions of interest. We again mention that we cannot cover all of them, but we have selected a solid mix to highlight the state of the art for these types of image parameters.

Color Histograms [38]: These histograms are one of the most basic image features that have great popularity in the community [102, 103, 104].

In order to generate the histograms, the color space is partitioned and for each of the partitions the pixels with a certain color range are counted. The result is a representation of frequencies of the occurring colors.

Most researchers use the RGB color space to create these histograms, after researchers have shown [27] that other color spaces offer very minor improvements.

Color Layout Descriptor (CLD) [105]: This image parameter captures the spatial distribution of color in an image or segment of an image. The spatial layout of the representative colors on a grid superimposed on a region or image is based on coefficients of the Discrete Cosine Transform (DCT). This descriptor is calculated by applying the DCT transformation on a 2-D array of local representative colors in YCbCr color space. CLD is utilized in the MPEG-7 standard.

Dominant Color Descriptor (DCD) [106]: This image parameter provides a compact description of the representative colors in an image or image region. CDC allows a specification of a small number of dominant color values as well as their statistical properties such as distribution and variance. The parameter's purpose is to provide an effective, compact and intuitive representation of colors present in a region or image. DCD is also utilized in the MPEG-7 standard.

Scalable Color Descriptor (SCD) [107]: This image parameter represents a color Histogram in the HSV Color Space encoded by a Haar transform. SCD is useful for image-to-image matching and retrieval based on color feature. SCD is utilized in the MPEG-7 standard.

Group of Frames (or Pictures) Descriptor [108]: This image parameter is an extension of the Scalable Color Descriptor to a group of frames in a video or a collection of pictures. It is based on the premise of aggregating the color properties of the individual images or video frames. The Group of Frames Descriptor is utilized in the MPEG-7 standard.

Color Structure Descriptor (CSD) [109]: This image parameter is also based on color histograms, and aims at identifying localized color distributions using a small structuring window. The Color Structure Descriptor is bound to the HMMD color space. CSD is utilized in the MPEG-7 standard.

Shape/Region Based Image Parameters

Detecting objects in images in order to perform image retrieval has been a very relevant topic in the last few years. While the textural and color based image parameters are good for multi-purpose approaches that require to match images based on their similarity, there are countless image parameters that focus on extracting information of the image based on the shapes and objects encountered within them. While these parameters are specialized and very thorough, they pose a problem to researchers in terms of how big they are in size and how computationally expensive they are. While we didn't have to use any of them in our solar image analysis based on the fact that we have no sharp edges and prominent objects in our figures, we performed some experimental analysis with one of them and obtained the very discouraging results found on figure 4.1. We again mention that we cannot cover all of the texture/shape based parameters available in the literature, but we have selected a solid mix to highlight the state of the art for these types of image parameters.

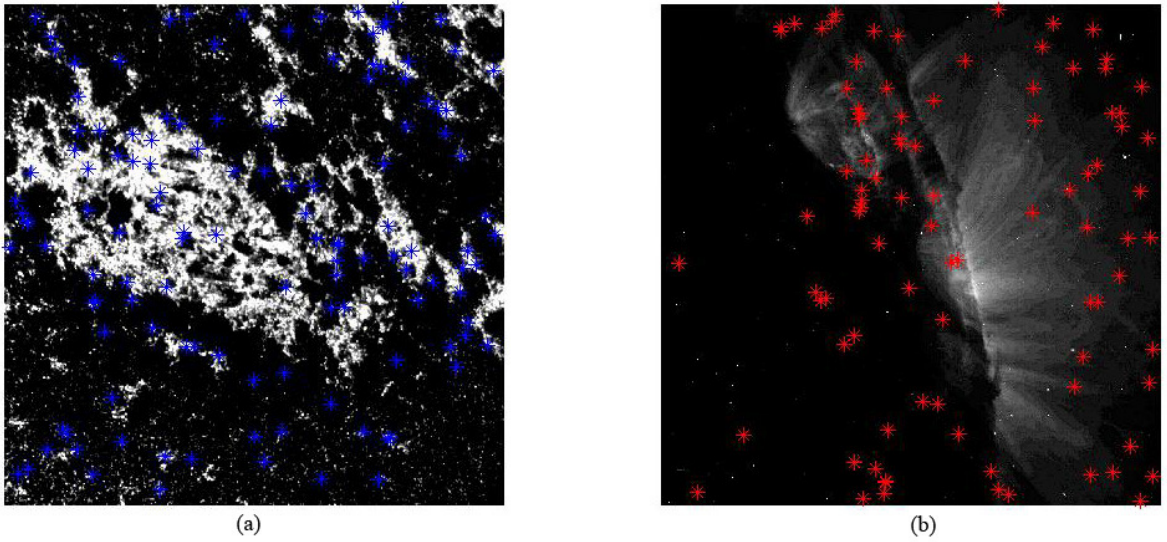


Figure 4.1. 2 Highly similar images according to the SIFT descriptors. (a) belongs to the Emerging Flux Class and (b) Filament class. As is clearly evident, both images are in fact very different.

Scale-Invariant Feature Transform (or SIFT) [110]: Originally proposed by David Lowe in [110], this image parameter transforms an image into a large collection of feature vectors. Each of these vectors is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. Key locations in the SIFT features are defined as maxima and minima of the result of a difference of Gaussians function applied in a scale-space to a series of smoothed and re-sampled images from the original image or segment. The low contrast candidate points and the edge response points along an edge are discarded. Next, dominant orientations are then assigned to localized keypoints. With these steps SIFT features ensure that the keypoints are more stable for matching and recognition. In our experiments (figure 4.1) and on the framework's selected implementation of the SIFT features, we produce a 128 feature vector.

Speeded Up Robust Features (SURF) [111]: Proposed in 2006, SURF is a robust image detector & descriptor, the method is based on sums of approximated 2D Haar wavelet transforms and making efficient use of integral images (unlike the SIFT re-sampling of images). SURF is very fast since it uses an integer approximation to the determinant of Hessian blob detector [112], which can be computed extremely fast with an integral image. For the features, SURF uses the sum of the Haar wavelet transform around the point of interest. This again can be computed with the aid of the integral image. One of the main advantages over SIFT is that the SURF method is not patented making implementations easily available for the community, hence more popular and easy to deploy.

Maximally Stable Extremal Regions (MSER) [113]: MSER generates features by thresholding. All the pixels below a certain threshold are labeled as black and all those above or equal to the threshold are white. If we are shown a sequence of thresholded images I_t with frame t corresponding to a threshold t , we would see first a white image, then black spots that correspond to local intensity minima will appear and then grow larger. These black spots will eventually merge until the whole image is black. The set of all connected components in the sequence is the set of all extremal regions (our features).

Because these regions are defined exclusively by an intensity function in the region and the outer border over a large range of thresholds, the local binarization is stable in some regions and has the following properties. 1) The extremal regions are invariant to affine transformation of image intensities. 2) Only the region whose support is nearly the same over a range of thresholds is selected, making it very selective. 3)

Multi-scale detection is performed with any smoothing involved. This allows the detection of both fine and large structures. 4) The set of all extremal regions can also be enumerated in worst-case $O(n)$, where n is the number of pixels in the image.

Image Parameter Extraction

Grid-based Image Segmentation

A solar phenomenon tends to occur in a small region of the solar images we are analyzing (see figure 4.2). By this observation, we can intuit that calculating our image parameters for the whole image will render them useless. Based on previous CBIR work done for the TRACE mission [9] and our own observations [82], dividing the image with an n by n grid before extracting the image parameters has been a popular and effective approach.

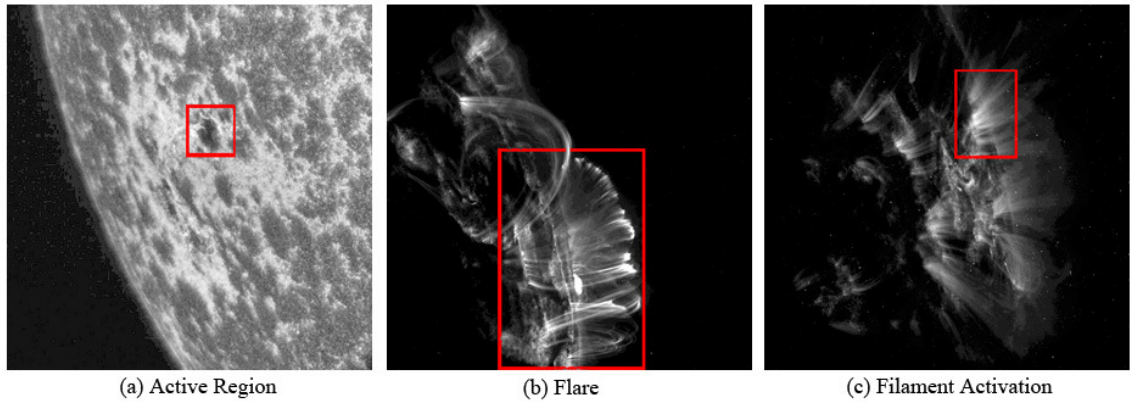


Figure 4.2. Regions of interest in this sample image from the (a) Active Region,. (b) Flare, and (c) Filament Activation classes

Our previous work [82], has shown that the most effective grid size for our current images in the Solar dataset is 128 by 128 pixels. This in turn returns a grid size of 8 by 8 segments or cells. After this image segmentation is performed, we proceed to

generate our feature vector for each image by extracting the image parameters for each of the cells of each image. This procedure is illustrated on figure 4.3.

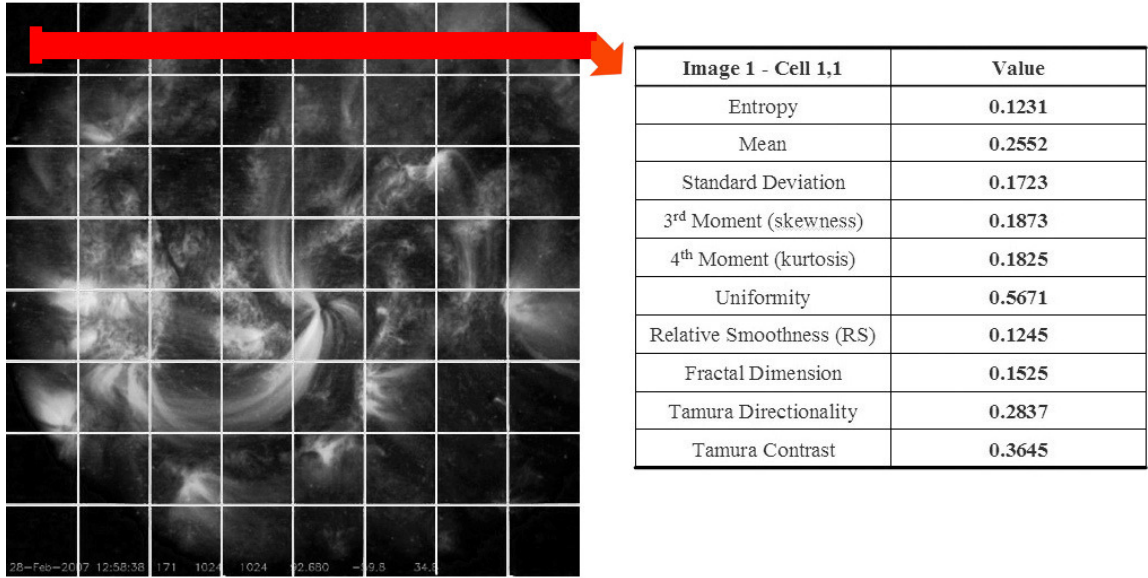


Figure 4.3. Example of our 128 by 128 pixel grid and image parameter extraction from one of the resulting cells

Computational Costs

Here we present the running times of the extraction of the proposed image parameters. Figure 4.4 presents the time needed to extract our image parameters on 1,600 images, taking into consideration that each image has been segmented in 64 cells. We have a total of 102,400 segments to extract image parameters from on our solar dataset.

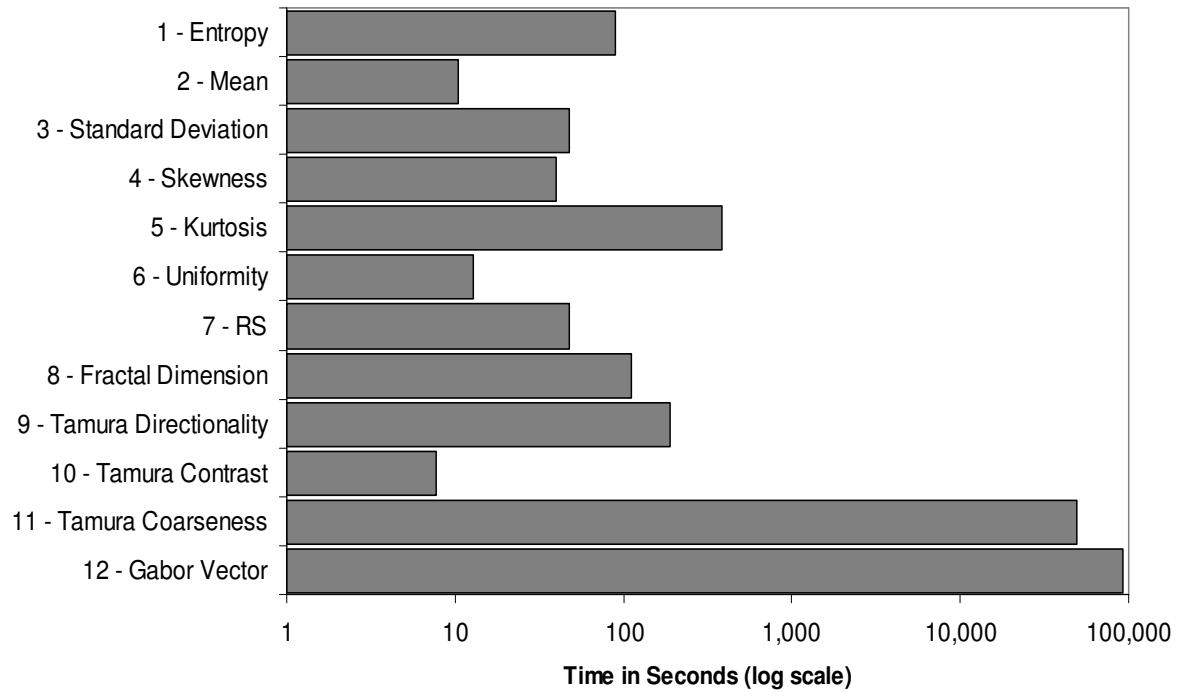


Figure 4.4. Image Parameter Extraction Times for 1,600 Images

The reason behind dropping expensive image parameters is that we will have to process data in near real time for the SDO mission, at a rate of eight 4096x4096 images every 10 seconds, so we will have a smaller window than this amount of time to be able to extract all the image parameters that are the most useful and cheapest to calculate.

After analyzing figure 4.2, the Tamura feature coarseness and the Gabor vector were discarded because of their costly computational expense. While the usefulness of these parameters has not been tested, the extreme computational cost of these 2 image parameters makes them immediate candidates to be discarded. From this chapter on, we will use only 10 different image parameters for our future evaluation results. Table 4.1 indicates the list of these 10 parameters and their label.

Table 4.1. List of Extracted Image Parameters

Label	Image parameter
P1	Entropy
P2	Mean
P3	Standard Deviation
P4	3 rd Moment (skewness)
P5	4 th Moment (kurtosis)
P6	Uniformity
P7	Relative Smoothness (RS)
P8	Fractal Dimension
P9	Tamura Directionality
P10	Tamura Contrast

All experiments in this dissertation were performed using Matlab R2009b and WEKA 3.6.4. These programs were run on a PC with AMD Athlon II X4 2.60 Ghz Quad Core processor with 8 GB's of RAM and Windows XP 64-bit Edition

CHAPTER 5

CLASSIFICATION ALGORITHMS FOR EVALUATION

Introduction

In order to provide quantitative evaluation results in each of the steps of our framework. The framework generates ARFF files that are compatible with WEKA [11] that will enable the user to test their experiments on the wide variety of classification algorithms offered in WEKA such as Naïve Bayes (NB) [114], ID3 [115], C4.5 [116], Support Vector Machines [117], Random Forests [118], etc. as well as boosting algorithms available via WEKA for said classifiers. The utilization of a popular tool such as WEKA allows the users of the framework to test on newer classifiers and use some the visualization tools offered by the software package. For our solar dataset and our comparative evaluation experiments in this work we selected three different classifiers to present our evaluation results on: Naïve Bayes (NB) [114], C4.5 [116] and Support Vector Machines [117] (SVM). We selected two linear classifiers (NB and SVM's with a linear kernel) because they achieve the grouping of items by allocating similar feature values into groups making a classification decision based on the value of the linear combination of the features (NB and SVM's use a different criterion to do this as outlined in sections 5.2 and 5.3 respectively). C4.5 is a decision tree classifier and uses an entropy-based information gain measure to split samples into classes, making it considerably different from the previous two classification algorithms. The framework

also provides scripting facilities to run experiments and retrieve results for these three classifiers and can easily be modified to adopt different ones.

Naïve Bayes Classifier

First introduced in [114], the Naïve Bayes (NB) classifier is based on Bayes' theorem and it assumes that the presence (or absence) of a particular feature of certain class is unrelated to the presence (or absence) of other features. In technical terms, based on Bayes theorem the classifier works in the following way:

Let X be a record whose class label is unknown. Let H be a hypothesis, such as "record X belongs to a class C ." For classification of this record, we want to determine $P(H|X)$ -- the probability the hypothesis H holds, given the observed record X .

$P(H|X)$ is the posterior probability of H conditioned on X , $P(H)$ is the prior probability of H and $P(X)$ is the prior probability of X . The posterior probability, $P(H|X)$, is based on more information (i.e. background knowledge) than the prior probability, $P(H)$, which is independent of X . Bayes theorem is useful since it provides a way of calculating the posterior probability, $P(H|X)$, from $P(H)$, $P(X)$, and $P(X|H)$. In the following way: $P(H|X) = P(X|H) P(H) / P(X)$

Even with such an independence assumption, the Naïve Bayes classifier has shown to be surprisingly accurate in most cases [119] and executes very fast, making it a highly useful classification algorithm for training on the massive repositories.

C4.5 Classifier

The C4.5 Classifier was first introduced in [116], this decision tree based classifier is an extension of the ID3 algorithm. This algorithm builds decision trees from a set of training data in a similar way than ID3 does, by using the concept of information entropy. The training data is a set $X = x_1, x_2, \dots, x_i$ of previously classified samples. Each sample $x_i = y_1, y_2, \dots, y_i$ is a vector where $y_1, y_2, \dots, y_i$ represent features of the sample. The training data is augmented with a vector $C = c_1, c_2, \dots, c_i$ where $c_1, c_2, \dots, c_i$ represent the class to which each sample belongs. At each node of the tree, the algorithm chooses one attribute of the data that most effectively splits its set of samples into smaller subsets enriched in one of the classes. The splitting criterion is based on the normalized information gain. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller subsets, the algorithm has several main cases:

- If all the samples in the dataset belong to the same class. The algorithm simply creates a leaf node for the decision tree saying to choose said class.
- None of the features provide any information gain. If this happens, the algorithm creates a decision node higher up in the tree using the expected value of said class.
- If a previously-unseen class is encountered. The algorithm creates a decision node higher up the tree using the expected value.

C4.5 is one of the most popular algorithms in the community [120], some of its advantages are that it takes a greedy approach and is also quick to compute.

Support Vector Machines Classifier

Originally developed by Vapnik in [117], the algorithm takes a set of input data points and predicts, for each given input, which of two possible classes the input is a member of. This makes the SVM a non-probabilistic binary linear classifier. Based on this, the training algorithm builds a model that predicts whether a new sample data point is member of one category or another. In other words, the SVM model is a representation of the sample points as points in a space that are mapped so that the examples of the different categories are separated by a clear gap that is as wide as possible. New sample points are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, that are used for classification. A good separation is achieved by the hyperplane that has the greatest distance to the nearest training data points of any class.

Most of the time the original problem is stated in a finite dimensional space, however, it often happens that in said spaces the sets of points to be discriminated are not linearly separable. For this reason the original finite dimensional space is mapped into a much higher dimensional space, with the hopes to make the separation easier in that space. SVM schemes use the mapping into a larger dimensional space allowing the cross products to be computed easily in terms of the variables in the original space, this makes the computational expense reasonable. The cross products in the larger dimensional space are defined in terms of a kernel function $K(x_i, y)$ particularly selected for different problems (in our solar case we use linear kernel functions). The hyperplanes in the large

space are defined as the set of points whose cross product with a vector in that space is constant. The vectors defining the hyperplanes are chosen to be linear combinations with parameters α_i of feature vectors that occur in the dataset. With this choice of a hyperplane, the points x in the feature space that are mapped to the hyperplane are defined by the relation: $\sum_i \alpha_i K(x_i, y) = \text{constant}$

Note that if $K(x_i, y)$ becomes small as y grows further from x , each element in the summation measures how close the test point x is to the corresponding data base point x_i . In this way the sum of kernels can be used to measure the nearness of each test point to the data points originating in one or the other of the sets to be classified.

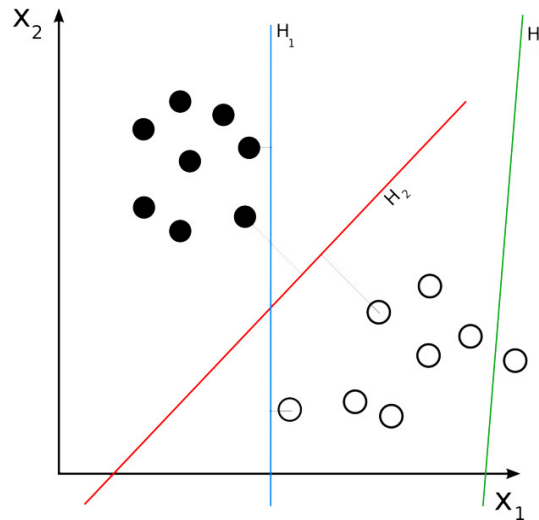


Figure 5.1. Examples of 3 SVM Hyperplanes in two dimensions. H_3 doesn't separate the 2 classes (black and white dots). H_1 does but with a small margin. H_2 achieves separation with the maximum margin. Figure from [121].

Classification Measures Utilized for Evaluation

In order to measure the performance of our classification algorithms we will use the following standard measures: Precision, Recall, F-Measure, Accuracy and Receiver Operating Characteristic (ROC) curves. These measures are defined as follows:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (5.1)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (5.2)$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5.3)$$

$$Accuracy = \frac{true\ positives + true\ negatives}{true\ positives + false\ positives + false\ negatives + true\ negatives} \quad (5.4)$$

The proposed measures derive from a **confusion matrix**. This matrix on each column represents the instances in a classified class, while on each row represents the instances in the actual class as show in figure 5.2.

		Classifier Prediction	
		Positive	Negative
Actual Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Figure 5.2. Example of a confusion matrix or truth table

Besides the previous measures, we also present the **Receiver Operating Characteristic (ROC)** curves for the best performing classifiers in order to observe how precise our accuracy is for certain classes within the datasets and to compare individual datasets on a per-class basis. The ROC curves are represented by plotting true positive rate vs. the false positive rate.

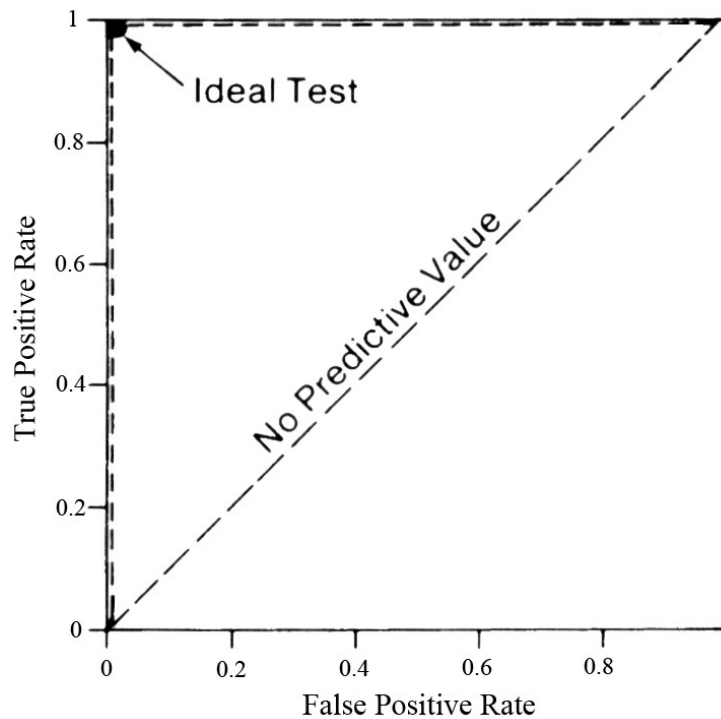


Figure 5.3. Optimal example of a ROC Curve evaluation. The closer the curve is to the Ideal Test, the better the classification

While all of these values might not be presented directly in our experimental evaluations, they can be found on the web at [122]. Values for all of these measures are outputted from the WEKA files that the framework conveniently generates for the user to perform their quantitative experimentation and are easily replicable by other researchers using the files provided at [122].

Table 5.1. Classifiers and their function calls used for this dissertation

Label	Classifier Name	WEKA Code and Parameters Used
NB	Naïve Bayes	<code>weka.classifiers.bayes.NaiveBayes</code>
C45	Decision Tree C4.5	<code>weka.classifiers.functions.SMO -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1</code>
SVM	Support Vector Machines	<code>weka.classifiers.trees.J48 -C 0.25 -M 2</code>

Initial Classification Evaluation Results

After the initial extraction of our proposed image parameters from table 4.1 (on chapter 4), in figure 5.4 we present the classification accuracy results for all 3 classifiers on the 6 datasets we analyzed. These results are presented to asses how accurate the image parameters/classifiers are before performing any of the attribute evaluation, dissimilarity evaluation and dimensionality reduction experiments presented in their respective chapters on this dissertation.

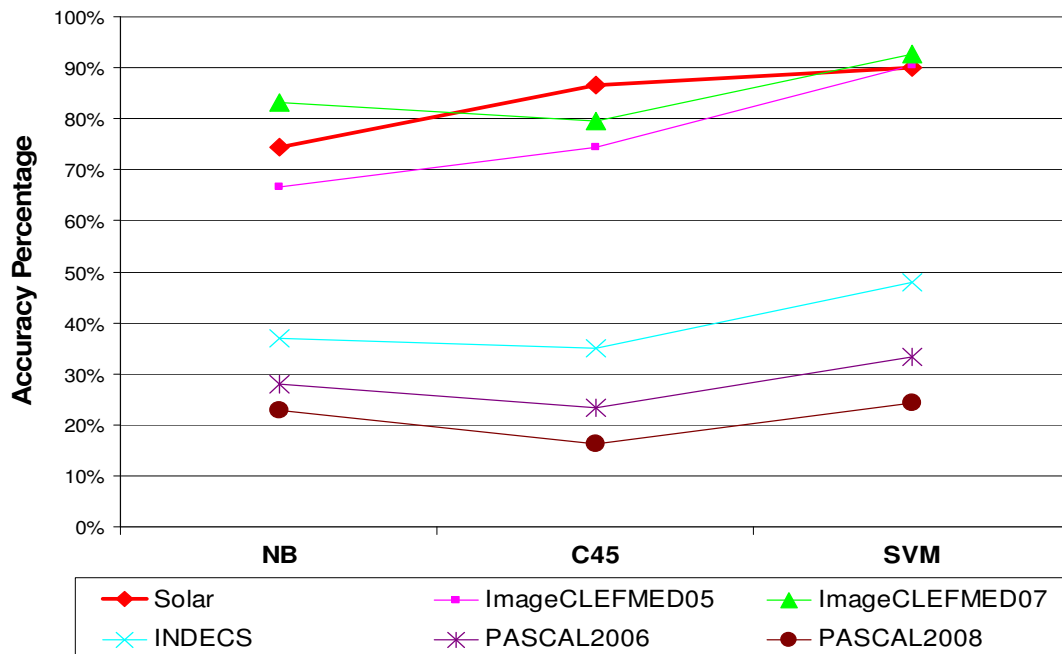


Figure 5.4. Initial classification accuracy for our six datasets, experiments using 10-fold cross-validation

These classification results are in no way optimal and do not represent the final classification accuracy on our dataset after it gets analyzed through the steps in this framework. However, we can start to observe the domain dependency of the image parameters in figure 5.4 since we have two main clusters in the classification accuracy results. The first cluster includes the solar and the medical datasets and the second cluster presents the PASCAL and INDECS datasets performing very poorly. More insights about why this happens will be presented in the remaining chapters of this dissertation. In the following figures and tables we present the precision, recall and F-Measure values for the 8 classes of the solar dataset. We also show the ROC Curves for all 8 our solar dataset for all 3 different classifiers to familiarize the reader with them.

Table 5.2. Precision values for the classes of the solar dataset (closer to 1 is better)

Class \ Classifier	NB	C45	SVM
Active Region	0.9650	0.9744	0.9898
Coronal Jet	0.6364	0.8743	0.9000
Emerging flux	0.9643	0.9848	0.9704
Filament	0.7196	0.8295	0.9397
Filament Activation	0.7486	0.9059	0.9694
Filament Eruption	0.5556	0.8177	0.8850
Flare	0.6788	0.7129	0.7474
Oscillation	0.7302	0.8333	0.7913
Average	0.7498	0.8666	0.8991

Table 5.3. Recall values for the classes of the solar dataset (closer to 1 is better)

Class \ Classifier	NB	C45	SVM
Active Region	0.9650	0.9500	0.9700
Coronal Jet	0.7000	0.8350	0.9450
Emerging flux	0.9450	0.9750	0.9850
Filament	0.7700	0.9000	0.9350
Filament Activation	0.6550	0.9150	0.9500
Filament Eruption	0.6750	0.8300	0.8850
Flare	0.4650	0.7200	0.7100
Oscillation	0.7850	0.8000	0.8150
Average	0.7450	0.8656	0.8994

Table 5.4. F-Measure values for the classes of the solar dataset (closer to 1 is better)

Class \ Classifier	NB	C45	SVM
Active Region	0.9650	0.9620	0.9798
Coronal Jet	0.6667	0.8542	0.9220
Emerging flux	0.9545	0.9799	0.9777
Filament	0.7440	0.8633	0.9373
Filament Activation	0.6987	0.9104	0.9596
Filament Eruption	0.6095	0.8238	0.8850
Flare	0.5519	0.7164	0.7282
Oscillation	0.7566	0.8163	0.8030
Average	0.7434	0.8658	0.8991

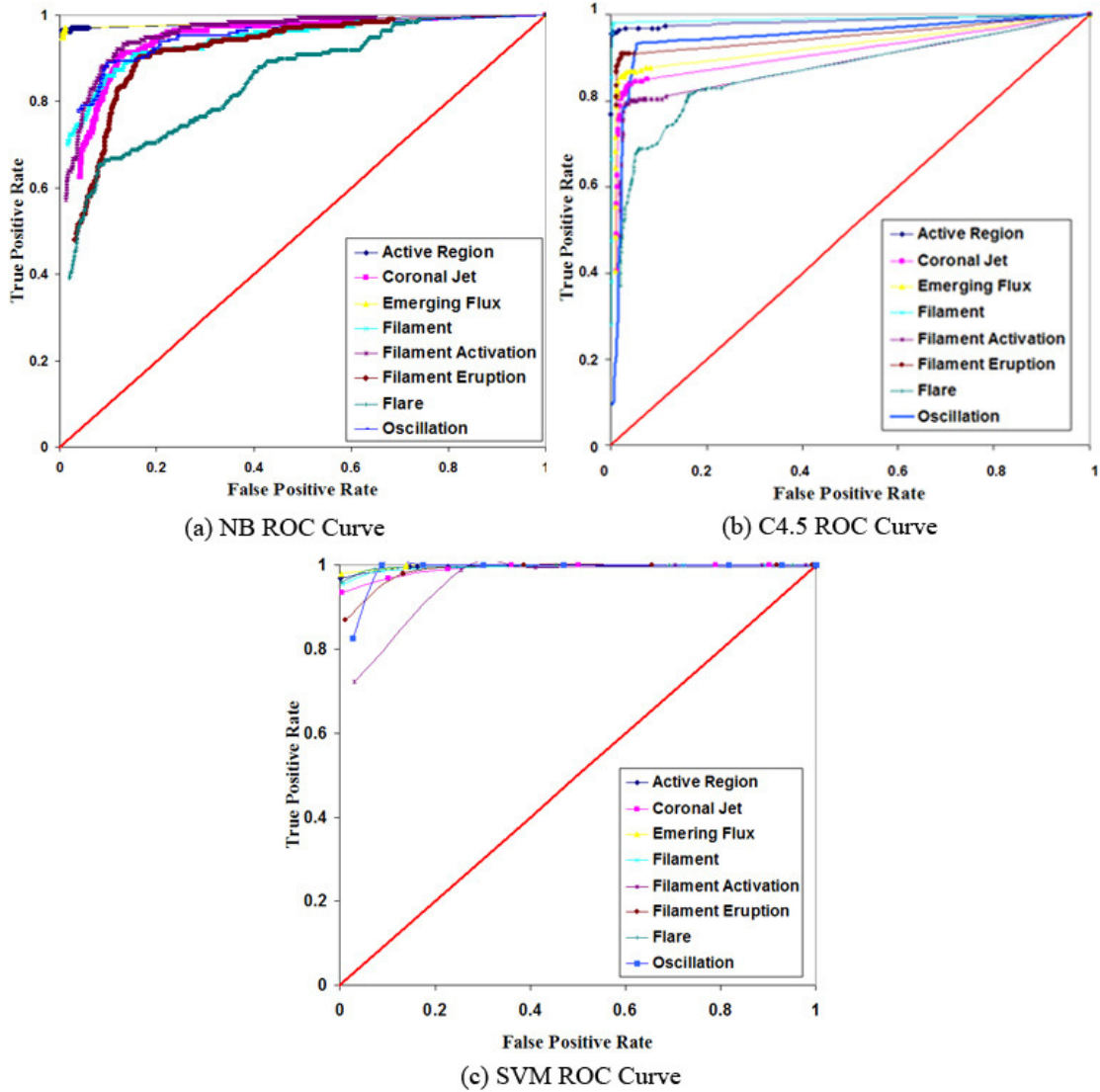


Figure 5.5. Initial ROC Curves for the 8 classes of the solar dataset, experiments using 10-fold cross-validation

CHAPTER 6

ATTRIBUTE EVALUATION

Introduction

Choosing one redundant image parameter in a CBIR system would add unneeded computational overhead when processing images, unnecessary data storage, and retrieval costs in our repository. One of the main goals of this chapter is to determine which image parameters we can safely remove while maintaining high quality parameter-based representation for our CBIR systems, in our case the SDO CBIR system.

We decide to start our work on by performing unsupervised attribute evaluation of the extracted image parameters. Our decision behind using a correlation comparison is based on the premise that we can achieve dimensionality reduction by finding strongly correlated (i.e redundant) image parameters. The use of statistical analysis seems to be more accepted within the astrophysics community and more popular than heuristics and supervised dimensionality reduction techniques (often used by experts coming from AI-related fields). The correlation based approach was also a starting point for the development of CBIR systems in other domains [6]. To coincide with this unsupervised approach, we also selected several supervised attribute evaluation methods: Chi Squared, Gain Ratio, and Info Gain. We experimented with this supervised methods in order to verify our findings from the unsupervised attribute analysis we performed. The framework outlined in this dissertation incorporates both methods (supervised and unsupervised) of attribute evaluation to allow the user test what works best with their

domain-specific CBIR and aid in the reduction of storage, retrieval and processing costs of his system.

Unsupervised Attribute Evaluation

Correlation Analysis

Automatic methods for image parameter selection have been proposed in [123, 124]. However, these automatic methods do not directly explain why features are chosen. The method proposed in [6] analyses correlations between the values of the parameters themselves, and instead of automatically selecting a set of features, provides the user with information to help them select an appropriate set of features.

To analyze the correlation between different image parameters, we evaluate the correlation between the Euclidean distances $d(q, X)$ obtained for each image parameter of each of the images X from the our benchmark given a query image q . For each pair of query image q and database image X we create a vector $(d_1(q, X), d_2(q, X), \dots, d_m(q, X), \dots, d_M(q, X))$ where $d_m(q, X)$ is the distance of the query image q to the benchmark image X for the m th image parameter, and M is the total number of image parameters. Then we calculate the correlation between the d_m over all queries $q = \{q_1, \dots, q_b, \dots, q_L\}$ and all images $X = \{X_1, \dots, X_n, \dots, X_N\}$.

The $M \times M$ covariance matrix, denoted as Σ_{ij} of the d_m is calculated over all N database images and all L query images as:

$$\Sigma_{ij} = \frac{1}{NL} \sum_{n=1}^N \sum_{l=1}^L (d_i(q_l, X_n) - \mu_i) \cdot (d_j(q_l, X_n) - \mu_j) \quad (6.1)$$

$$\text{with } \mu_i = \frac{1}{N L} \sum_{n=1}^N \sum_{l=1}^L d_i(q_l, X_n) \quad (6.2)$$

By segmenting each of our images into 64 grid cells (for our solar dataset), the number of queries we have is equal to the number of grid cells. Given the covariance

$$\text{matrix } \Sigma_{ij}, \text{ we calculate the correlation matrix } R \text{ as } R_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \Sigma_{jj}}} \quad (6.3)$$

The entries of this correlation matrix can be interpreted as similarities of different features. A high value R_{ij} means a high similarity between features i and j . This similarity matrix can then be analyzed to find out which of our parameters have highly correlated values and which do not. Figure 6.1 shows how the one-to-one grid cell comparison is done.

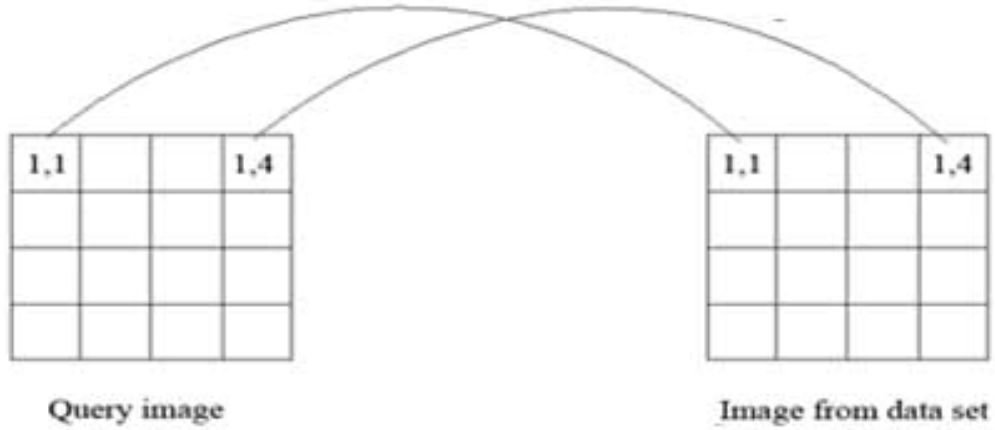


Figure 6.1. Shows the one-to-one comparison between Query image (q_l) cells against Images from the data set cells (X_n). 1,1 will only be compared to 1,1 and so on

Supervised Attribute Evaluation

Chi Squared Method

This method evaluates the worth of an attribute by computing the value of the chi-squared distribution with respect to the class [14].

Gain Ratio Method

This method evaluates the worth of an attribute by measuring the gain ratio with respect to the class [115]. This method biases the decision tree against considering attributes with a large number of distinct values. Solving the weakness presented by the information gain method.

Info Gain Method

This method evaluates the worth of an attribute by measuring the information gain with respect to the class [14]. A notable problem occurs when information gain is applied to attributes that can take on a large number of distinct values.

Evaluation Results for the Solar Dataset

Unsupervised Attribute Evaluation

In one scenario we randomly selected a query image from each class and analyzed the correlation between this image and the rest of the same class in order to observe which image parameters are strongly correlated and which ones are not within the same class. In the second scenario, we analyzed the correlation between the query images and the rest of the images of the same wavelength from all other classes in order to observe

which image parameters feature similar correlation to the rest of the classes than to the same class.

Since Active Regions are some of the most distinguishable classes to discover in our benchmark data set and they are also one of the basic indicators of solar activity occurrence, we decided to start our investigation by analyzing the correlations between queries of this class.

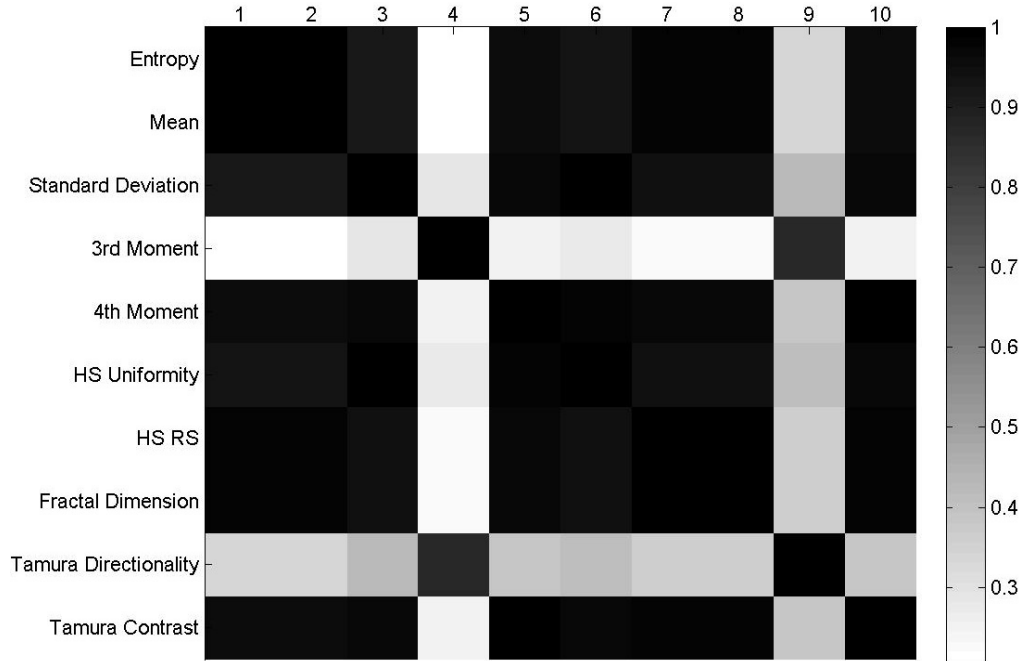


Figure 6.2. Average correlation map for the Active Region class in the one image as a query against the same class scenario (intra-class correlation)

In figure 6.2 we observe the correlation map generated by using one query image of the Active Region class against all the rest of the images within the same class. We can clearly see some image parameters that are weakly correlated against pretty much every other parameter, them being: 3^{rd} moment (P4) and *Tamura Directionality* (P9). In the strongly correlated section we have: *entropy* (P1), 3^{rd} moment (P4) and 4^{th} moment (P5)

with *uniformity* (P6) and *RS* (P7), *uniformity* (P6) with *RS* (P7) and *Tamura Contrast* (P10) with *standard deviation* (P3) and *mean* (P2).

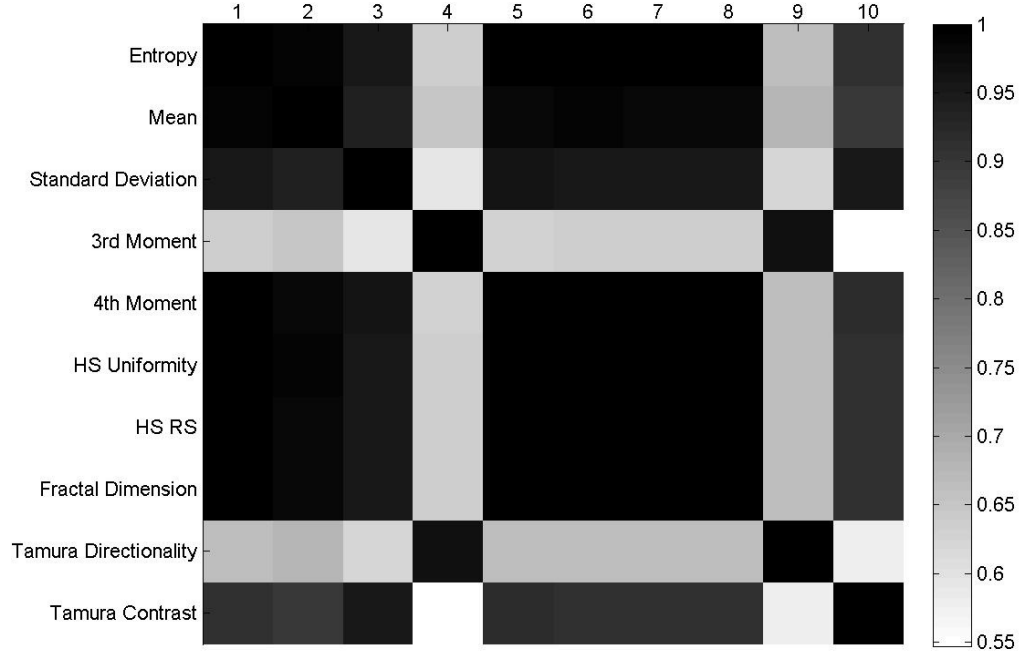


Figure 6.3. Average correlation map of the Active Region class in the one image as a query against other classes scenario (inter-class correlation)

In figure 6.3 we can observe the correlation map generated by selecting one query image of the Active Region class and calculating the correlations against all the images for the remaining classes. The weakly correlated parameters are: *3rd moment* (P4) with everything except *Tamura Directionality* (P9), *Tamura Directionality* (P9) with everything except *3rd moment* (P4). These findings are mostly consistent with the correlation analysis of the same class. As for the highly correlated image parameters we have: *entropy* (P1) and *4th moment* (P5), *uniformity* (P6) and *RS* (P7) and *Tamura Contrast* (P10) and *standard deviation* (P3) amongst others. Again these findings are very consistent with the correlation analysis within the same class.

Since the correlation differences are not that visible from the correlation maps, we decided to use Multi Dimensional Scaling (MDS) [125] to better visualize our correlations. Close parameters in the MDS 2D plots indicate strong correlation and the more distant they are the weaker their correlation. For all the MDS 2D plots we replaced the image parameter name with its corresponding number according to table 4.1 (this organization is seen in the x-axis of figures 6.2 and 6.3).

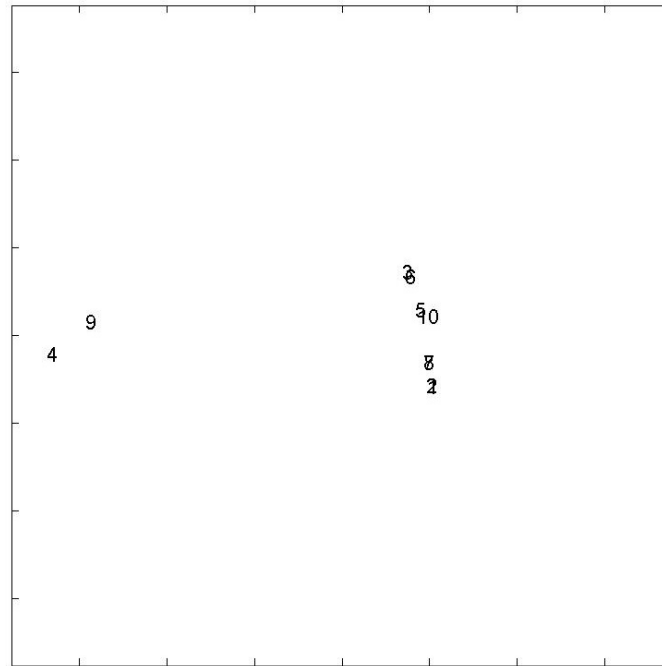


Figure 6.4. Multi Dimensional Scaling 2D plot of correlation map in Fig. 6.2

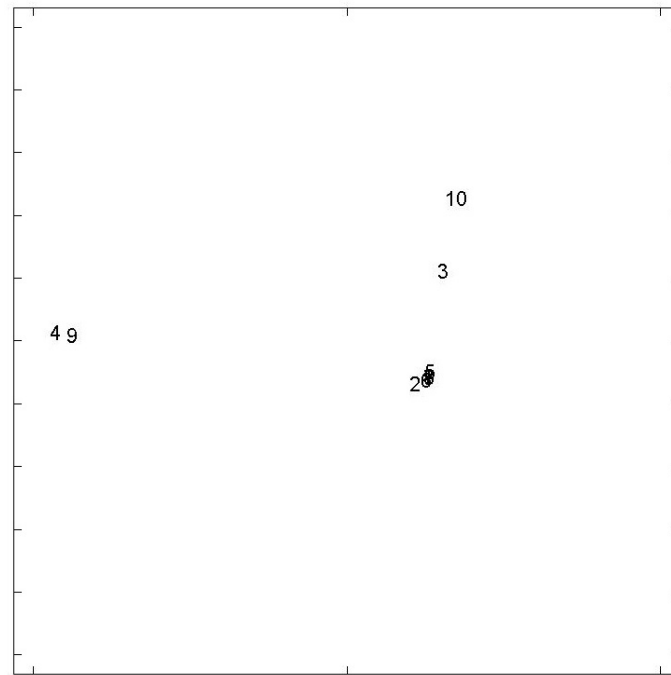


Figure 6.5. Multi Dimensional Scaling 2D plot of correlation map in Fig. 6.3

As we can see from the multi dimensional scaling maps in figures 6.4 and 6.5 we can identify almost identical clusters of highly correlated image parameters and the less correlated parameters can be found on the edges of the map.

Based on the correlation maps and the MDS 2D plots, we were able to perform a wide variety of experiments using classifiers in order to determine if we could attempt any dimensionality reduction to save our storage space and computing costs. These experiments are presented in section 6.4.3.

Supervised Attribute Evaluation

For the supervised attribute evaluation we present results from three different methods: Chi squared, Gain Ratio and Info Gain. Table 6.1 shows the attribute ranking results for these methods.

Table 6.1. Supervised Attribute Ranking Results

Chi Squared		Info Gain		Gain Ratio	
Ranking	Label	Ranking	Label	Ranking	Label
2039.06	P7	0.8440	P7	0.3597	P9
2036.05	P6	0.8425	P6	0.3152	P4
2016.69	P1	0.8292	P1	0.3140	P5
1887.89	P4	0.7848	P4	0.3116	P6
1883.68	P9	0.7607	P5	0.3102	P7
1839.05	P5	0.7552	P9	0.3025	P1
1740.33	P2	0.6882	P2	0.2781	P8
1578.76	P10	0.6627	P10	0.2647	P10
1370.68	P3	0.5702	P3	0.2493	P2
1134.43	P8	0.5344	P8	0.2399	P3

Based on the ranking results we have that on average the top four image parameters are P6, P7, P1 and P4 (as listed on Table 4.1). The bottom four parameters are P8, P3, P2 and P10 (as listed on Table 4.1). After looking at these results and comparing them to our unsupervised method, we see similar grouping of image parameters. Based on these experiments, we can now make educated assumptions as to which parameters whose removal we should experiment with in order to keep our image parameter extraction computational costs to a minimum, while maintaining levels of phenomena recognition comparable to the ones obtained by using all of the parameters.

Classification Results for Both Supervised and Unsupervised Attribute Evaluations

Utilizing the unsupervised and supervised attribute evaluation method results, we set up two different scenarios involving the removal of correlated and uncorrelated image parameters. Based on the ranking results presented in table 6.1, we created seven different experiments that are explained in Table 6.2. These experiments involve running

the selected classifiers and comparing the percentages of accurately classified instances while reducing the number of image parameters.

Table 6.2. Description of Attribute Evaluation Experiments

Label - Experiment Description
Exp 1 - All Parameters used
Exp 2 - Parameters P2, P9, P10 removed
Exp 3 - Parameters P3, P6, P10 removed
Exp 4 - Parameters P2, P3, P8 removed
Exp 5 - Parameters P1, P6, P7 removed
Exp 6 - Parameters P2, P3, P10 removed
Exp 7 - Parameters P1, P4, P7 removed

Experiments 2 and 3 are based on removing image parameters that are uncorrelated and correlated respectively, according to the unsupervised attribute evaluation we performed (figures 6.2-6.5). Experiments 4 and 6 are removing the bottom 3 attributes based on their ranking weights in the supervised attribute evaluation experiments. Experiments 5 and 7 remove the top 3 attributes bases on their ranking weights on the supervised attribute evaluation experiments.

Table 6.3. Classification Accuracy Results Obtained for our experiments

Label	NB	C45	SVM	Average
Exp 1	74.50%	86.56%	89.94%	83.67%
Exp 2	74.56%	84.69%	90.19%	83.15%
Exp 3	76.81%	85.75%	89.94%	84.17%
Exp 4	74.44%	85.13%	89.69%	83.08%
Exp 5	71.75%	83.50%	89.25%	81.50%
Exp 6	75.19%	85.25%	89.75%	83.40%
Exp 7	72.88%	85.56%	92.56%	83.67%

From Table 6.3 we can see that Exp 3, provides us with the best average classification results, also Exp 6 and Exp 7 give us good classification results and considering we removed 3 image parameters (or 30%) this is a very interesting result and

useful result. The main conclusion we can derive from this comparison is that the more we know about the image parameters and how they are related to each other, the better decision we can make in removing them without suffering a considerable loss in accuracy, in this case no loss in accuracy. We can see that the poor selection of some parameters that are more important in helping the classifiers identify trends in our data can decrease our classification accuracy like in Exp 5.

These experimental results clearly show the value and need of performing attribute evaluation in order to determine relevance of the images parameters selected for a CBIR system. By reducing even one parameter the storage, processing and retrieval costs can be considerably reduced, this will enable our CBIR system to better serve its purpose and be a useful tool for researchers. In the next few tables we present the average precision, recall and f-measure values for the experiments on table 6.2.

Table 6.4. Average Precision Values for our Attribute Evaluation Experiments

Label	NB	C45	SVM
Exp 1	0.7498	0.8666	0.8991
Exp 2	0.7465	0.8469	0.9010
Exp 3	0.7738	0.8590	0.8983
Exp 4	0.7467	0.8509	0.8951
Exp 5	0.7217	0.8342	0.8913
Exp 6	0.7592	0.8529	0.8962
Exp 7	0.7345	0.8560	0.9250

Table 6.5. Average Recall Values for our Attribute Evaluation Experiments

Label	NB	C45	SVM
Exp 1	0.7450	0.8656	0.8994
Exp 2	0.7456	0.8469	0.9019
Exp 3	0.7681	0.8575	0.8994
Exp 4	0.7444	0.8513	0.8969
Exp 5	0.7175	0.8350	0.8925
Exp 6	0.7519	0.8525	0.8975
Exp 7	0.7288	0.8556	0.9256

Table 6.6. Average F-Measure Values for our Attribute Evaluation Experiments

Label	NB	C45	SVM
Exp 1	0.7434	0.8658	0.8991
Exp 2	0.7445	0.8467	0.9008
Exp 3	0.7691	0.8581	0.8984
Exp 4	0.7405	0.8507	0.8956
Exp 5	0.7146	0.8343	0.8916
Exp 6	0.7531	0.8527	0.8963
Exp 7	0.7265	0.8557	0.9249

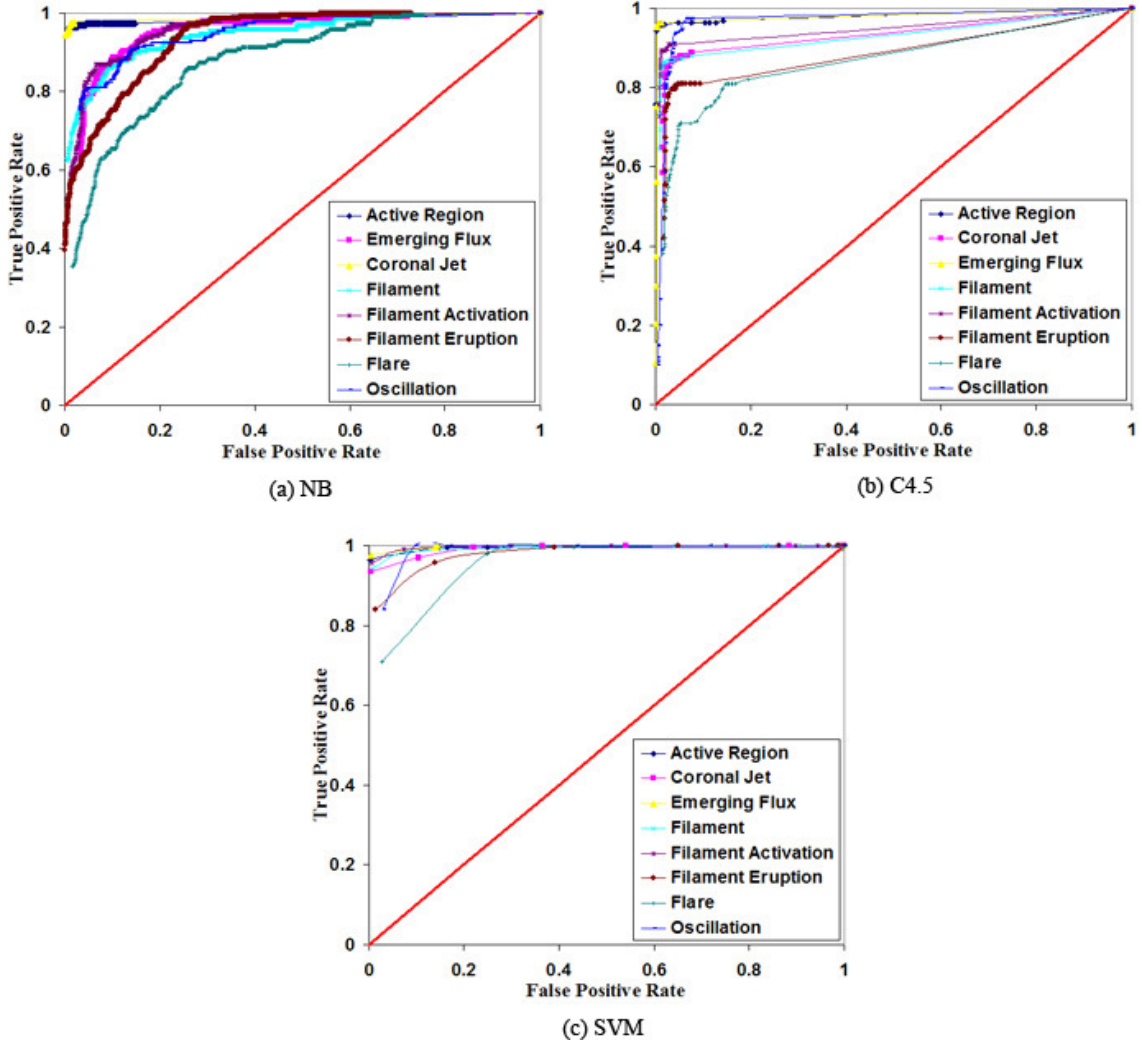


Figure 6.6. ROC Curves for the best performing experiment (Exp 3), experiments using 10-fold cross-validation

As we can see in figure 6.6, the ROC Curves remain almost identical than the original ones on figure 5.5. This indicates that we can safely assume we can remove 3 image parameters (30% of our feature vector) and still maintain good classification accuracy. In the next chapter we will analyze these parameters in detail with different dissimilarity measures to determine their usefulness in a different way. The following correlation maps for the remaining datasets are shown for illustrative purposes. For high quality versions of them and the MDS maps (not included here) go to [122].

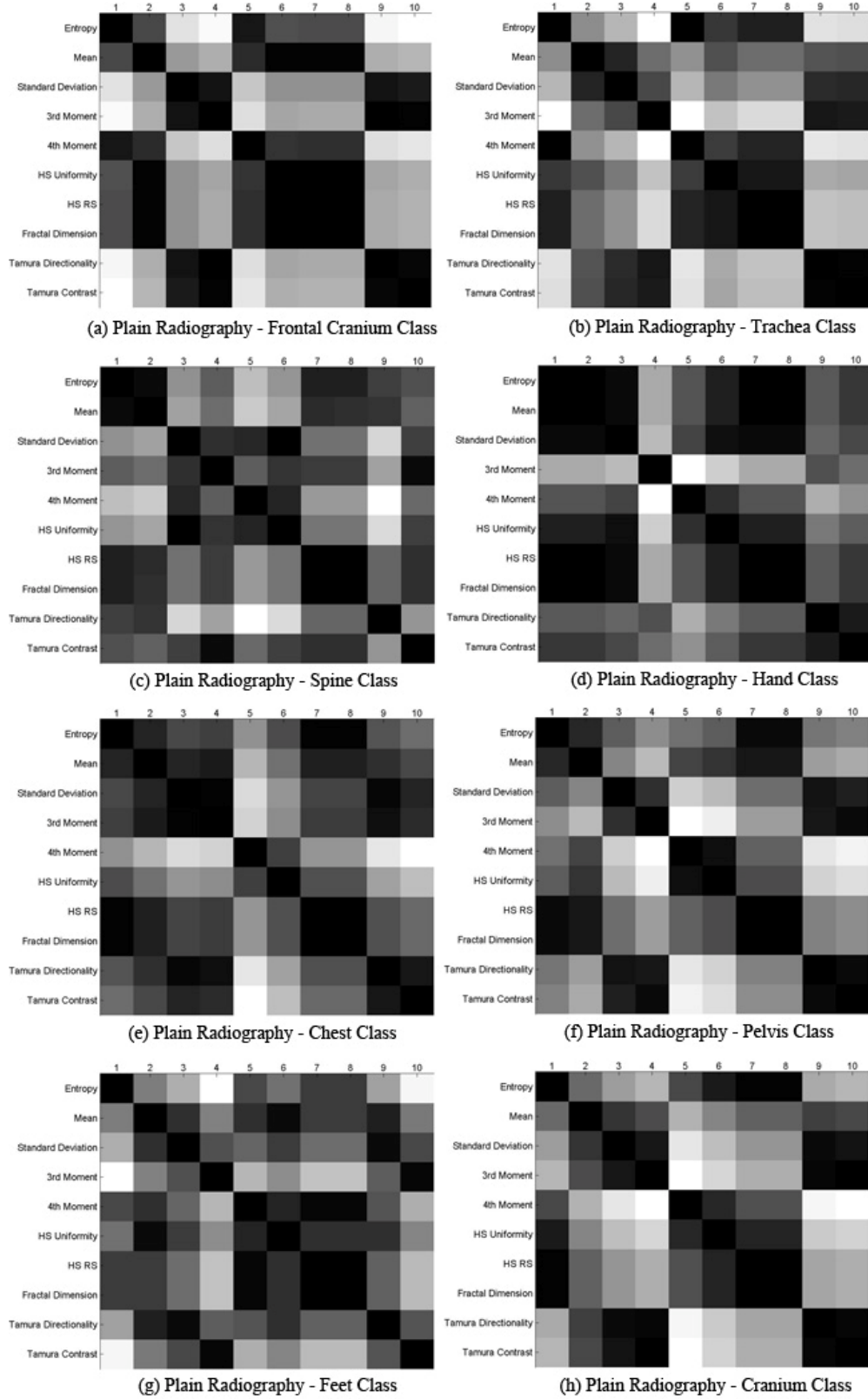


Figure 6.7. Inter-class Correlation Maps for the ImageCLEFMED05 Dataset

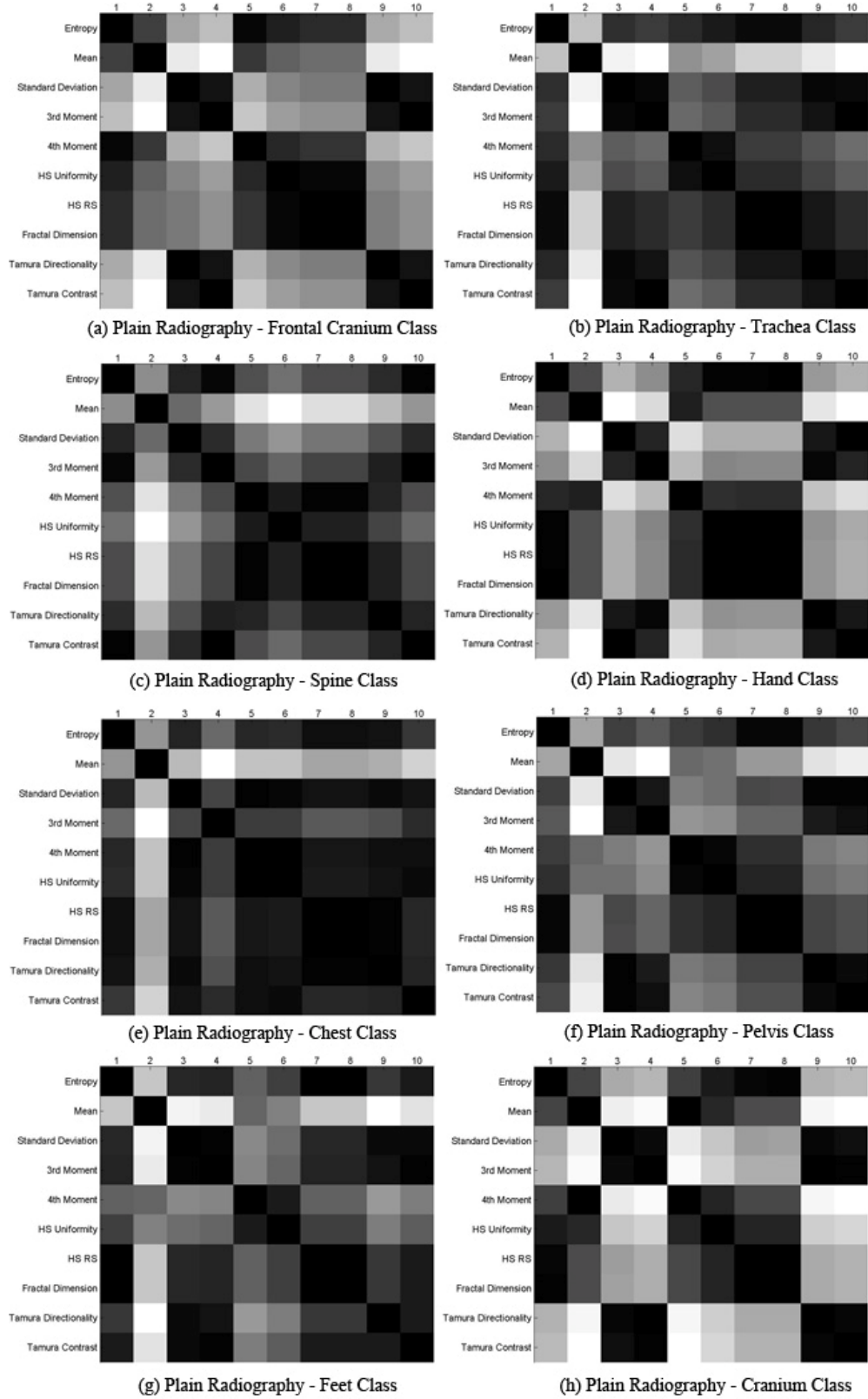


Figure 6.8. Intra-class Correlation Maps for the ImageCLEFMED05 Dataset

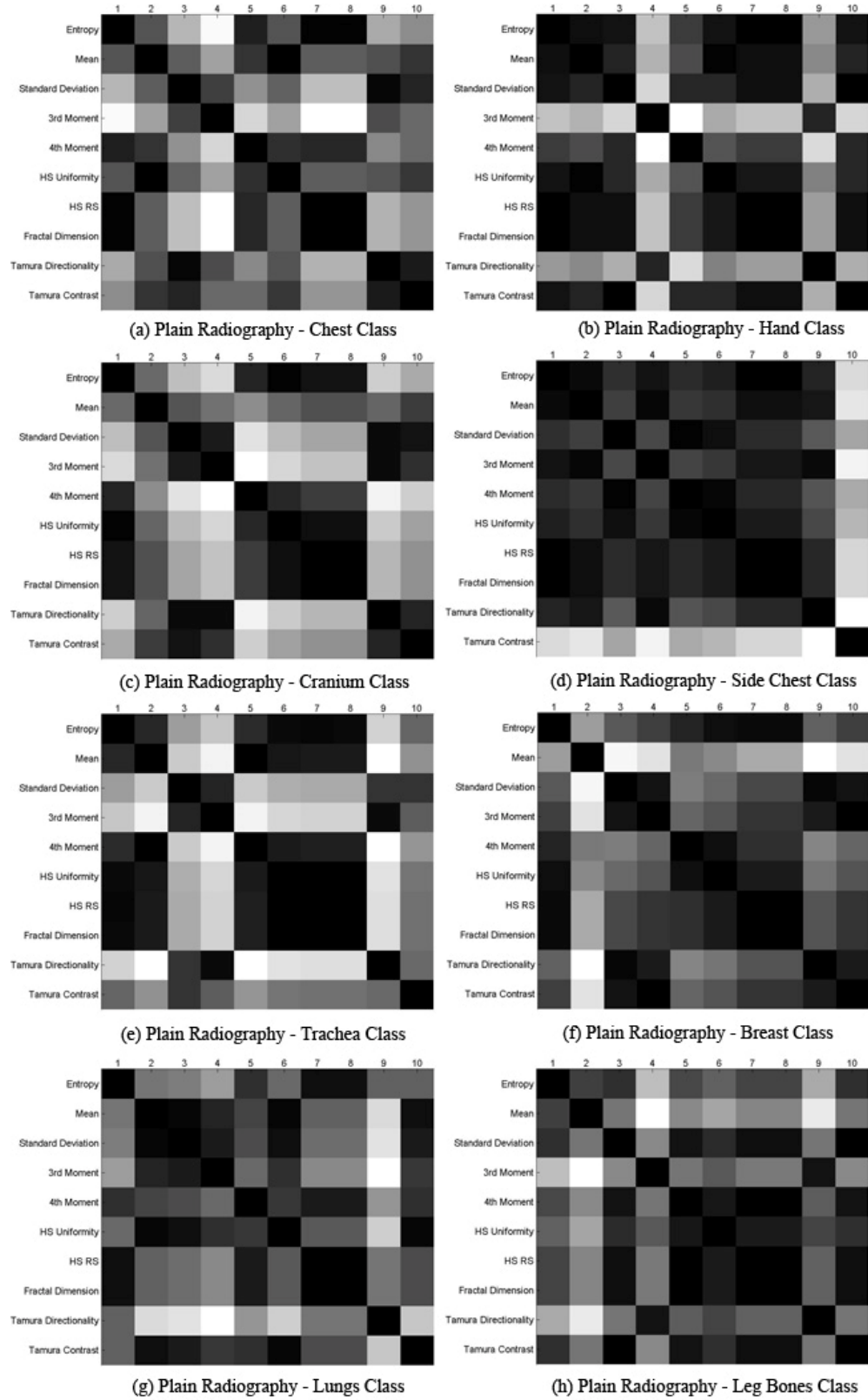


Figure 6.9. Inter-class Correlation Maps for the ImageCLEFMED07 Dataset

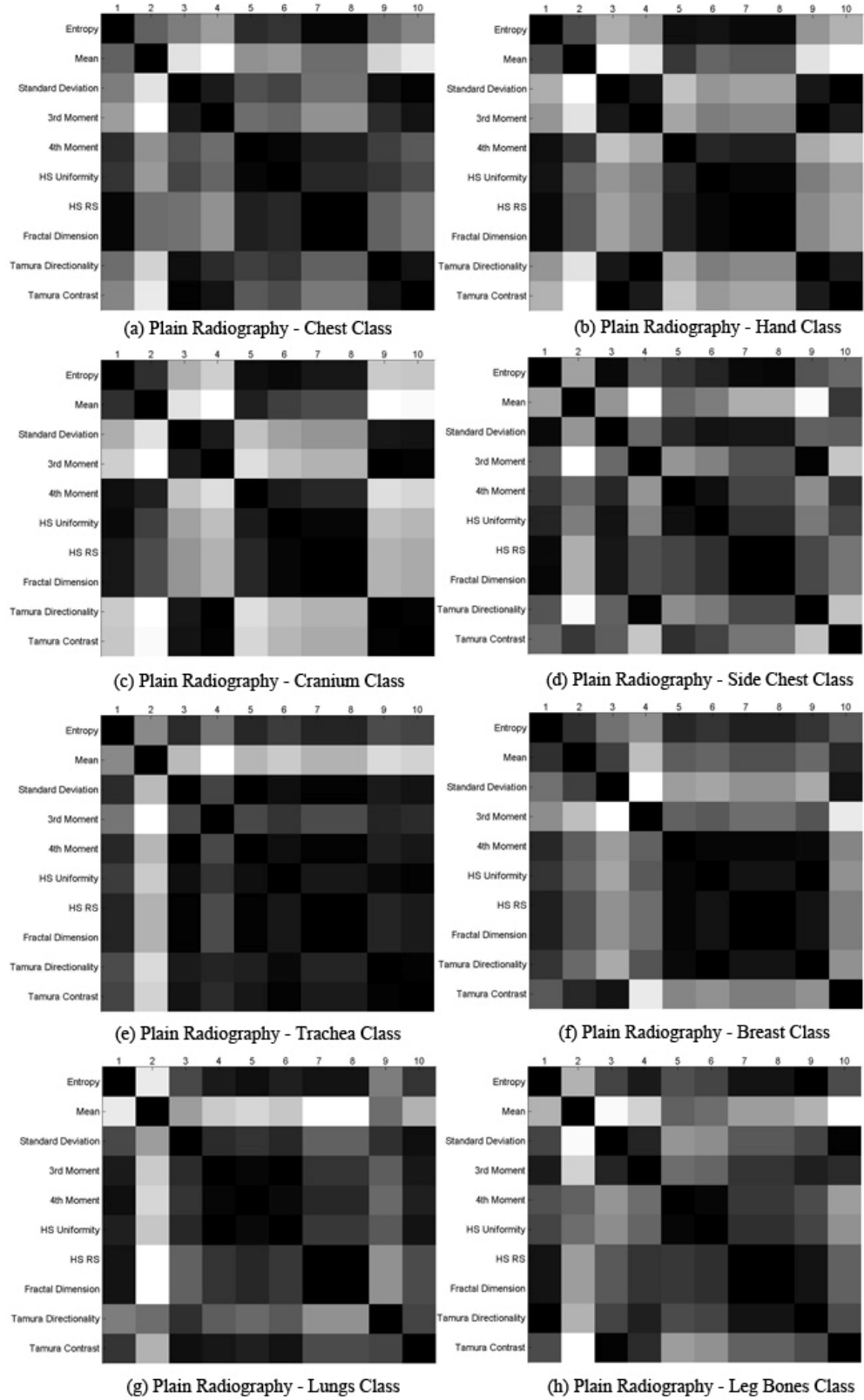


Figure 6.10. Intra-class Correlation Maps for the ImageCLEFMED07 Dataset

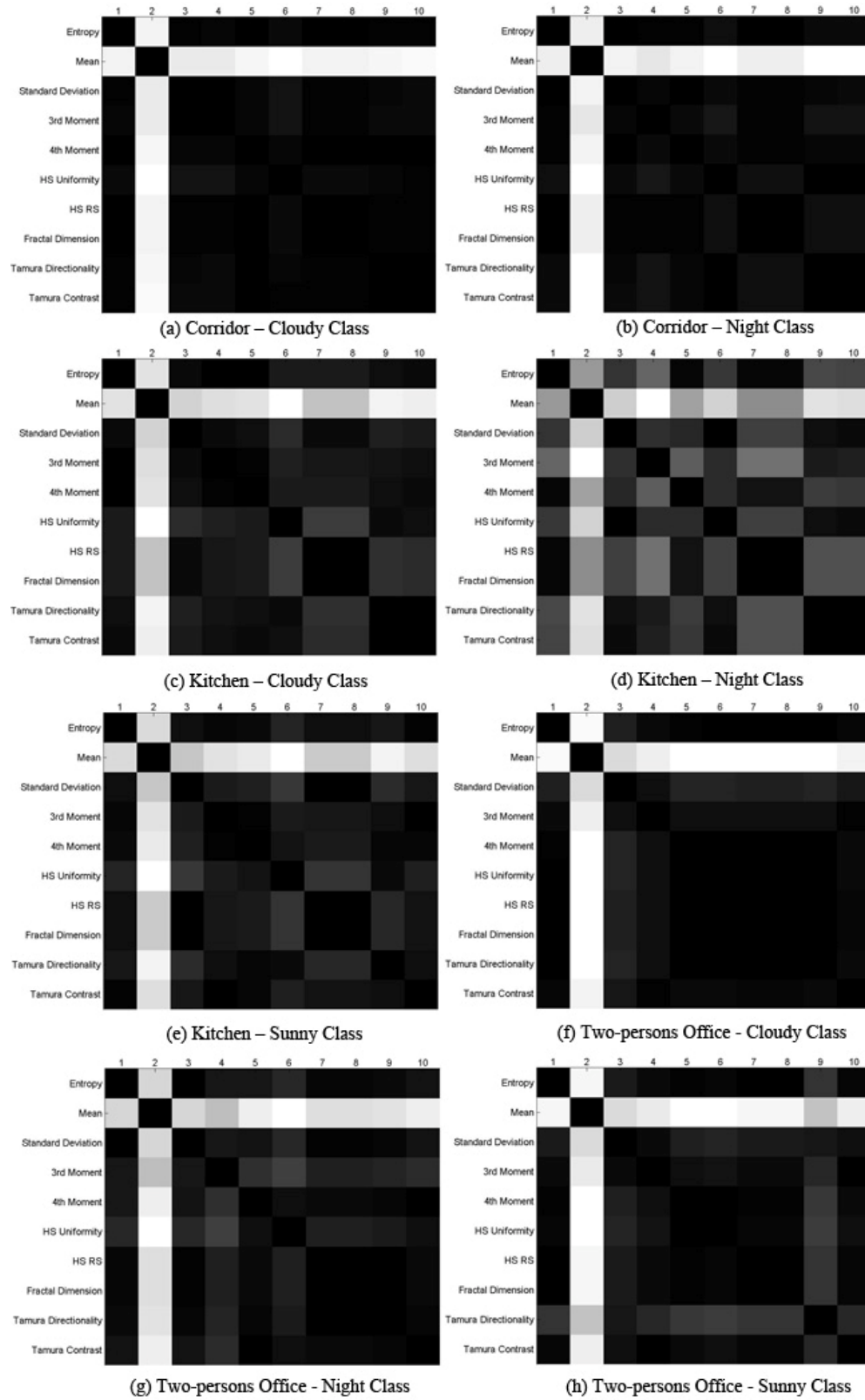


Figure 6.11. Inter-class Correlation Maps for the INDECS Dataset

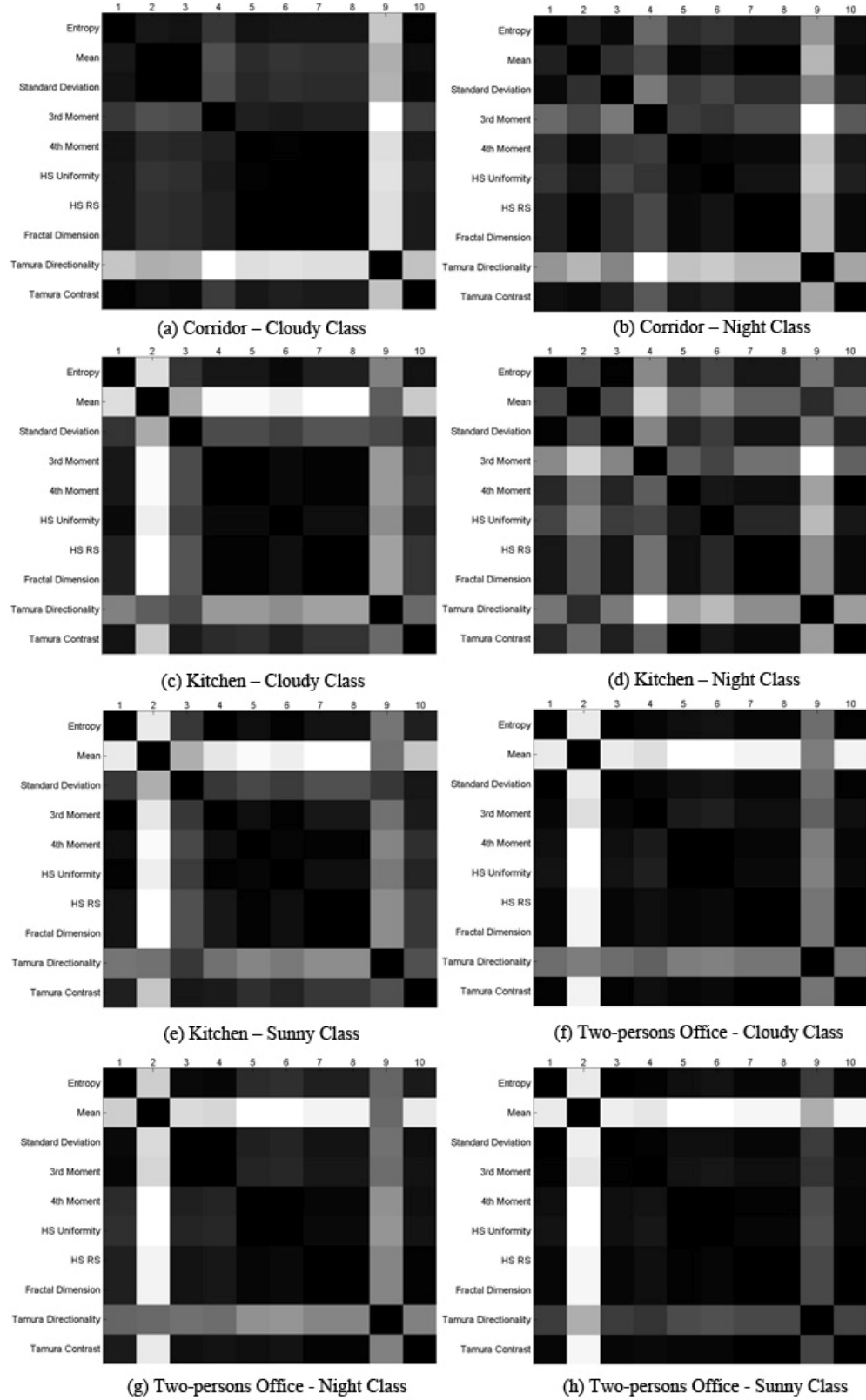


Figure 6.12. Intra-class Correlation Maps for the INDECS Dataset

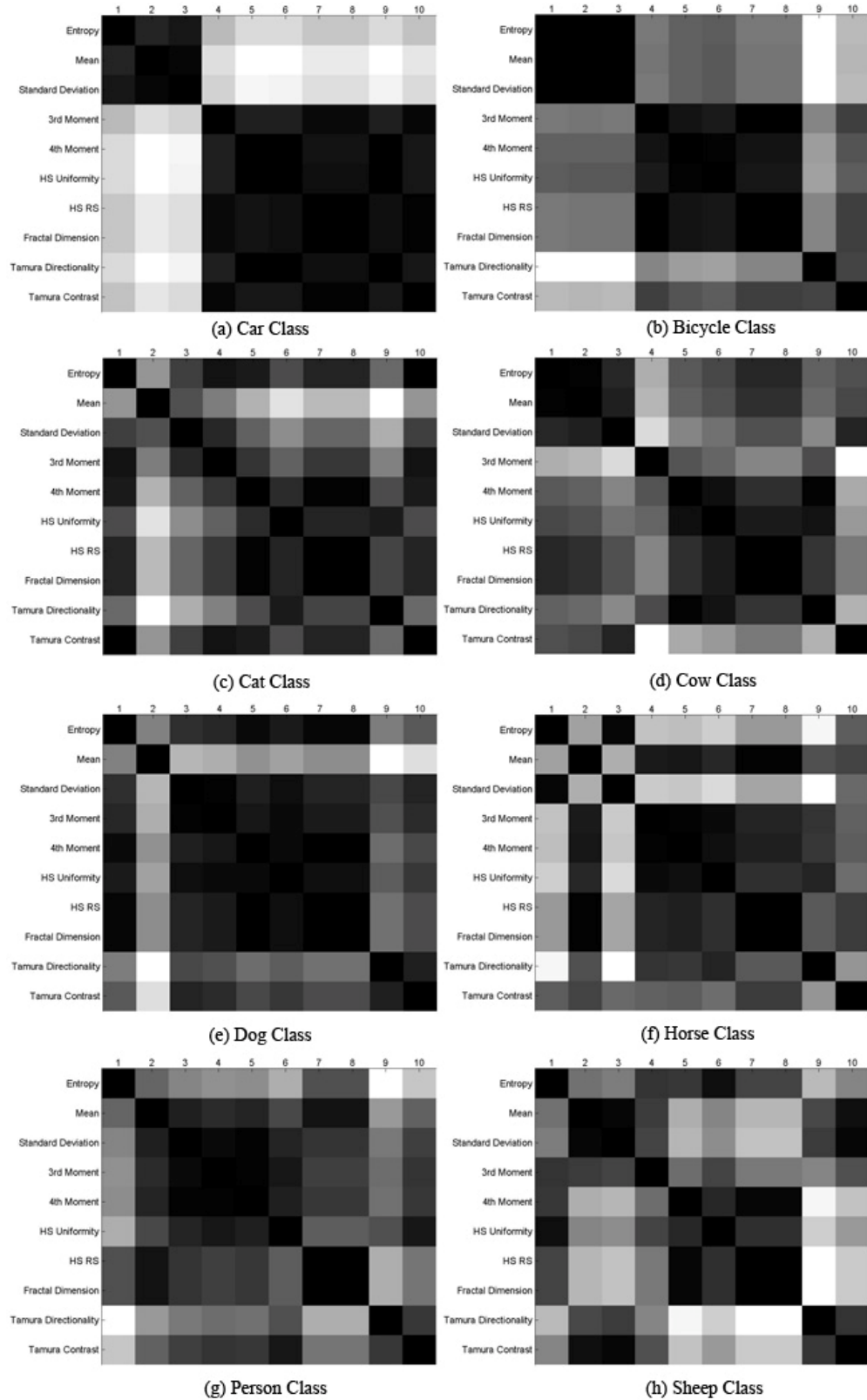


Figure 6.13. Inter-class Correlation Maps for the PASCAL2006 Dataset

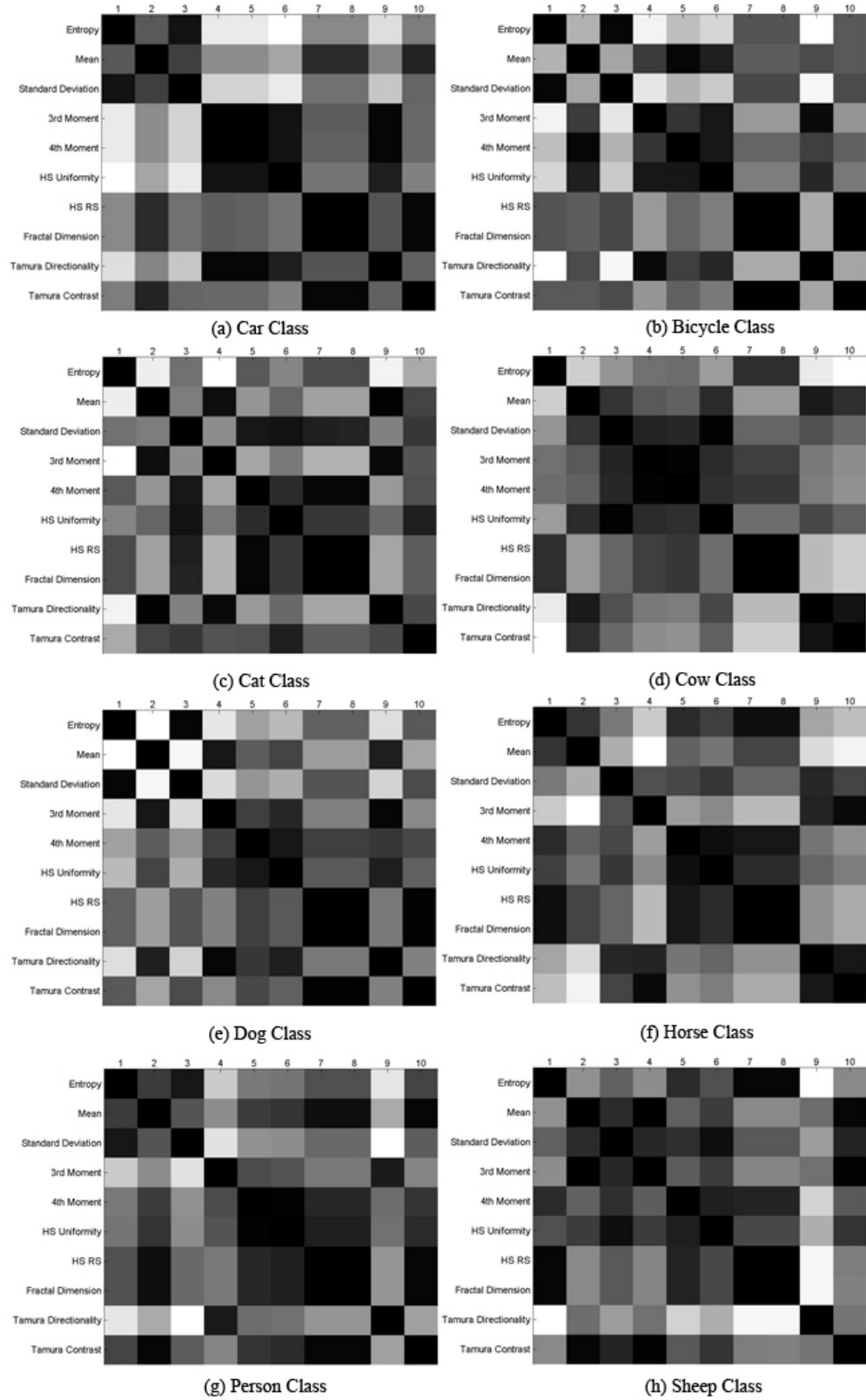


Figure 6.14. Intra-class Correlation Maps for the PASCAL2006 Dataset

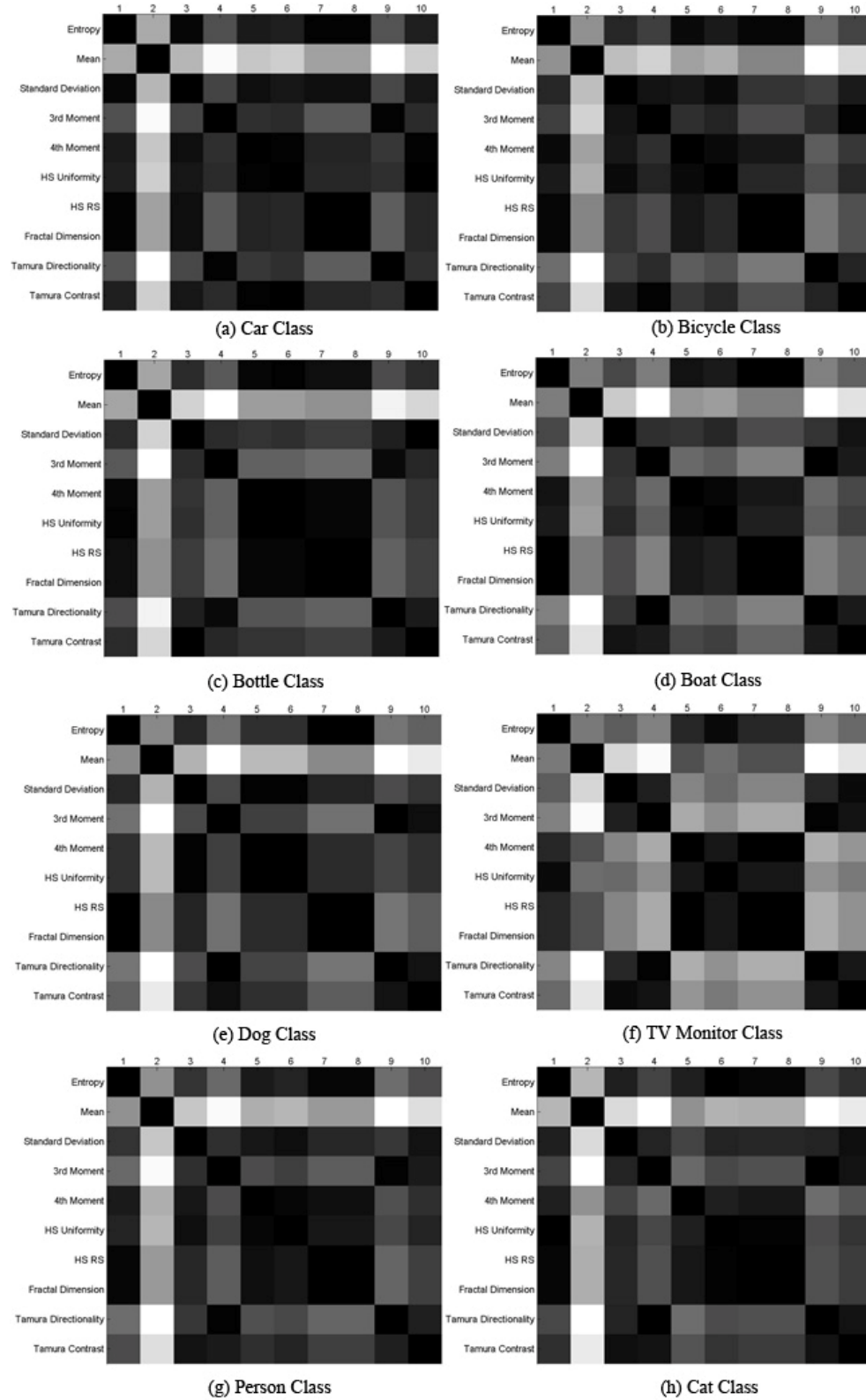


Figure 6.15. Inter-class Correlation Maps for the PASCAL2008 Dataset

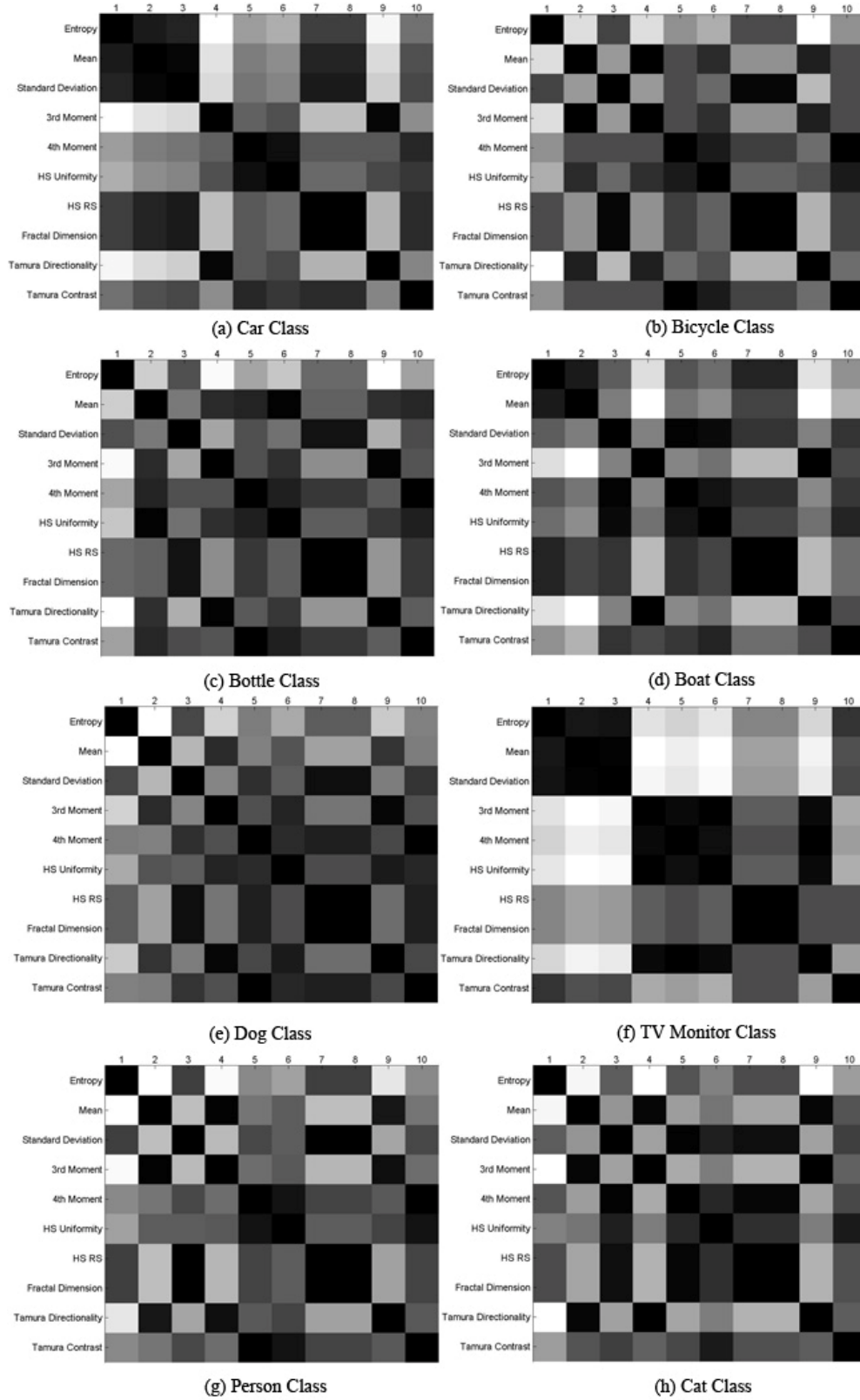


Figure 6.16. Intra-class Correlation Maps for the PASCAL2008 Dataset

CHAPTER 7

DISSIMILARITY EVALUATION MEASURES

Introduction

After investigating supervised and unsupervised attribute evaluation methods in chapter 7. We are now confronted with the problem of determining the most informative dissimilarity measures for our solar images. Having this in mind we experimented with 18 similarity measures that are widely used for clustering, classification and retrieval of images [126, 127, 128, 129, 130] in order to determine which ones would provide a better differentiation of solar activity in our classes. In order to determine which combination of image parameters and similarity measures work best, we investigated over 360 combinations. These experiments were performed to help identify the most (and least) informative and useful combinations for our solar dataset.

Besides qualitatively determining which combinations of dissimilarity measures and image parameters work best via the analysis of dissimilarity matrices, we also performed extensive quantitative analysis based on multidimensional scaling (MDS) to the resulting dissimilarity matrices with classification algorithms. The MDS method for visualization and dimensionality reduction has been widely used by researchers in different areas for image processing and retrieval [125, 131, 132]. By applying MDS to our dissimilarity matrices, we achieved two things: 1) The mechanism for the construction of a 2D or 3D visualization of our image dataset dissimilarities that depicts class separation in a convenient way. 2) Verification of the amount of dimensionality

reduction that we can achieve with our data points mapped into a new artificial dimensional space via the use of several classification algorithms.

To measure the quality of our dissimilarity measure - image parameter combinations, as well as our dimensionality reduction methodology, we set up two different ways of limiting the MDS components. We quantitatively evaluate our work (presented in section 8.15) using comparative analysis, where we compare the two different component selection methods by presenting comparative classification results for three different classifiers. This allowed us to determine how to select our components in order to achieve similar (or even better) classification results than with our original data. In our particular application, with the expected growth of our repository, the dimensionality reduction is very important in terms of allowing us to considerably reduce our storage costs. As mentioned in [750], the results are very specific to the domain of individual solar phenomena. They allow researchers who are working on a particular type of solar event (e.g. flares) to use a combination of image parameters/distance measures that better serve their classification purposes.

Euclidean Distance [133]

This distance is defined as the distance between two points given by the Pythagorean Theorem. This is a special case of the Minkowski metric where $p=2$,

$$D1 = \sqrt{(x_s - x_t)(x_s - x_t)'} \quad (7.1)$$

Where x_s and x_t are vectors of points.

Standardized Euclidean Distance [133]

This distance is defined as the Euclidean distance calculated on standardized data, in this case standardized by the standard deviations.

$$D2 = \sqrt{(x_s - x_t)V^{-1}(x_s - x_t)'} \quad (7.2)$$

Where V is the n -by- n diagonal matrix whose j^{th} diagonal element is $S(j)^2$, where S is the vector of standard deviations, and x_s and x_t are vectors of points.

Mahalanobis Distance [133]

This distance is defined as the Euclidean distance with normalization based on a covariance matrix to make the distance metric scale-invariant.

$$D3 = \sqrt{(x_s - x_t)C^{-1}(x_s - x_t)'} \quad (7.3)$$

Where C is the covariance matrix, and x_s and x_t are vectors of points.

City Block Distance [133]

This distance is also known as Manhattan distance, it represents distance between points in a grid by examining the absolute differences between coordinates of a pair of objects. Special case of the Minkowski metric where $p=1$,

$$D4 = \sum_{j=1}^n |x_{sj} - x_{tj}| \quad (7.4)$$

Where x_s and x_t are vectors of points.

Chebychev Distance [133]

The Chebychev distance measures distance assuming only the most significant dimension is relevant. Special case of the Minkowski metric where $p = \infty$,

$$D5 = \max_j \{ |x_{sj} - x_{tj}| \} \quad (7.5)$$

Where x_s and x_t are vectors of points.

Cosine Distance [134]

This distance measures the dissimilarity between two vectors by determining the cosine of the angle between them,

$$D6 = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \quad (7.6)$$

Where x_s and x_t are vectors of points.

Correlation Distance [134]

The correlation distance measures the dissimilarity of the sample correlation between points as sequences of values

$$D7 = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'} \sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)'}} \quad (7.7)$$

Where $\bar{x}_s = \frac{1}{n} \sum_{j=1}^n x_{sj}$, $\bar{x}_t = \frac{1}{n} \sum_{j=1}^n x_{tj}$ and x_s and x_t are vectors of points.

Spearman Distance [135]

The Sperman distance measures the dissimilarity of the sample's Spearman rank [135] correlation between observations as sequences of values,

$$D8 = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'} \sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \quad (7.8)$$

Where r_{sj} is the rank of x_{sj} taken over $x_{1j}, x_{2j} \dots x_{nj}$, r_s and r_t are the coordinate-wise rank vectors of x_s and x_t , i.e., $r_s = (r_{s1}, r_{s2}, \dots, r_{sm})$ and

$$\bar{r}_s = \frac{1}{m} \sum_{j=1}^m r_{sj} = \frac{(m+1)}{2}, \quad \bar{r}_t = \frac{1}{m} \sum_{j=1}^m r_{tj} = \frac{(m+1)}{2}$$

Hausdorff Distance [15]

We present and use the Hausdorff distance in terms of histograms, since it is widely used for smaller-scale image analysis in different areas [90, 94, 6, 93, 91]. The distance is intuitively defined as the maximum distance of a histogram to the nearest point in the other histogram. In order to represent our feature vector as a histogram, we treated each element of n as a bin ($n = 64$). For example, we convert x_s to the histogram A , the value in each bin A_j (for $j = 1$ to n) is equal to each x_{sj} (for $j = 1$ to n).

$$D9(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} \quad (7.9)$$

Where *sup* represents the *supremum*, *inf* the *infimum*, and $d(a, b)$ represents any distance measure between two points. In our case Euclidean distance, and A and B are histograms

Jensen–Shannon Divergence (JSD) [13]

We present and use the Jensen-Shannon divergence in terms of histograms, based on our literature review of smaller-scale image analysis in different areas [90, 94]. The Jensen-Shannon divergence is also known as total divergence from the average, this measure is a symmetrized and smoothed version of the *Kullback–Leibler divergence*.

$$D10(A, B) = \sum_{j=1}^n A_j \log \frac{2A_j}{A_j + B_j} + B_j \log \frac{2B_j}{B_j + A_j} \quad (7.10)$$

Where A and B are histograms. In order to represent our feature vector as a histogram, we treated each element of n as a bin ($n = 64$). For example, we convert x_s to the histogram A , the value in each bin A_j (for $j = 1$ to n) is equal to each x_{sj} (for $j = 1$ to n).

χ^2 Distance [136]

We present and use the χ^2 distance in terms of histograms, based on our literature review of smaller-scale image analysis in different areas [6, 93]. The χ^2 distance measures the likeliness of one histogram being drawn from another one,

$$D11(A, B) = \sum_{j=1}^n \frac{A_j - B_j}{A_j + B_j} \quad (7.11)$$

Where A and B are histograms. In order to represent our feature vector as a histogram, we treated each element of n as a bin ($n = 64$). For example, we convert x_s to the histogram A , the value in each bin A_j (for $j = 1$ to n) is equal to each x_{sj} (for $j = 1$ to n).

Kullback–Leibler Divergence (KLD) [14]

We present and use the Kullback–Leibler divergence (KLD) in terms of histograms, based on our literature review of smaller-scale image analysis in different areas [6, 93, 91]. The KLD divergence measures the difference between two histograms A and B . Often intuited as a distance metric, the *KL Divergence* is not a true metric since it is not symmetric, the *KL Divergence* from A to B is not necessarily the same as the *KL divergence* from B to A

$$D12(A, B) = \sum_{j=1}^n A_j \log \frac{A_j}{B_j} \quad (7.12)$$

$$D13(B, A) = \sum_{j=1}^n B_j \log \frac{B_j}{A_j} \quad (7.13)$$

Where A and B are histograms. Since this is the only non-symmetric measure we use for this work. We treat it as a directed measure and consider A to B and B to A as two different distances. In order to represent our feature vector as a histogram, we treated each element of n as a bin ($n = 64$). For example, we convert x_s to the histogram A , the value in each bin A_j (for $j = 1$ to n) is equal to each x_{sj} (for $j = 1$ to n).

Fractional Minkowski [133]

For the fractional Minkowski metrics (D14 to 18), we have selected five different factional values for the Minkowski metric ($p = 0.25, 0.50, 0.80, 0.90$, and 0.95). Previous

research [129, 130] has shown that these fractional metrics out-perform the traditional *Euclidean* ($p=2$) and *city block* ($p=1$) distances for several artificial datasets.

$$D_{14-18} = \left(\sum_{j=1}^n |x_{sj} - x_{tj}|^p \right)^{1/p} \quad (7.14)$$

Where x_s and x_t are vectors of points.

Evaluation Results for the Solar Dataset

All experiments were performed using Matlab© R2009b. For the exponential curve fitting we used the Ezyfit Toolbox [137]. As we mentioned before, in order to properly evaluate each of the extracted image parameters (table 4.1) we decided to treat them individually. We created a 64-element vector (from our 8 x 8 grid segmentation) per image parameter, for each image. In order to use these vectors as histograms correctly when calculating the *KLD* and *JSD* measures we need to make sure the sum of the bins adds to one. To achieve this, we normalized every single parameter per image in the following way:

$$Normalize(A) = \frac{A_m}{\sum_{m=1}^n A_m} \text{ for } m = 1 \text{ to } n \quad (7.15)$$

where $n=64$, since we have a total of 64 elements.

For each bin in the histograms, this allows us to scale our histograms and preserve their shape. For bins equaling zero, we add a very small quantity (0.1×10^{-8}) in order to avoid divisions by zero on the *KLD* measure.

After all our data had been normalized in this manner, we calculated the pair-wise distance between the histograms using Matlab© `pdist` function. As this function is highly optimized for performance, the computation time for our first eight (and last five) measures was very short. The *Hausdorff*, *KLD*, *JSD* and χ^2 distances where implemented from scratch and yielded a higher computational expense due to their nature. In total we produced 180 dissimilarity matrices (18 measures, counting *KLD A-B* and *B-A* separately, times a total of 10 different image parameters). All these dissimilarity matrices are symmetric, real and positive valued, and have zero-valued diagonals. Thus they fit the classical MDS requirements. In table 7.1 we outline the ordering utilized for the classification accuracy figures presented in this chapter.

Table 7.1. Dissimilarity measure labels and ranges used in our classification accuracy figures

Label	Distance	Range	Used in Figures
D1	Euclidean	11-20	7.8, 7.10, 7.11 and 7.12
D2	Standardized Euclidean	21-30	7.8 and 7.10
D3	Mahalanobis	31-40	7.8 and 7.10
D4	City Block	41-50	7.8, 7.10, 7.11 and 7.12
D5	Chebychev	51-60	7.8, 7.10, 7.11 and 7.12
D6	Cosine	61-70	7.8 and 7.10
D7	Correlation	71-80	7.8 and 7.10
D8	Spearman	81-90	7.8 and 7.10
D9	Hausdorff	91-100	7.8 and 7.10
D10	Jensen-Shannon divergence (JSD)	101-110	7.8 and 7.10
D11	χ^2	111-120	7.8 and 7.10
D12	Kullback-Leibler divergence (KLD) A-B	121-130	7.8 and 7.10
D13	Kullback-Leibler divergence (KLD) B-A	131-140	7.8 and 7.10
D14	Fractional p=0.25	141-150	7.11 and 7.12
D15	Fractional p=0.50	151-160	7.11 and 7.12
D16	Fractional p=0.80	161-170	7.11 and 7.12
D17	Fractional p=0.90	171-180	7.11 and 7.12
D18	Fractional p=0.95	181-190	7.11 and 7.12

Table 7.2 presents the ordering used for the scaled image plots of the dissimilarity matrices (**using scaled image plots**) in figures 7.2, 7.5 and 7.6. It also lists the colors used for the MDS plots presented in figures 7.2, 7.3, 7.5 and 7.6.

Table 7.2. Scaled image plot ranges and colors corresponding to the classes of the solar dataset

Event Name	Scaled Image Plot Range	MDS Color
Active Region	1-200	Red
Coronal Jet	201-400	Green
Emerging Flux	401-600	Blue
Filament	601-800	Yellow
Filament Activation	801-1000	Magenta
Filament Eruption	1001-1200	Gray
Flare	1201-1400	Orange
Oscillation	1401-1600	Black

Dissimilarity Matrix Calculations

Figure 7.1 presents the average time (in minutes) that is required to calculate one 1,600 by 1,600 dissimilarity matrix for each of the 18 dissimilarity measures. Averages are reported based on 10 calculations, as separate matrices were calculated for each of the 10 image parameters (table 4.1).

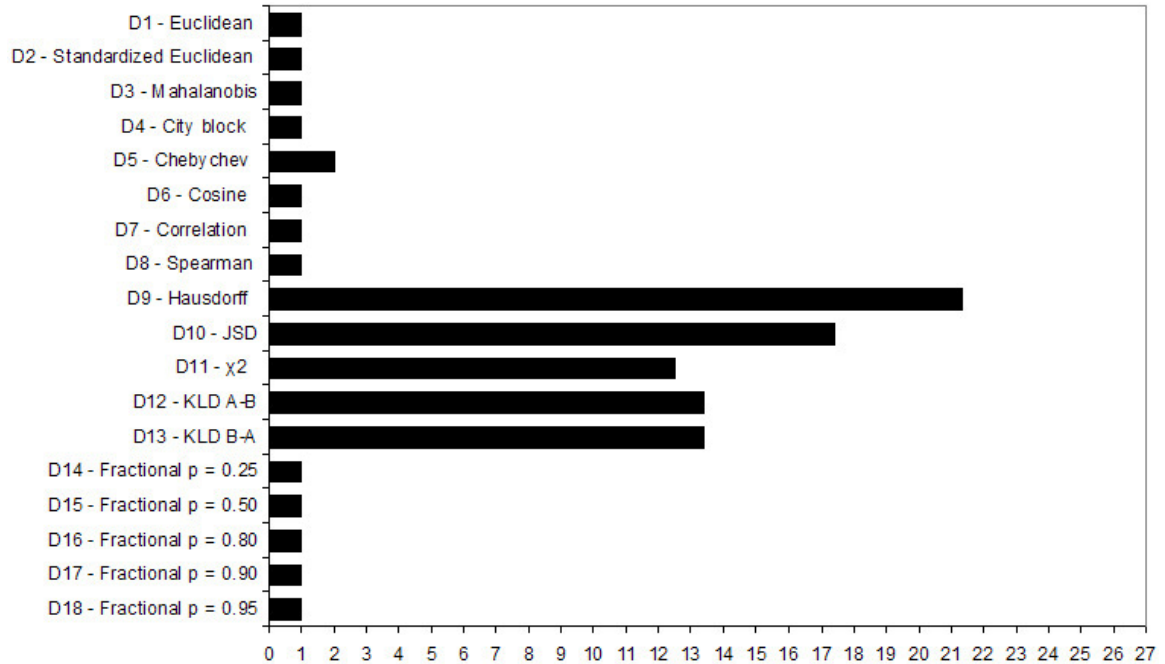


Figure 7.1. Average time in minutes required to calculate one 1,600 by 1,600 dissimilarity matrix

Plots of these dissimilarity matrices help us to qualitatively identify which image parameters and measures provide informative differentiation for our images between the eight different classes of our solar dataset. In this paper we show three of the most interesting dissimilarity measure – image parameter combinations we generated (good and bad). However, all 180 combinations are found at [122]. Here the classes of our benchmark are separated on the axes, after 200 units (images) the next class starts. The classes are ordered in the same way as in table 7.2.

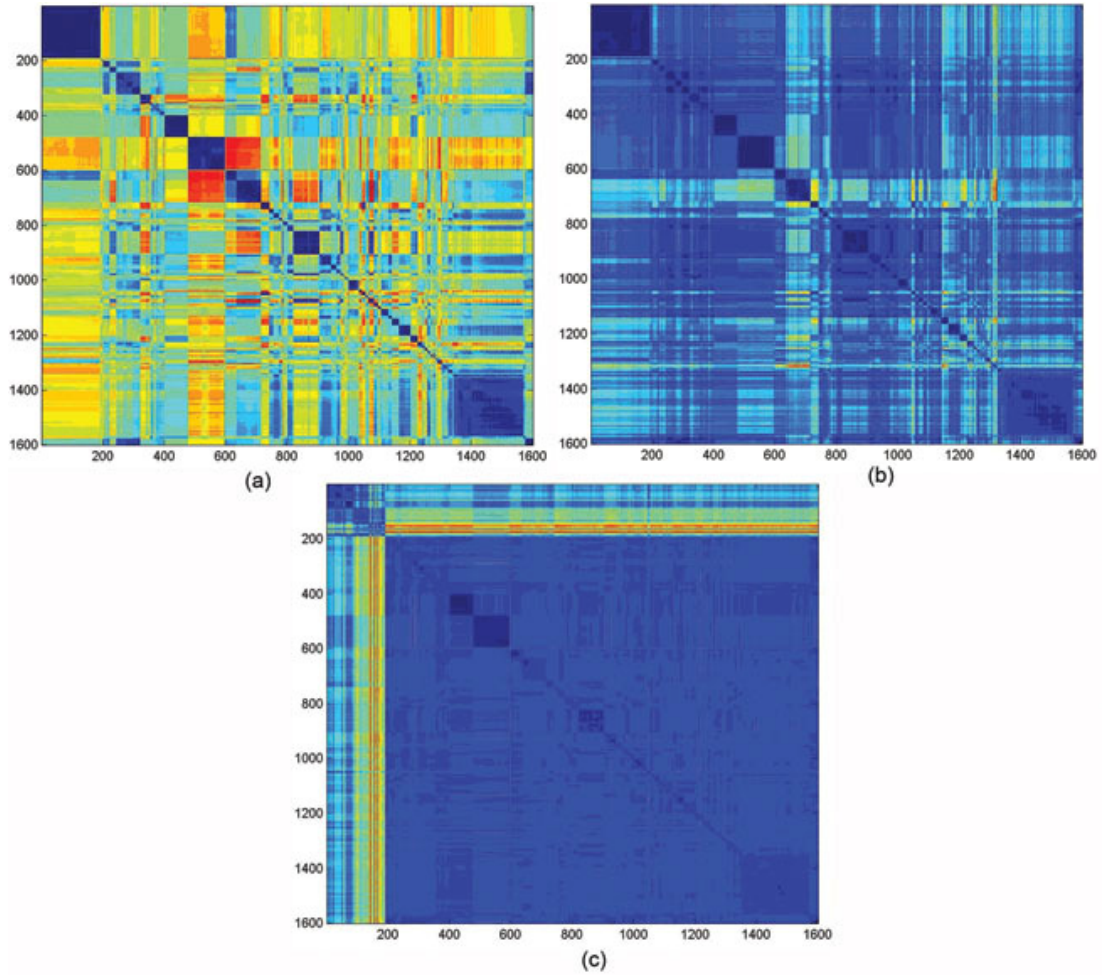


Figure 7.2: Scaled image plot of dissimilarity matrix for (a) *Correlation* (D7) measure with image parameter *mean* (P2), (b) *JSD* (D10) measure with image parameter *mean* (P2), (c) *Chebychev* (D5) measure with image parameter *relative smoothness* (P7)

As we can see from figure 7.2(a), this combination of similarity measure D7 (*correlation*) and image parameter P2 (*mean*) produces a good separation between all classes, hence the very diverse coloring throughout the figure. Blue means low dissimilarity, and red means high dissimilarity. Figure 7.2 (b) shows that the D10 (*JSD*) measure produces an entirely different dissimilarity matrix for the same image parameter P2 (*mean*), which highlights different dissimilarities than the correlation measure (shown in figure 7.2 (a)). Figure 7.2(c) is a clear example of a combination of a dissimilarity

measure D5 (*Chebyshev*) and an image parameter P7 (*relative smoothness*) that highlights dissimilarities within only one class of the benchmark, but recognizes that everything else is highly similar for the rest of the classes. This validates our idea of testing every pair of dissimilarity measures and image parameters individually, since there are combinations that will allow us to notice different relationships between the classes of solar images.

Transformation of Dissimilarity Matrices via Multidimensional Scaling (MDS)

After generating 180 dissimilarity matrices, we performed classical multidimensional scaling using Matlab's `cmdscale` function. MDS has been widely utilized in image retrieval research to reduce dimensionality [131, 132], and to aid in the visual depiction of image similarity in a convenient two and three dimensional plot [125]. However, these works present results on a considerably smaller number of images and use a considerably smaller number of dimensions. The colors used in the MDS plots are presented in table 7.3. The most commonly used MDS plots involve using the first two or three components of the calculated coordinate matrix. In figure 7.3 we show both 2D and 3D plots of these components.

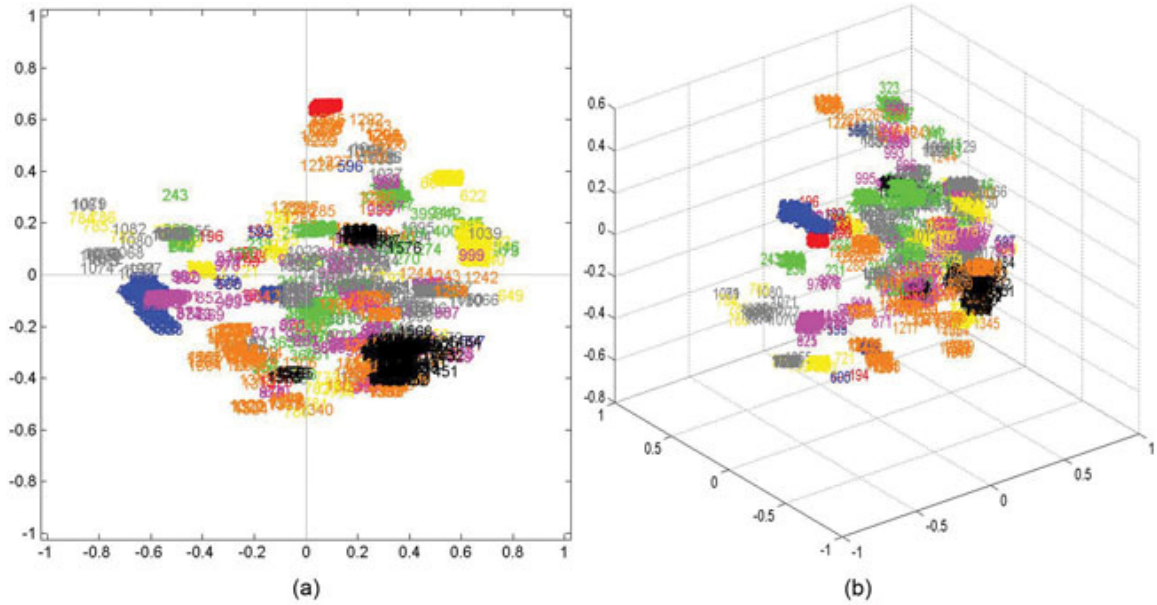


Figure 7.3. MDS map for the *correlation* measure D7 with image parameter *mean* (P2) a) top 2 and b) top 3 components

As we expected, in figure 7.3 we cannot easily identify a clear separation between our eight different classes on 2 or 3 dimensions. However, selecting the correct combination might yield interesting 3D maps for certain dissimilarity measure –image parameter pairs. In figure 7.4 we show the 3D MDS component plots for the dissimilarity matrices of figure 7.2. Here we show that, while for a) and b) the maps do not really highlight any clusters, in c) we have a clear cluster for class *Active Region* (color red from table 7.2). This is exactly what is indicated in figure 7.2 part c). These 3D MDS components plots sometimes show very clear clusterings. Other times they can be used to aid in the interpretation of the scaled image plots of the dissimilarity matrices (we will talk about this when we discuss the fractional Minkowski metrics results)

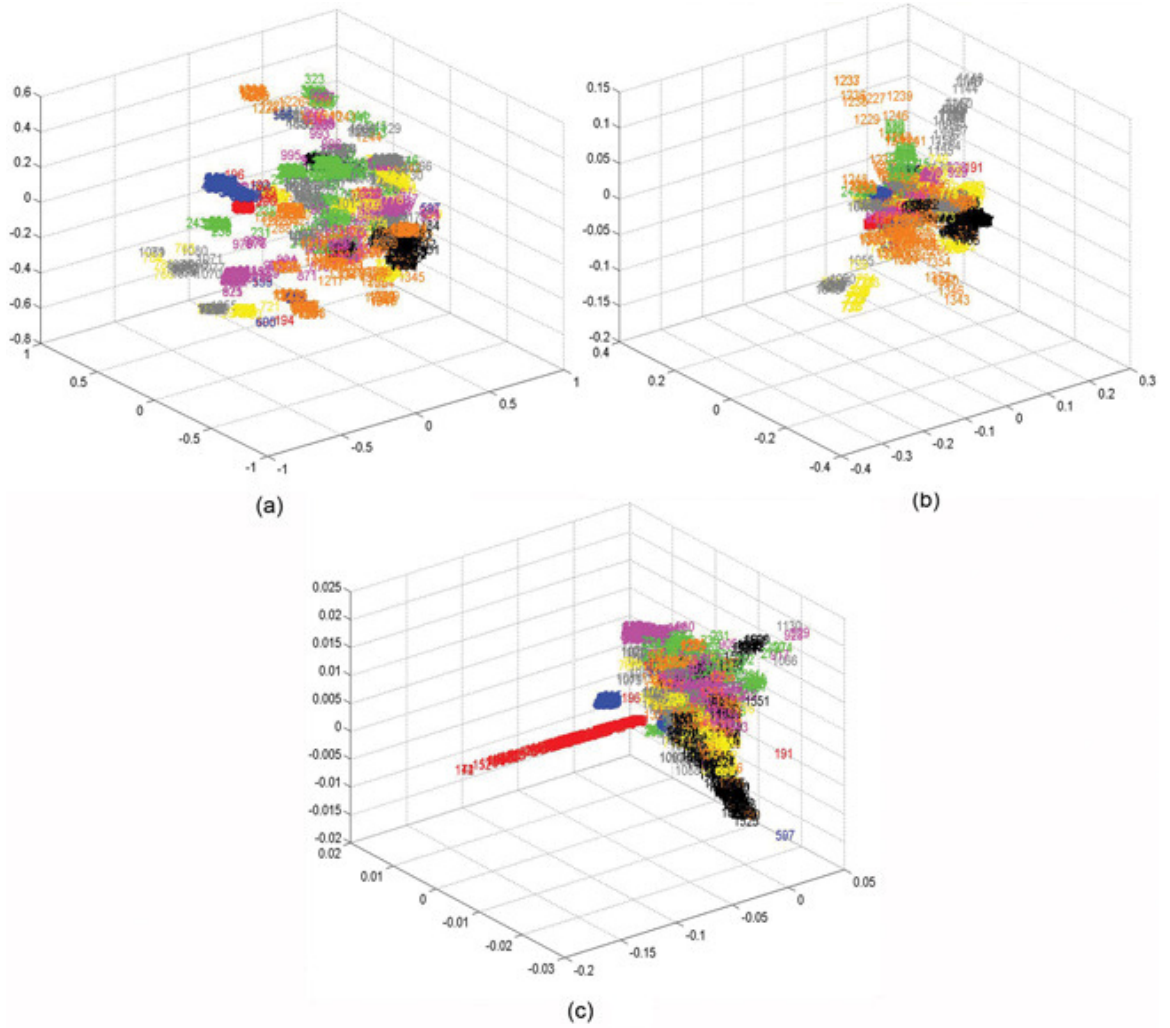


Figure 7.4. 3D MDS components plot for (a) *Correlation* (D7) measure with image parameter *mean* (P2), (b) *JSD* (D10) measure with image parameter *mean* (P2), (c) *Chebychev* (D5) measure with image parameter *relative smoothness* (P6).

Fractional Minkowski Metrics

Due to the fact that Minkowski-based fractional metrics have provided very interesting results in other domains [129, 130], and are supposed to be mathematically proven to be better than other more traditional non-fractional Minkowski metrics [129], we wanted to investigate their behavior on our solar dataset. In this section, using figures

7.5 and 7.6, we present a comparison between the scaled image plot and the 3D MDS components plot for two different fractional metrics of the same image parameter.

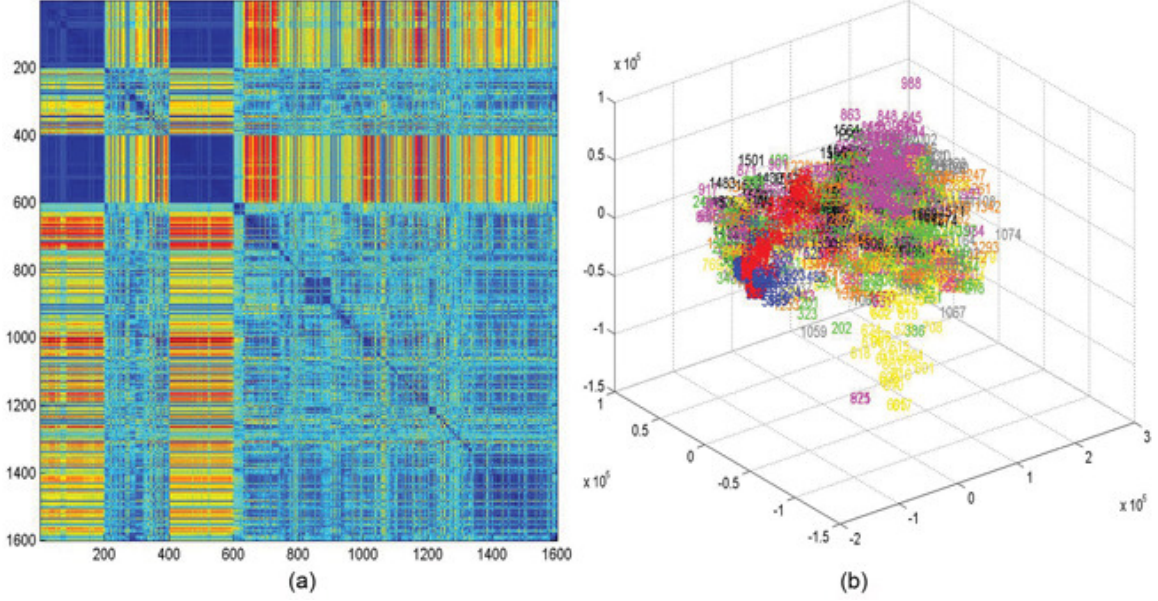


Figure 7.5. Plots for *Fractional Minkowski* metric with $p=0.25$ (D14) and the *fractal dimension* image parameter (P8): (a) Scaled Image Plot, (b) 3D MDS components plot.

From figure 7.5, we can observe a relatively clean separation of two classes (*Active Region* (axis-range: 1-200) and *Emerging Flux* (axis-range: 401-600)), when it comes to the scaled image plot of the dissimilarity matrix (a). However, in terms of the 3D MDS components plot (b) the clusters corresponding to *Active Region* (red color) and *Emerging Flux* (blue color) seem a bit fuzzy and mixed in with the rest of the classes. According to our literature review [129, 130] these fractional metrics tend to improve when p approaches one. In figure 7.6 we try to verify these claims for our own domain-specific dataset.

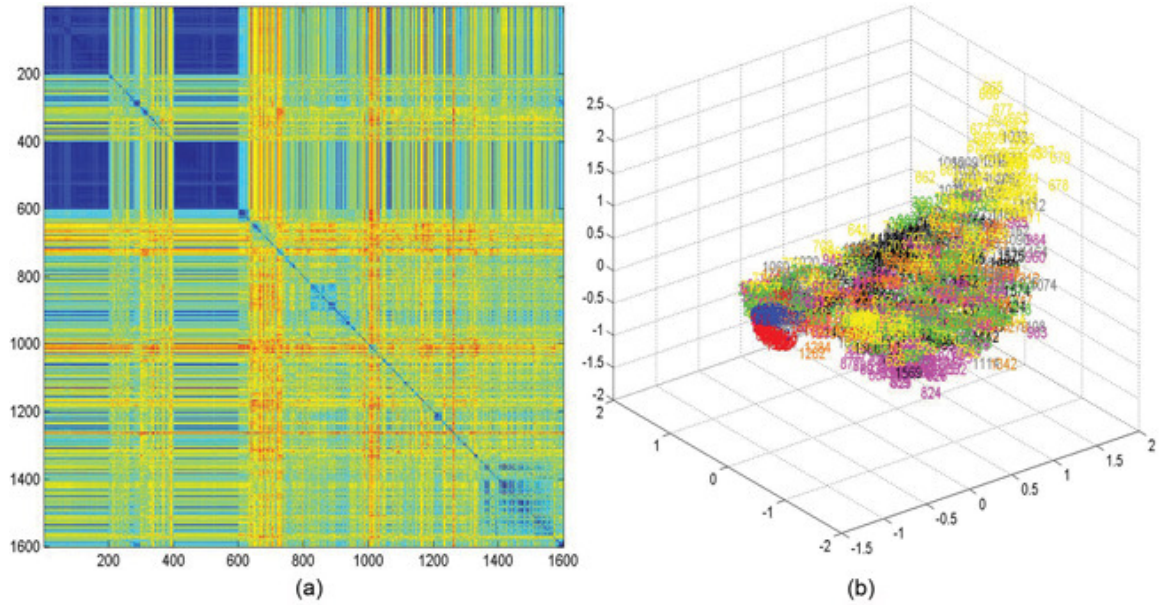


Figure 7.6. Plots for *Fractional Minkowski* metric with $p=0.95$ (D18) and the *fractal dimension* image parameter (P7): (a) Scaled Image Plot, (b) 3D MDS components plot

As the literature mentions [129, 130], once p is closer to 1 we see that on both the scaled image plot of the dissimilarity matrix (a) and the 3D MDS components plot (b), the classes *Active Region* and *Emerging Flux* are more separable and hence have better clustering. It is also very interesting that the rest of the classes tend to get ‘dragged away’ from these two classes, making the *Active Region* and *Emerging Flux* classes even more distinguishable. However, this increases the separation of the other classes. This only shows that each combination of dissimilarity measure and image parameter is very unique, and that their behaviors need to be analyzed individually to get the most out of this type of analysis, a daunting task indeed considering the number of plots to analyze is equal to the number of image parameters times the number dissimilarity measures being analyzed.

Component Thresholding

As we mentioned before, MDS is also used for dimensionality reduction. We analyzed the magnitudes (importance) of the MDS components in order to determine how many components we need in order to maintain a good representation of our data in the new dimensional space, and how many components we can discard in order to reduce our storage costs.

In order to determine the aforementioned number of components we plotted the magnitude of each component. Since the MDS coordinate matrix output is ordered by importance, the magnitudes should be decreasing as the number of components increases. After empirical analysis of the magnitude of the resulting MDS components, we observed that after 10 components the decrease of their magnitudes moderates (in most of the cases), therefore, as a first approach, we decided to take a somewhat naïve approach and applied a threshold of 10 components per dissimilarity measure - image parameter combination.

In our second approach we utilized exponential curve fitting [138] to find a function that models this behavior. Our intent in this phase was to locate a threshold for the number of necessary components. We utilize a 135 degree angle of the tangent line to this function to determine the threshold and discard the components whose magnitudes are not providing significant improvement over the previous ones.

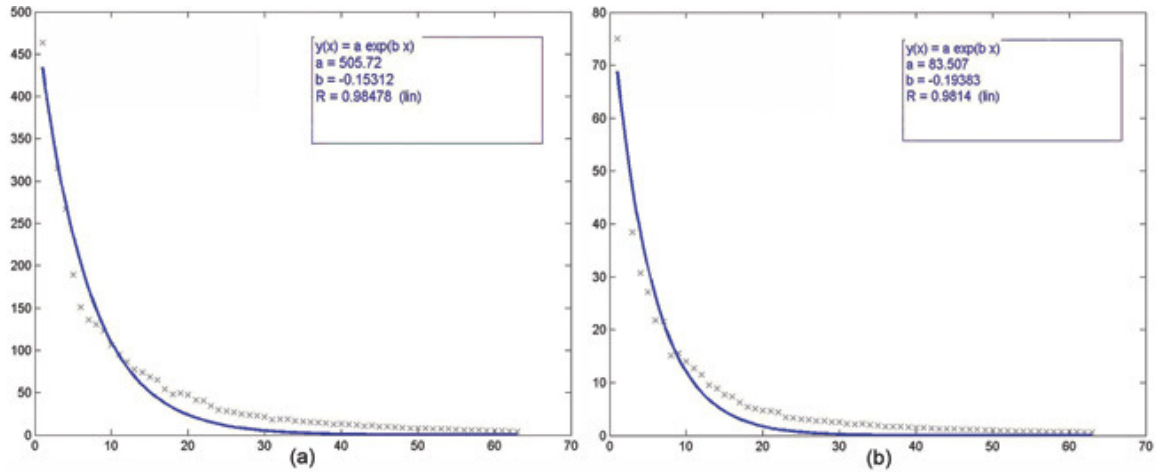


Figure 7.7. Exponential curve fitting for: (a) *correlation* measure (D7) with image parameter mean (P2), (b) JSD measure (D10) with image parameter mean (P2). The x-axis indicates the number of components (1-63) and the y-axis indicates the magnitude of these components.

As can be seen from figure 7.7, the magnitudes of the components decrease up to a certain point. After this point the change is very minimal and thus not too important for the new dimensional space.

Based on these curve fitting results and the threshold output, we determined a specific number of components per combination of dissimilarity measure - image parameter. We can now determine how well this reduced dimensional space performs in our classification tasks in section 7.15.5.

Quantitative Evaluation via Comparative Analysis of Classifiers

We have described how we applied the dissimilarity measures to our image parameters and produced dissimilarity matrices. We also analyzed how MDS transformed these dissimilarity matrices into a different dimensional space, one that will require,

hopefully, fewer dimensions in order to distinguish different types of solar phenomena. We now describe the classification experiments we performed on the naïve 10-component threshold and the resulting tangent thresholded components, and compared them to our original data. All classification experiments were run using 10 fold cross-validation. We ran a total of 360 different datasets through the three classifiers described in chapter 5. In the following figures we present the overall results of these classification experiments and offer a more detailed explanation of the most interesting results.

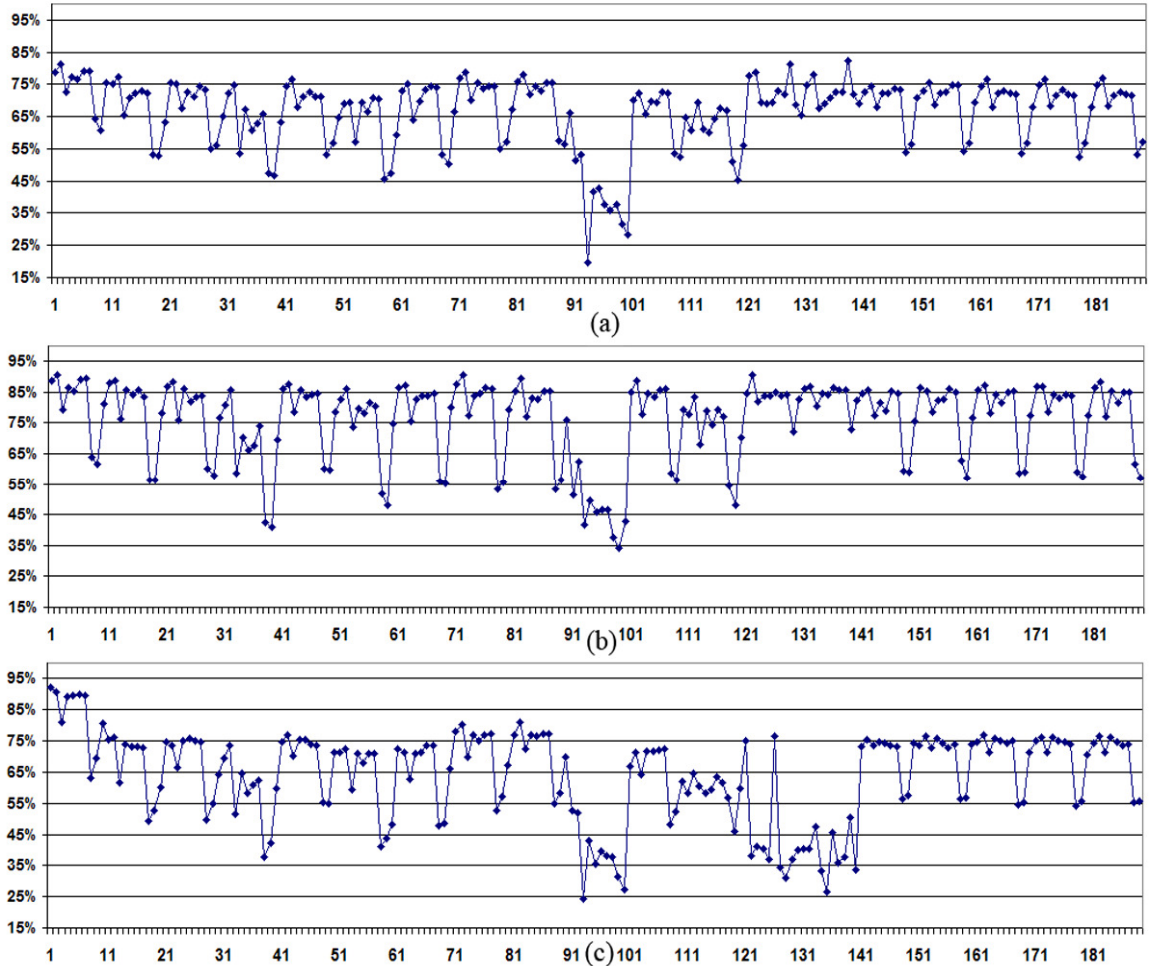


Figure 7.8. Percentage of correctly classified instances for the 10 component threshold: a) for the Naïve Bayes classifier, b) for the decision-tree (C4.5) classifier, and c) for the SVM classifier, for the original data (1-10) and our 180 experiments. Experiments using 10-fold cross-validation

Figure 7.8 shows the classification accuracy of our selected classifiers on our 10 component per measure/image parameter combination. The first 10 columns indicate our original normalized dataset values with no measure or dimensionality reduction applied to them. The rest of the columns (11 to 190) indicate our measure/image parameter combinations in the order they are presented in table 7.1. For instance, element 11 indicates dissimilarity measure (D1) (i.e. *Euclidean distance*) from applied on parameter P1 (i.e. *Entropy*) from table 4.1. Experiment 12 indicates dissimilarity measure (D1) applied on parameter P2, and so on (e.g. experiments 29 and 39, feature dissimilarity measures (D2) and (D3) respectively applied to image parameter P9).

As is shown in figure 7.8(a) and 7.8(b), our 10-components-only approach produces very similar classification results to our original data for most combinations of measure and image parameters. We also notice that the worst performing measure/image parameter combination is presented in columns 91 to 100, corresponding to the *Hausdorff* dissimilarity measure (D9).

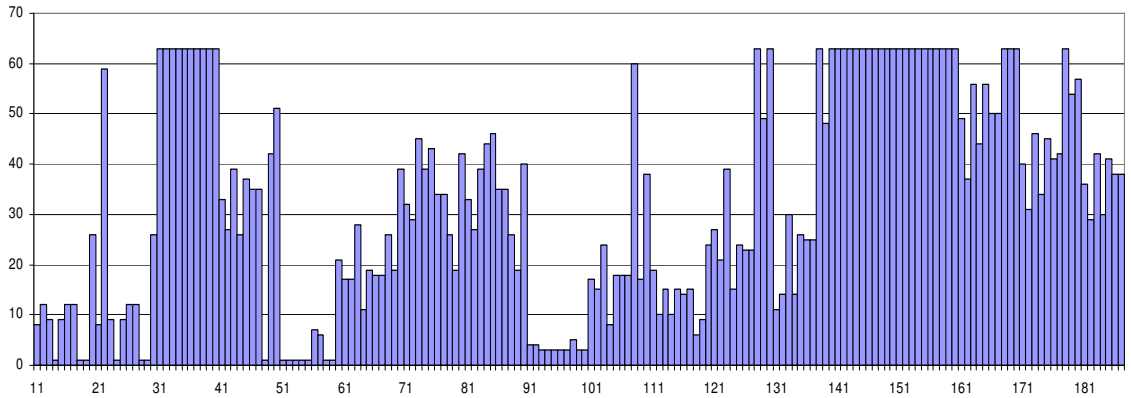


Figure 7.9. Number of components to use indicated by the tangent thresholding method.

In figure 7.9 we show the resulting number of components to be used based on the tangent thresholding separated in two rows. The columns represent the 190 different image parameter/measure combinations (with the omission of the first 10, which are the original dataset). In this figure, a high number of components indicate that the components do not seem to steadily decrease in variance.

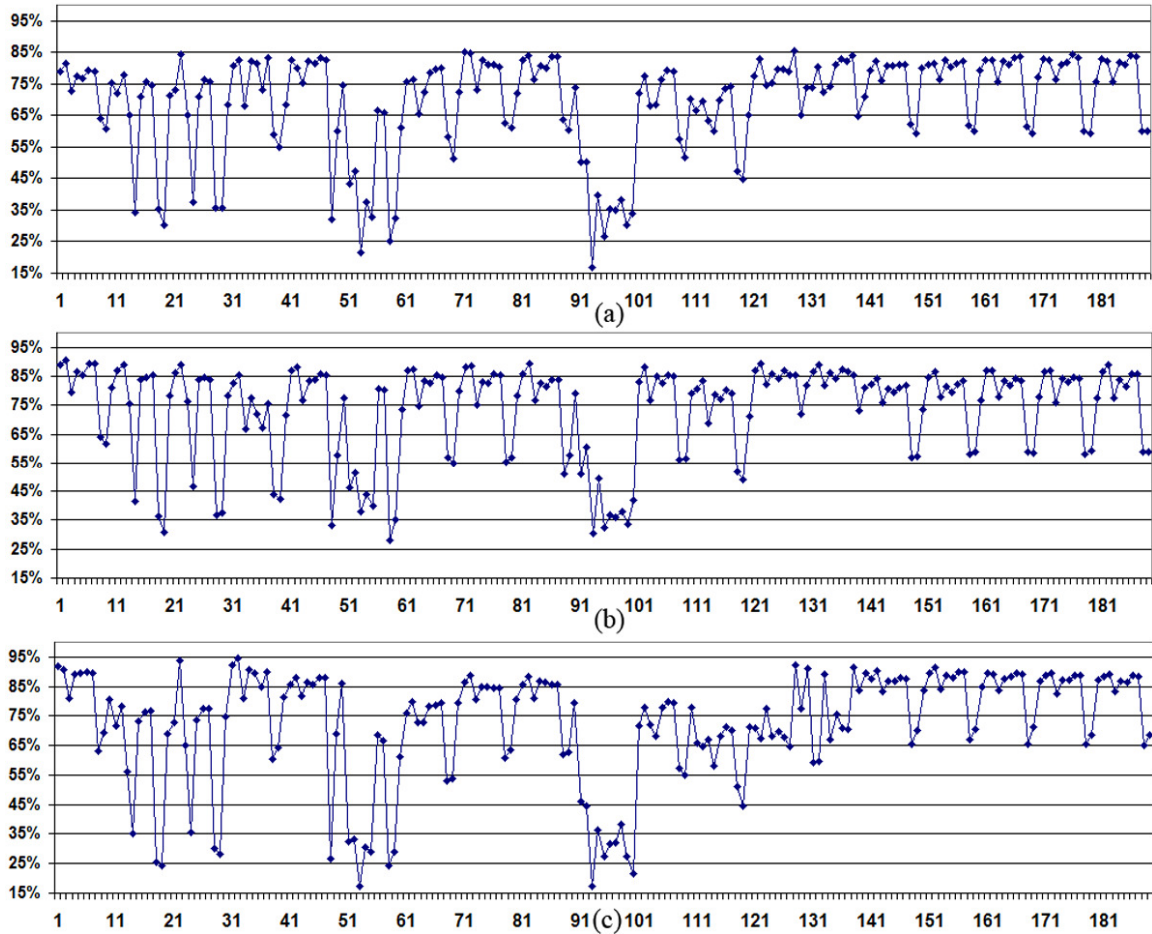


Figure 7.10. Percentage of correctly classified instances for the tangent-based component threshold: a) for the Naïve Bayes classifier, b) for the decision-tree (C4.5) classifier, and c) for the SVM classifier, for the original data (1-10) and our 180 experiments.

Experiments using 10-fold cross-validation

In figure 7.10 we show the tangent thresholded classification results. The number of components selected varies between 1 and 63 depending on the combination of measure/image parameter. For direct comparison the experiments were ordered the same way as on figure 7.8.

In these thresholded components classification results we get very similar results in comparison to classification results that use only 10 components for NB and C4.5, and in some cases we get considerable drops e.g., for the *Chebychev* measure (D5) (see experiments 51-60). This is due to the fact that the tangent based thresholding selects less than 10 components per combination of dissimilarity measure - image parameter and in some instances even only 1 component (see figure 7.9). An interesting thing to notice is that the overall classification percentages increase consistently for the *KLD A-B* and *KLD B-A* (D12 and D13) combinations (experiments 121 to 140), although this might be due to the fact that the tangent thresholding selected 63 components for several of the image parameter combinations. We also observe that providing large numbers of components to greedy classifiers (NB and C4.5) does not help them improve much, whereas SVM take clear advantage of more data.

With the aforementioned results for both of the tangent thresholding and the 10 component limiting we observe that with only 10 components we can achieve good accuracy results (around 80-90%) for the selected classifiers. This translates to an average 70% dimensionality reduction of our original dataset. We can also see which image parameters perform the best with which measures, which was one of our objectives with this research.

Quantitative Evaluation via Comparative Analysis of Classifiers for the Fractional Minkowski Metrics

In figures 7.11 and 7.12, we take an in-depth look at the classification results for the *Fractional Minkowski* dissimilarity measures (D14 to D18) paired with our other three dissimilarity measures based on the *Minkowski* metric. These dissimilarity measures are: *City Block* distance ($p=1$) (D4), *Euclidean* distance ($p=2$) (D1), and *Chebyshev* measure ($p=\infty$) (D5).

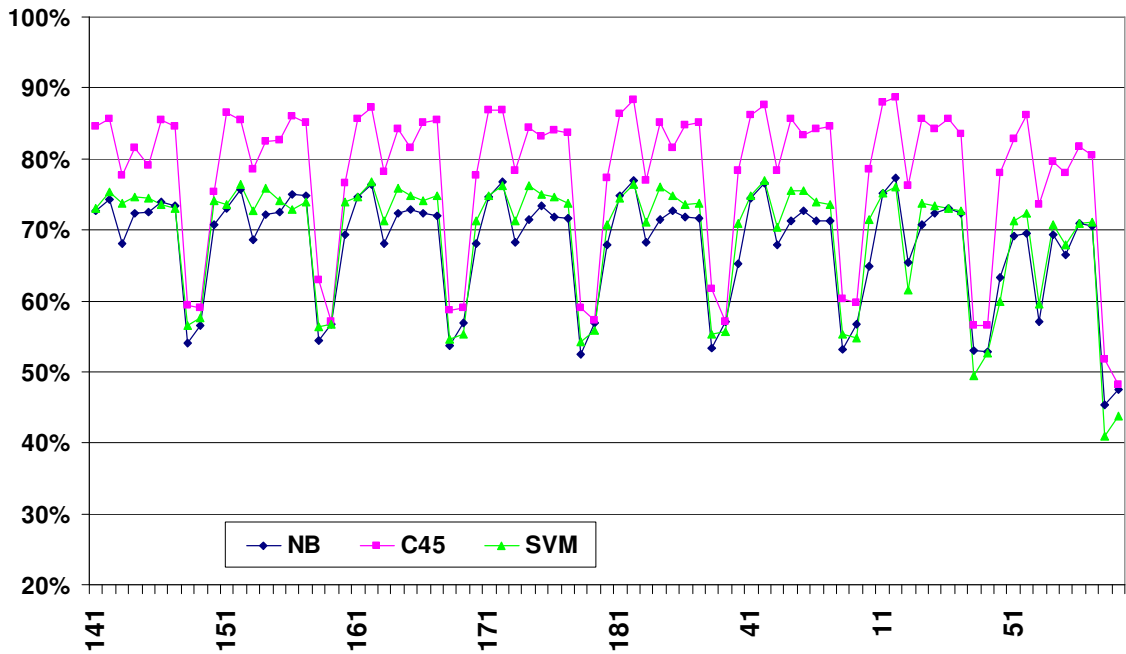


Figure 7.11. Percentage of classification accuracy for the 10-component Minkowski based metrics with $p = .25$ to $p = \infty$ (D14-18, D4, D1, D5), ordered by image parameter and using the same ordering from table 7.1. Experiments using 10-fold cross-validation

As we can see in figures 7.8 and 7.10 (experiments 141 to 190) and in figure 7.11, the fractional metrics tend to perform in a very stable manner, dropping for the same image parameters (*Fractal Dimension* (P8) and *Tamura Directionality* (P9)) for all of the different p values we investigated. An interesting thing to notice in this figure is that as p

gets closer to 1, the experiments do not show a clear tendency of increasing classification accuracy. Also, while we have different results for intra-class separation for some class' behavior, as we show in section 7.15.3, the inter-class separation of the remaining classes seems to balance out the classification results. We also have bigger drops as p gets close to infinity for the *Chebyshev* dissimilarity measure (D5), indicating that *Minkowski* metrics with a p closer to 2, perform better (and more stably) than this dissimilarity measure. A final observation from figure 7.11 is that with the 10-component threshold the tree-based classifier (C4.5) achieves the best classification results, while SVM seems to stay almost 10% behind for this low number of components. However, this will change in high dimensional space, which we will discuss in the next paragraph.

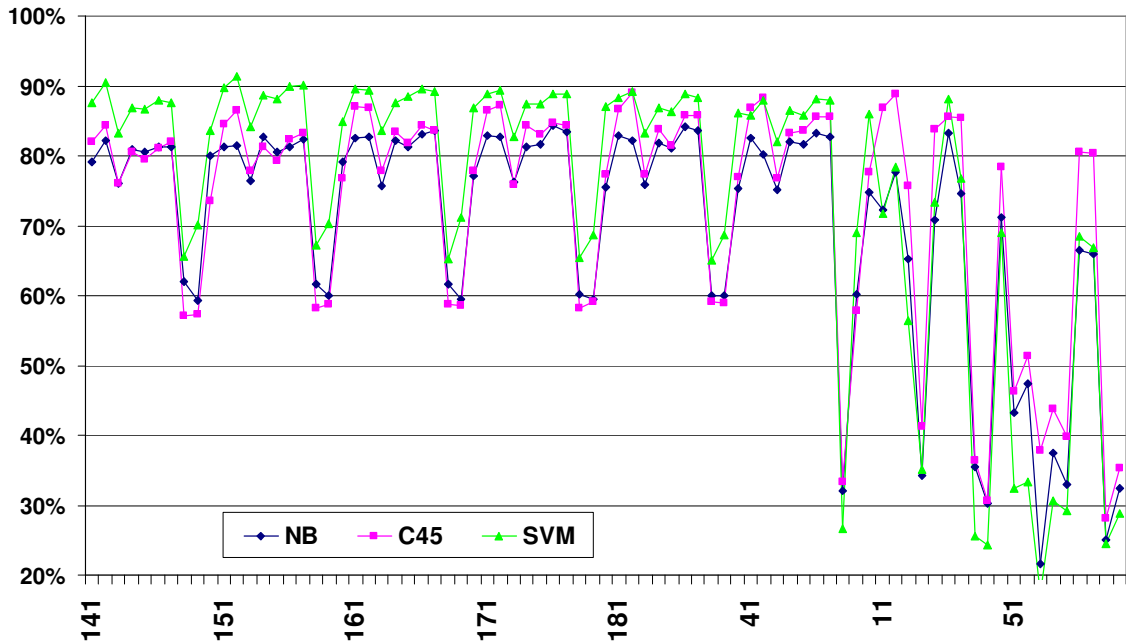


Figure 7.12. Percentage of classification accuracy for the Tangent thresholded Minkowski based fractional metrics with $p = .25$ to $p = \infty$ (D14-18, D4, D1, D5), ordered by image parameter and using the same ordering from table 7.1. Experiments using 10-fold cross-validation

In figure 7.12, we see two very interesting differences between the tangent threshold and our 10-component threshold. The first is that SVM takes a definite lead in terms of classification, with accuracy approaching 90% for the majority of dissimilarity measure/image parameters pairs and reaching our overall best result of 94.4% for the *mahalanobis* dissimilarity measure (D3) combined with the *mean* image parameter (P2). The tree-based classifier C4.5 keeps almost the same classification accuracy as with 10-components, causing us to believe that high dimensionality does not benefit this classification model. The improvement of the SVM accuracy shown in figure 7.10, is due to the fact that we selected a considerably larger amount of components with the tangent threshold, between 46 and 63, with the majority being 63 (see figure 7.9). This greatly improves the performance of the SVM classifier in the majority of our experiments. The selection of this high number of components is due to the fact that the decrease in magnitude of the components is very small making curve fitting ineffective

The second thing to notice is that our classification accuracy drops considerably for p values greater and equal to one, over 30-40% for some image parameters, and for the majority of the *Chebyshev* measure (D5) results. This could be partly due to the fact that the number of components selected is less than 30 for most instances (see figure 7.10, experiments 41 to 50 for $p = 1$ (D4), 11 to 20 for $p = 2$ (D1), and 51 to 60 for $p = \infty$ (D5)), and the actual dissimilitude found for the combination of dissimilarity measure - image parameter in question. In table 7.3 we present the top 10 classification results for both the tangent thresholded and the 10 components limited datasets. This will help us to quantitatively evaluate the differences between these two methods.

Table 7.3. Top 10 classification accuracy results for 10 component thresholded dimensionality reduction experiments. D indicates dissimilarity measure and P indicates image parameter number from table 4.1

NB		C45		SVM	
D13-P8	82.44%	D7-P2	90.63%	D8-P2	81.13%
D12-P8	81.25%	D12-P2	90.63%	D7-P2	80.19%
D12-P2	78.88%	D8-P2	89.63%	D7-P1	77.88%
D7-P2	78.75%	D1P2	88.69%	D8-P7	77.44%
D8-P2	78.19%	D10-P2	88.63%	D8-P6	77.44%
D13-P2	78.06%	D2-P2	88.25%	D7-P7	77.13%
D12-P1	77.81%	D18-P2	88.25%	D7-P4	77.00%
D1-P2	77.25%	D1P1	87.88%	D7-P6	77.00%
D7-P1	76.94%	D7-P1	87.69%	D4-P2	76.88%
D18-P2	76.88%	D4-P2	87.50%	D8-P1	76.88%

Table 7.4. Top 10 classification accuracy results for tangent thresholded dimensionality reduction experiments. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C indicates number of components

NB		C45		SVM	
C-63-D12-P8	85.50%	C-21-D12-P2	89.38%	C-63-D3-P2	94.44%
C-32-D7-P1	85.06%	C-27-D8-P2	89.19%	C-59-D2-P2	93.88%
C-29-D7-P2	84.69%	C-12-D1-P2	88.81%	C-63-D3-P1	92.13%
C-59-D2-P2	84.31%	C-59-D2-P2	88.75%	C-63-D12-P8	92.13%
C-27-D8-P2	84.19%	C-14-D13-P2	88.75%	C-63-D13-P8	91.38%
C-63-D13-P8	83.94%	C-29-D7-P2	88.63%	C-63-D12-P10	91.00%
C-35-D8-P7	83.75%	C-15-D10-P2	88.31%	C-63-D3-P4	90.56%
C-35-D8-P6	83.75%	C-27-D4-P2	88.25%	C-63-D3-P7	90.06%
C-63-D3-P7	83.25%	C-32-D7-P1	88.25%	C-63-D13-P10	89.50%
C-35-D4-P6	83.25%	C-25-D13-P6	87.31%	C-63-D3-P5	89.38%

In table 7.3 (the 10-component threshold) we have very different measures performing the best for the different classifiers. In terms of our image parameters, *mean* (P2) manages to appear in our top 10 almost 50% of the times, next to *fractal dimension* (P8), and *entropy* (P1). It is interesting to note that the *KLD* (D12 and D13) measure seems to perform very well for our Naïve Bayes (NB) classifier, taking five out of 10

spots in this table. *Correlation* (D7) also takes the first two spots when combined with the *mean* parameter (P2). Both directions of this *KLD* measure (*A-B* and *B-A*) (D12-D13) seem to appear close to each other, indicating that we might be able to only use one of the directions and considerably reduce its computational expense. We noticed that for the remaining two classifiers (C4.5 and SVM) we have different combinations of the *mean* (P2) parameter and very different types of measures (*Correlation* (D7), *KLD* (D12 and D13) and *Spearman* (D8)) taking the first few places. However, the most interesting result for us was that the fractional metrics only appear twice in this table, showing that they are not very good for our dataset. They are also easily beaten by the *euclidean* distance (D1) ($p=2$) on both occasions, showing a different trend than in the artificial dataset comparison presented in [129, 130] in terms of classification accuracy.

When it comes to the tangent thresholded classification results in table 7.4, we have different trends when it comes to the top 10 classification results. The combinations presented for the Naïve Bayes classification results only beat the 10-component results by 3-4% and have almost three times as many components (from 29 to 63) showing that the 10-component threshold actually performs pretty well when it comes to this classifier. We also have surprising results for the C4.5 classifier. Here, for the top 3 results, we actually have a drop in accuracy of 1% when using three times as many components, showing the behavior we mentioned about trees keeping similar classification results for both thresholding methods (but with a considerable increase in the number of data points). The combination of dissimilarity measure - image parameter seems to hold for these two classifiers when it comes to achieving the highest accuracy. One thing to note

again is that none of the *fractional metrics* (D14-D18) factor in when it comes to this thresholding method. This reinforces the claim that they are very stable (hence their classification accuracy is not increasing for a higher number of components selected by the tangent thresholding method), but they are also not ideal for our own solar dataset.

As we have shown in this chapter, the analysis of different dissimilarity measures for purposes of attribute evaluation is something very useful when it comes to discovering interesting relationships in our data. These experiments and their results have shown that using the dissimilarity measures together with Multi-Dimensional Scaling (MDS) we can achieve some certain degree of dimensionality reduction by taking advantage of the underlying properties of the data that are discovered through the different measures. The biggest drawback in utilizing MDS for dimensionality reduction in a CBIR system is the fact that the method does not provide a proper mapping function for unseen data. This makes the method not very useful when we are looking to add new unseen data to the CBIR system. To overcome this, we add to our framework the capabilities of analyzing dimensionality reduction methods that do poses proper mapping functions for unseen data. The usefulness and selection of an optimal method we suspect is again very domain dependent and we will provide an analysis of this in chapter 8.

For illustrative purposes and for other researchers interested in looking at different types of images (medical, natural scenes, etc), we present some of the results of the dissimilarity measures evaluation with the remaining datasets (proposed in chapter 3) in the following pages. We will present at table of the top 10 performing combinations of dissimilarity measure / image parameters for both the tangent and the 10-component

thresholds, for all 3 classifiers. We will then present the scaled image plot, 2D and 3D MDS plots for one of top performing combinations of dissimilarity measure - image parameter and provide a brief discussion about them for each of the datasets. The remaining scaled image plots, MDS plots, and classification results as well as the data files to reproduce these experiments for the solar and the rest of the datasets can be found at [122].

Table 7.5. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on ImageCLEFMed05 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components

NB		C45		SVM	
C-37-D8-P2	78.69%	C-63-D13-P7	89.31%	C-41-D13-P1	88.00%
C-36-D7-P2	76.63%	C-63-D12-P6	88.94%	C-40-D12-P1	85.75%
C-15-D1-P2	76.31%	C-41-D13-P1	88.81%	C-37-D12-P2	85.31%
C-40-D12-P1	76.31%	C-63-D12-P7	88.69%	C-43-D13-P2	85.19%
C-10-D7-P2	75.25%	C-63-D12-P5	88.56%	C-10-D12-P2	84.19%
C-10-D1-P2	75.19%	C-63-D13-P6	88.56%	C-63-D13-P8	83.06%
C-63-D13-P8	74.81%	C-40-D12-P1	88.31%	C-10-D13-P2	82.31%
C-10-D8-P2	74.69%	C-38-D4-P2	86.88%	C-10-D13-P1	82.06%
C-38-D4-P2	74.63%	C-37-D8-P2	86.63%	C-10-D12-P1	81.50%
C-10-D17-P2	74.44%	C-63-D12-P3	86.56%	C-10-D13-P8	81.06%

Table 7.6. Scaled image plot ranges and colors corresponding to the classes of the ImageCLEFMed05 dataset

Class Name	Scaled Image Plot Range	MDS Color
Plain Radiography - Frontal Cranium	1-200	Red
Plain Radiography - Trachea	201-400	Green
Plain Radiography - Spine	401-600	Blue
Plain Radiography - Hand	601-800	Yellow
Plain Radiography - Chest	801-1000	Magenta
Plain Radiography - Pelvis	1001-1200	Gray
Plain Radiography - Feet	1201-1400	Orange
Plain Radiography - Cranium	1401-1600	Black

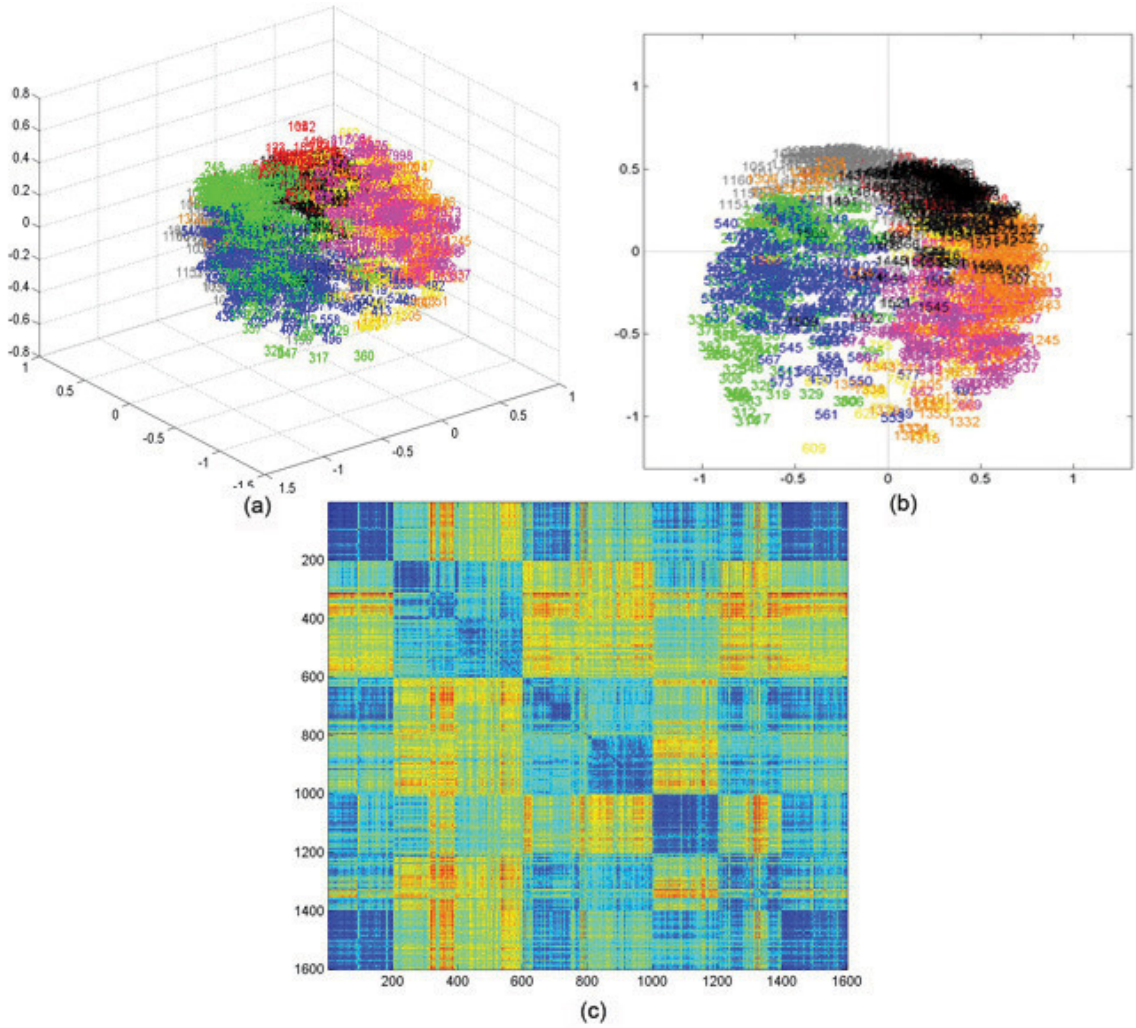


Figure 7.13. (a) 3D MDS plot, (b) 2D MDS plot, (c) Scaled Image Plot, for the ImageCLEFMed05 Dataset combination of Spearman (D8) measure with the *mean* (P2) image parameter

For the ImageCLEFMed05 dataset, this C-37-D8-P2 experiment actually appears in the top 10 for two different classifiers. If you notice in figure 7.13, we can see a somewhat clear separation of the classes (different colored clusters on (a) and (b)), and nice blocks of lower dissimilarity on (c). This behavior is one of the main things we are trying to outline with this dissimilarity measure evaluation module.

Table 7.7. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on ImageCLEFMed07 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components

NB		C45		SVM	
C-10-D7-P2	85.44%	C-63-D12-P6	94.31%	C-40-D12-P1	90.44%
C-10-D8-P2	85.31%	C-63-D13-P7	92.81%	C-54-D13-P1	90.44%
C-40-D12-P1	84.06%	C-63-D12-P7	92.69%	C-43-D13-P2	89.81%
C-38-D8-P2	83.94%	C-40-D12-P1	92.31%	C-63-D13-P8	89.19%
C-39-D7-P2	83.81%	C-63-D13-P6	92.13%	C-32-D12-P2	89.00%
C-10-D16-P2	83.38%	C-54-D13-P1	91.81%	C-63-D12-P8	88.75%
C-10-D14-P2	82.94%	C-63-D12-P5	91.75%	C-10-D13-P1	88.69%
C-10-D17-P2	82.88%	C-63-D12-P8	91.31%	C-10-D12-P2	87.88%
C-10-D12-P1	82.63%	C-59-D13-P4	90.88%	C-10-D13-P2	87.75%
C-10-D4-P2	82.63%	C-40-D4-P2	90.56%	C-10-D12-P1	87.38%

Table 7.8. Scaled image plot ranges and colors corresponding to the classes of the ImageCLEFMed07 dataset

Class Name	Scaled Image Plot Range	MDS Color
Plain Radiography - Chest	1-200	Red
Plain Radiography - Hand	201-400	Green
Plain Radiography - Cranium	401-600	Blue
Plain Radiography - Side Chest	601-800	Yellow
Plain Radiography - Trachea	801-1000	Magenta
Plain Radiography - Breast	1001-1200	Gray
Plain Radiography - Lungs	1201-1400	Orange
Plain Radiography - Leg Bones	1401-1600	Black

An interesting thing to note here is that we are starting to notice the *KLD* measure (D12 and D13) to appear more often in the top results. We will go into detail about why this happens when we talk about the INDECS and PASCAL datasets in the next few pages. For now, we selected the C-40-D4-P2 experiment (*City block* measure (D4) with *mean* parameter (P2)) as the one to analyze due to the fact that it appears in both the NB and the C45 top 10's.

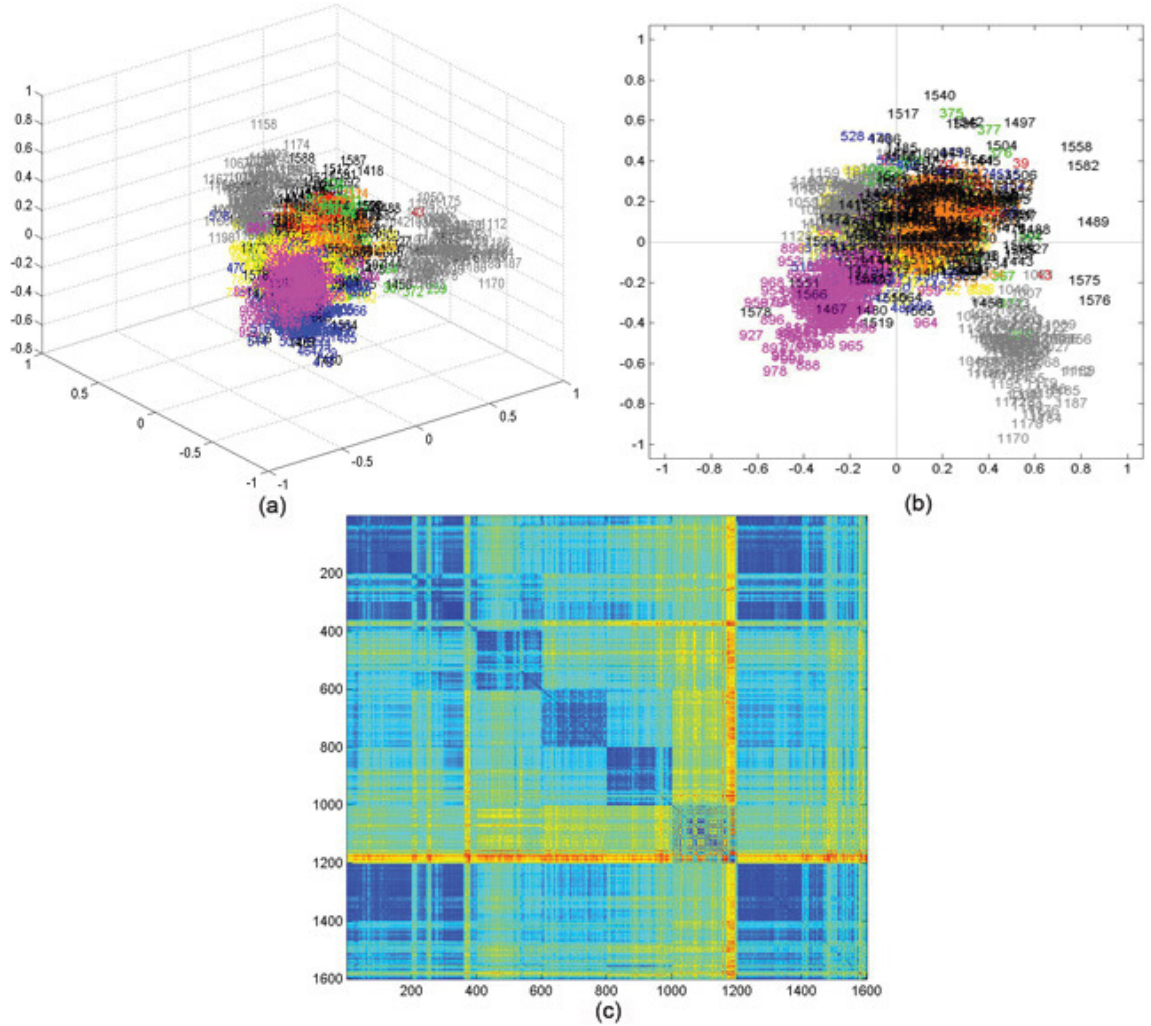


Figure 7.14. (a) 3D MDS plot, (b) 2D MDS plot, (c) Scaled Image Plot, for the ImageCLEFMed07 Dataset combination of City-block (D4) measure with the *mean* (P2) image parameter

Again for the ImageCLEFMed07 dataset figure 7.14 MDS plots ((a) and (b) we can actually see somewhat clear separation for two of the classes (magenta and gray). In the scaled image plot (c), we can see exactly where it happens (the 800 to 1200) range. These plots show to be of great aid when identifying clusters (and hence improving classification) on datasets that can be separated nicely by our combinations of dissimilarity measures and image parameters.

For the experimental results on the INDECS dataset we found very surprising and interesting results that show how useful some of the measures/image parameters combinations (*KLD* with everything) can be in finding the underlying dissimilarities between the data. Aided by MDS, these dissimilarities can considerably improve our classification results. However, due to the fact that MDS does not provide a proper mapping function, these results are interesting in theory but not in actual practice since we can't properly map new data to this very efficient new dimensional space.

Table 7.9. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on INDECS dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components

NB		C45		SVM	
C-63-D13-P8	87.94%	C-63-D12-P8	92.25%	C-63-D12-P8	90.56%
C-63-D12-P8	84.06%	C-63-D13-P8	90.88%	C-63-D13-P8	90.25%
C-10-D13-P8	83.75%	C-39-D13-P2	81.56%	C-10-D13-P8	85.75%
C-10-D12-P8	81.00%	C-63-D13-P6	80.94%	C-10-D12-P8	84.63%
C-34-D13-P1	66.69%	C-39-D12-P2	79.94%	C-34-D13-P1	66.19%
C-37-D12-P1	61.00%	C-10-D13-P8	79.81%	C-37-D12-P1	65.44%
C-39-D13-P2	59.88%	C-63-D13-P7	77.88%	C-39-D13-P2	62.19%
C-10-D13-P1	58.69%	C-63-D12-P6	77.81%	C-10-D13-P1	59.88%
C-10-D12-P1	57.06%	C-10-D12-P8	77.56%	C-39-D12-P2	58.94%
C-39-D12-P2	54.25%	C-63-D12-P7	75.19%	C-10-D12-P1	57.38%

Table 7.10. Scaled image plot ranges and colors corresponding to the classes of the INDECS dataset

Class Name	Scaled Image Plot Range	MDS Color
Corridor – Cloudy	1-200	Red
Corridor – Night	201-400	Green
Kitchen – Cloudy	401-600	Blue
Kitchen – Night	601-800	Yellow
Kitchen – Sunny	801-1000	Magenta
Two-persons Office - Cloudy	1001-1200	Gray
Two-persons Office - Night	1201-1400	Orange
Two-persons Office - Sunny	1401-1600	Black

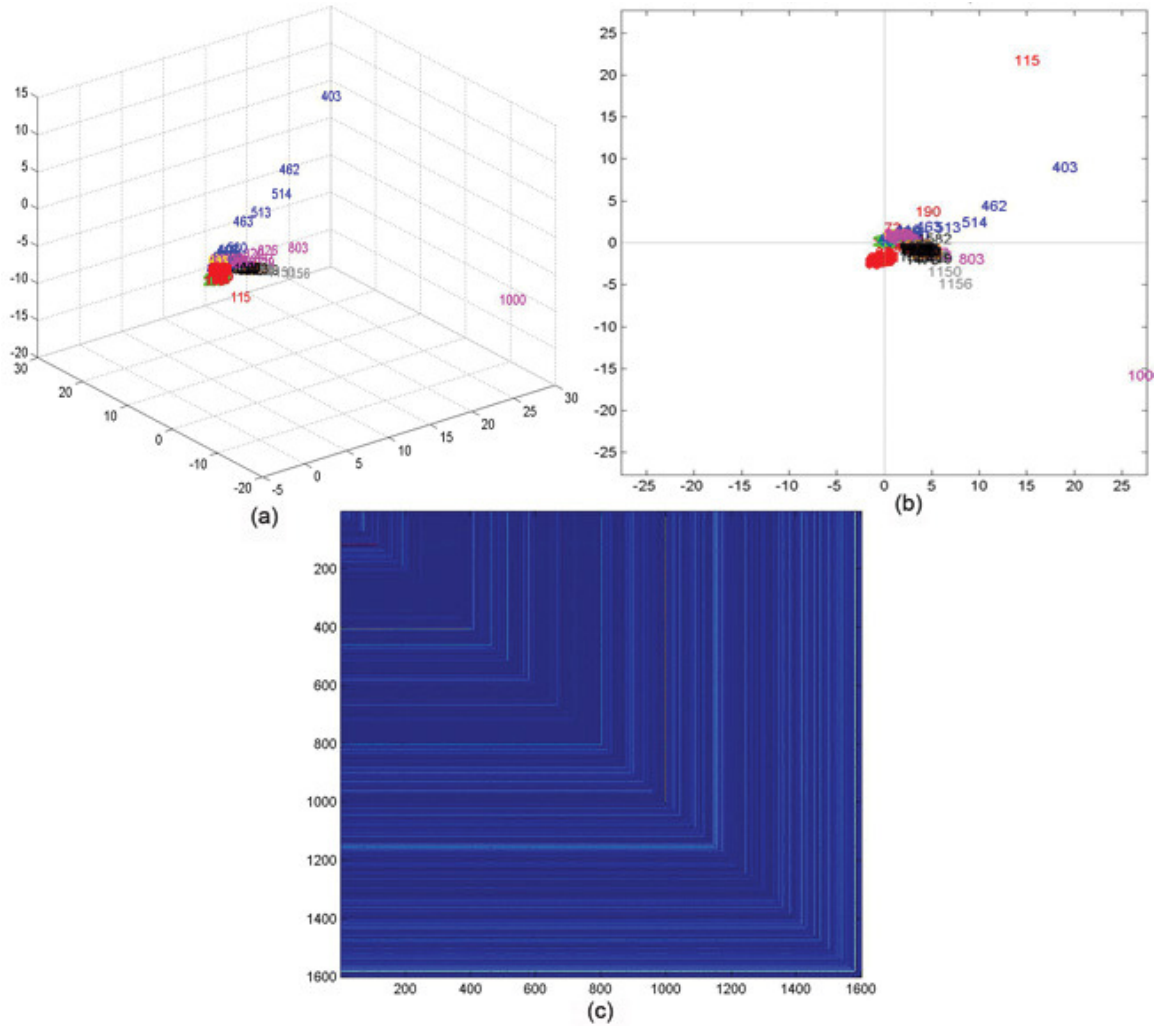


Figure 7.15. (a) 3D MDS plot, (b) 2D MDS plot, (c) Scaled Image Plot, for the INDECS Dataset combination of *KLD A-B* (D12) measure with the *fractal dimension* (P8) image parameter

Initially we can see that the 3D and 2D MDS plots from figure 7.15 are not very helpful since almost everything seems to be very close. This is demonstrated on the scaled image plot in (c). The few images that seem out of the main clusters of (a) and (b) are also reflected in the lighter lines on (c), showing them as outliers or dissimilar to the rest. Intuitively this might lead you to believe that classification on such results might be very bad, however, the MDS with 10 and 63 components actually seems to have more

easily separable clusters (we can't visualize all 10 dimensions at once, but the classification results show this). This has to be thanks to the particular combination of the fractal dimension parameter with the *KLD* (both *A-B* and *B-A*) measures for both the tangent and the 10-component thresholding experiments (top 4 on table 7.9). Immediately after we have a drop of 15% accuracy for the next combination of dissimilarity measure and image parameter, and this trend continues until reaching the 30% accuracy the original 640 dimensional INDECS dataset achieves. This just shows that while the *fractal dimension* seems to be a bad image parameter for these types of images when looked at with the Euclidean distance (as traditional correlation analysis, traditional classification and dimensionality reduction algorithms rely on), it can be very useful in separating the INDECS images when paired with the *KLD* measure. We see similar results for the PASCAL 2006 and PASCAL2008 datasets for several different image parameters when paired with the *KLD* measure.

This leads to two very interesting things 1) That image parameter and *KLD* works very well for some particular datasets making that dissimilarity measure - image parameter combination valuable for that dataset only (domain specific). 2) Our dissimilarity measure analysis mode is greatly validated by these experimental results since we uncovered a parameter that seemed very un-useful to be actually quite useful when properly paired with a certain dissimilarity measure. We can also say that for the datasets where our parameters are selected properly (i.e solar, ImageCLEFMed05 and 07), the *KLD* makes some impact but not a considerable one and we have other parameters/measures taking the lead (best classification accuracy results), whereas for the

datasets where the parameters are not properly selected we have uncovered very interesting results while using this measure.

Table 7.11. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on PASCAL2006 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components

NB		C45		SVM	
C-62-D13-P8	80.56%	C-63-D12-P5	86.63%	C-62-D13-P8	89.25%
C-60-D12-P8	78.81%	C-62-D13-P8	86.00%	C-60-D12-P8	88.06%
C-48-D13-P1	78.75%	C-63-D12-P6	85.44%	C-47-D12-P1	82.69%
C-10-D12-P8	78.38%	C-60-D12-P8	85.31%	C-48-D13-P1	81.69%
C-63-D12-P5	78.06%	C-48-D13-P1	84.81%	C-10-D12-P8	79.50%
C-47-D12-P1	77.94%	C-63-D13-P5	84.13%	C-10-D13-P8	79.50%
C-10-D13-P8	74.81%	C-63-D13-P6	83.94%	C-32-D12-P2	77.19%
C-63-D12-P7	74.81%	C-55-D12-P4	83.69%	C-31-D13-P2	76.44%
C-63-D13-P6	73.81%	C-63-D13-P7	81.88%	C-63-D12-P3	75.94%
C-55-D12-P4	72.25%	C-63-D12-P7	81.81%	C-63-D13-P3	72.31%

Table 7.12. Scaled image plot ranges and colors corresponding to the classes of the PASCAL2006 dataset

Class Name	Scaled Image Plot Range	MDS Color
Car	1-200	Red
Bicycle	201-400	Green
Cat	401-600	Blue
Cow	601-800	Yellow
Dog	801-1000	Magenta
Horse	1001-1200	Gray
Person	1201-1400	Orange
Sheep	1401-1600	Black

We again see that *KLD* (D12 and D13) measure appears on the top spots and as we gain considerable classification accuracy improvements. The same parameters as before P8 and P1 (*fractal dimension* and *entropy*) seem to be benefited for these PASCAL datasets

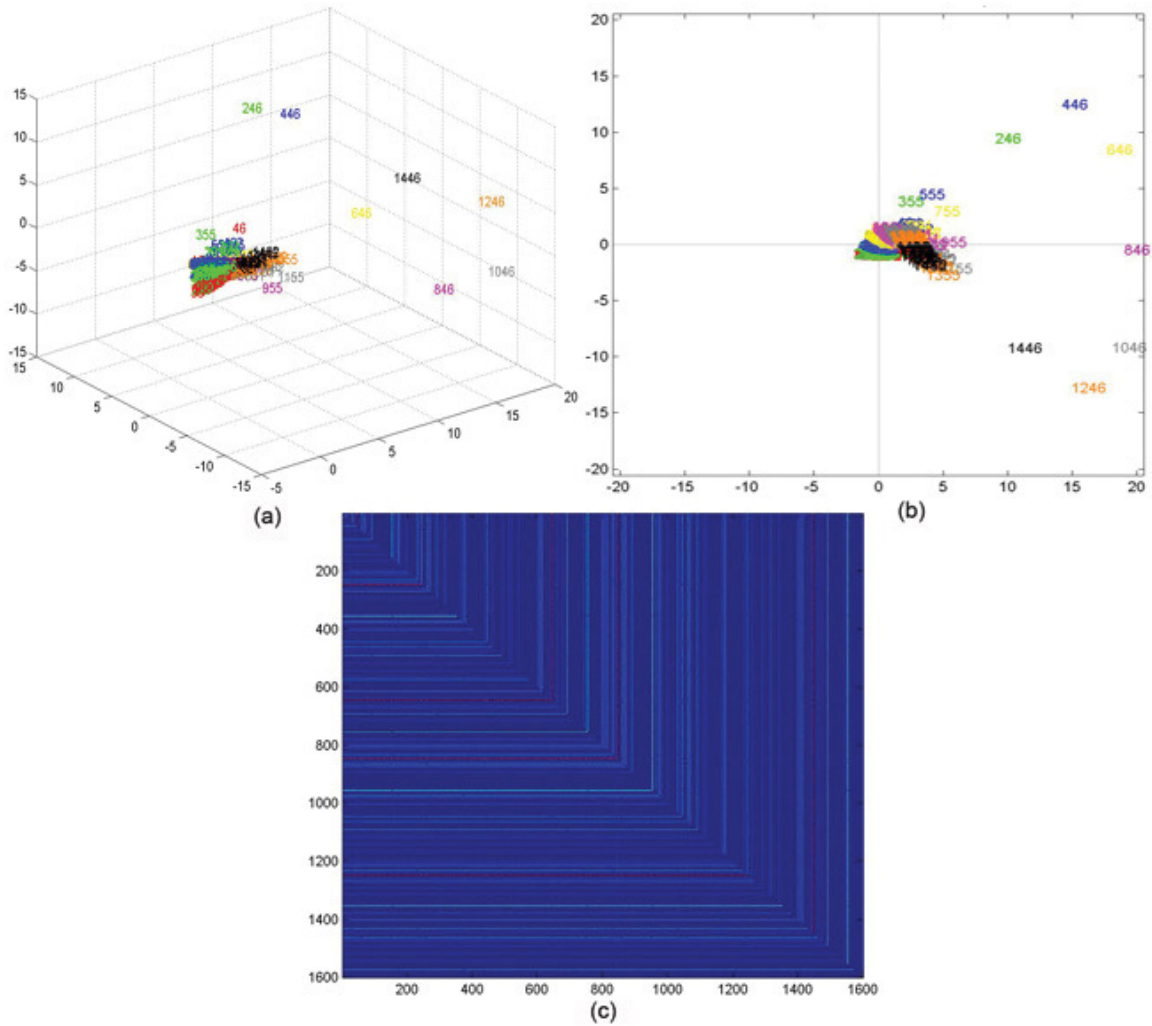


Figure 7.16. (a) 3D MDS plot, (b) 2D MDS plot, (c) Scaled Image Plot, for the PASCAL2006 Dataset combination of $KLD\ A-B$ (D12) measure with the $fractal\ dimension$ (P8) image parameter

We again see the same behavior as for the INDECS dataset on figure 7.16, the only noticeable difference is that there are more outlying (high dissimilarity) images hence more numbers outside the clusters in (a) and (b), and different colored ‘lines’ on (c). These results serve to validate the usefulness of the analysis of all these measures by uncovering the very useful relationship of the KLD measure with a few image parameters when it comes to analyzing datasets that contain highly dissimilar images.

Table 7.13. Top 10 classification accuracy results for both thresholded dimensionality reduction experiments on PASCAL2008 dataset. D indicates dissimilarity measure, P indicates image parameter number from table 4.1, and C- indicates number of components

NB		C45		SVM	
C-26-D12-P1	82.25%	C-63-D13-P8	89.63%	C-63-D13-P8	91.00%
C-27-D13-P1	82.06%	C-63-D12-P8	88.94%	C-63-D12-P8	88.63%
C-10-D12-P8	79.94%	C-26-D12-P1	85.94%	C-26-D12-P1	86.63%
C-10-D13-P8	79.63%	C-27-D13-P1	84.00%	C-10-D13-P8	85.94%
C-63-D13-P8	78.06%	C-63-D13-P6	82.94%	C-27-D13-P1	84.81%
C-63-D12-P8	77.69%	C-63-D12-P6	82.25%	C-10-D12-P1	82.25%
C-10-D13-P1	74.31%	C-10-D13-P8	79.75%	C-10-D12-P8	81.88%
C-10-D12-P1	73.44%	C-63-D12-P7	79.38%	C-10-D13-P1	76.25%
C-39-D12-P3	67.63%	C-63-D13-P7	78.56%	C-39-D12-P3	75.81%
C-10-D12-P3	64.75%	C-63-D12-P5	76.69%	C-37-D13-P3	73.94%

Table 7.14. Scaled image plot ranges and colors corresponding to the classes of the PASCAL2008 dataset

Class Name	Scaled Image Plot Range	MDS Color
Car	1-200	Red
Bicycle	201-400	Green
Bottle	401-600	Blue
Boat	601-800	Yellow
Dog	801-1000	Magenta
Tv Monitor	1001-1200	Gray
Person	1201-1400	Orange
Cat	1401-1600	Black

As we anticipated, the *KLD* measure achieves the top 10 (and even top 40 (2 *KLD* measures times 20 different experiments)) classification results bumping the original classification results by over 60%. This measure does in fact gradually reduce its accuracy with different parameters and reaching the ‘normal’ 20% accuracy for this dataset.

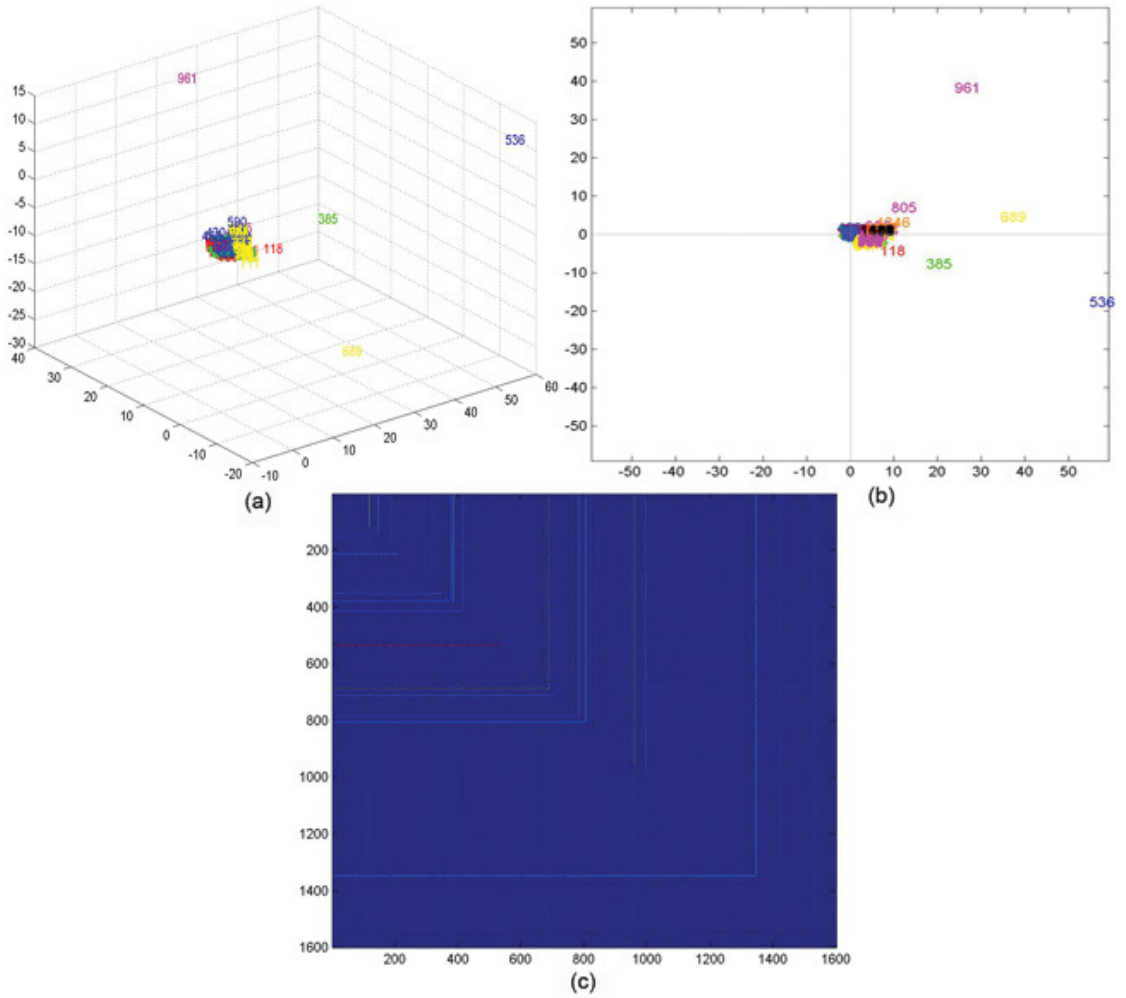


Figure 7.17. (a) 3D MDS plot, (b) 2D MDS plot, (c) Scaled Image Plot, for the PASCAL2008 Dataset combination of *KLD A-B* (D12) measure with the *entropy* (P1) image parameter

While the PASCAL2008 dataset seems to have less outlying images, we still observe the same behavior of dissimilarity between all images and just a few very different ones according to the measure (the outliers). This explains the single cluster on figure 7.17 (a) and (b), similar behavior than on the INDECS and PASCAL2006 datasets. This results again in high classification results for the combinations of the *KLD* measure and most image parameters. This again validates the need to analyze these relationships (in the dissimilarity evaluation module of our framework) and find these peculiar cases when seemingly bad image parameters perform well for certain datasets.

CHAPTER 8

DIMENSIONALITY REDUCTION

Introduction

Dimensionality reduction methods have been shown to produce accurate representations of high dimensional data in a lower dimensional space. Applications of some of these methods have already been successfully implemented in image retrieval applications in other domains [139, 140, 141, 142, 143]. With the need for greatly reducing data storage and processing costs in mind, in this work we investigate four linear and four non-linear dimensionality reduction methods with eight different numbers of dimensions as parameters for each, in order to present a comparative analysis. This will help us determine which method is optimal for our specific dataset and what percentage of dimensionality reduction we can achieve to alleviate our search, retrieval, and storage costs.

The novelty of our work is to help determine which dimensionality reduction methods produce the best and most consistent results and with which classifiers, on specific image parameters selected for solar data. Due to domain-specific results, reported by multiple-researchers working on dimensionality reduction in the past [139, 140, 141, 142, 143], we believe our results will be also of special interest to researchers from the field of medical image analysis, as these images seem to be the closest to our dataset. We also identify some interesting combinations of dimensionality reduction methods and classifiers that behave differently across the presented datasets. Based on

our literature review, we decided to utilize four different linear and four non-linear dimensionality reduction methods. Based on previous works [140, 143, 144] linear dimensionality reduction methods have proved to perform better than non-linear methods in most artificial datasets and some natural datasets, however all these results have been domain dependent. Classical methods like Principal Component Analysis (PCA) [16] and Singular Value Decompositions (SVD) [145] are widely used as benchmarks to provide a comparison versus the newer non-linear methods. We selected our eight different methods based on their popularity in the literature, the availability of a mapping function or methods to map new unseen data points into the new dimensional space, computational expense, and the particular properties of some methods such as the preservation of local properties between the data and the types of distances between the data points (Euclidean versus geodesic).

Linear Dimensionality Reduction Methods

In the following definitions of the dimensionality reduction methods, our feature vector becomes the m -by- n matrix M , where m indicates the number of images (1,600) and n indicates the number of dimensions of our feature vector (640).

Principal Component Analysis (PCA) [16]

PCA is defined as an orthogonal linear transformation that transforms data into an artificial space such that the greatest variance by any projection of the data comes to lie on the first principal component, and so on.

For a data matrix, M^T , where each row represents a different observation on the

dataset, and each column contains the variables, the PCA transformation is given by:

$$Y^T = M^T U = V \Sigma^T \quad (8.1)$$

Where the matrix Σ is an m -by- n diagonal matrix with nonnegative real numbers on the diagonal and $U \Sigma V^T$ is the singular value decomposition of M . The low-dimensional representations y_i of the m_i data points in the low-dimensional space Y are computed by multiplying the M matrix by the resulting V matrix.

Singular Value Decomposition (SVD) [145]

Singular Value Decomposition is defined for an m -by- n matrix M as the factorization of the form

$$M = U \Sigma V^T \quad (8.2)$$

Where U is an m -by- m unitary matrix, the matrix Σ is m -by- n diagonal matrix with nonnegative real numbers on the diagonal, and V^T denotes the conjugate transpose of V , an n -by- n unitary matrix. Normally, the diagonal entries $\Sigma_{i,i}$ are ordered in a descending way. The columns of V form a set of orthonormal input basis vector directions for M . (The eigenvectors of $M^T M$.) The columns of U form a set of orthonormal output basis vector directions for M . (The eigenvectors of MM^T). The low-dimensional representations y_i of the m_i data points in the low-dimensional space Y are computed by multiplying the M matrix by the resulting V matrix.

The main difference from a practical point of view, between PCA and SVD is that SVD is directly applied to an m -by- n matrix (i.e. m -by- n long feature vectors), while in PCA, SVD is applied to a covariance matrix (m -by- m). This results with the starting

matrix being different for each case.

Factor Analysis (FA) [146]

FA is a statistical method used to describe variability among observed variables in terms of a potentially lower number of unobserved variables called factors. The information gained about the interdependencies between observed variables is used later to reduce the set of variables in a dataset, achieving dimensionality reduction.

To perform FA we compute the maximum likelihood estimate of the factor loadings matrix Λ in the factor analysis model

$$M = \mu + \Lambda f + e \quad (8.3)$$

Where M is the m -by- n data matrix, μ is a constant vector of means, Λ is a constant m -by- n matrix of factor loadings, f is a vector of independent and standardized common factors, and e is a vector of independent specific factors. M , μ , and e are of length m and f is of length n . The low-dimensional representations y_i of the m_i data points in the low-dimensional space Y are computed by using a combination of f and the independent factors e .

Locality Preserving Projections (LPP) [147]

LPP generates linear projective maps by solving a variational problem that optimally preserves the neighborhood structure of a dataset. These projections are obtained by embedding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. The resulting mapping corresponds to the following eigenvalue problem:

$$MLM^T w = \lambda MDX^T w \quad (8.4)$$

Where L is the Laplacian matrix, (i.e. $D - S$, where S corresponds to the similarity values defined, and D is a column matrix which reflects how important certain projections are). LPP is defined everywhere in the ambient space, not only on the training data points like Isomaps and LLE. The low-dimensional representations y_i of the m_i data points in the low-dimensional space Y are computed by: $y_i = A^T m_i$, where $A = (w_0, w_1, \dots, w_{n-1})$.

Non-Linear Dimensionality Reduction Methods

Isomap [148]

Isomap preserves pair-wise geodesic distances between data points. The geodesic distances between the data points x_i of M are calculated by constructing a neighborhood graph G , in which every data point x_i is connected with its k nearest neighbor's x_{ij} in the dataset M . The low-dimensional representations y_i of the data point's x_i in the low-dimensional space Y are computed by applying multidimensional scaling (MDS) [125] on the resulting m -by- m geodesic adjacency matrix.

Kernel PCA [149]

Kernel PCA computes the principal eigenvectors of the kernel matrix, rather than those of the covariance matrix. The reformulation of PCA in kernel space is straightforward, since a kernel matrix is similar to the inner-product of the data points in the high dimensional space that is constructed using the kernel function

Kernel PCA computes the kernel matrix K of the data-points x_i . The entries in the

kernel matrix are defined by

$$k_{i,j} = (x_i, x_j) \quad (8.5)$$

Where $k_{j,i}$ is a kernel function. Then the kernel matrix K is centered by subtracting the mean of the features in traditional PCA.

To obtain the low-dimensional data representation, the data is projected onto the eigenvectors of the covariance matrix α_i . The entries of Y , the low dimensional representation are given by:

$$y_i = \left\{ \sum_{j=1}^n \alpha_1^j k(x_j, x_i), \dots, \sum_{j=1}^n \alpha_d^j k(x_j, x_i) \right\} \quad (8.6)$$

Where α_l^j indicates the j th value in the vector α_l and k is the kernel function used in the computation of the kernel matrix. Since Kernel PCA is a kernel-based method, the mapping performed by Kernel PCA relies on the choice of the kernel function k . For this work we utilized a Gaussian kernel.

Laplacian Eigenmaps (Laplacian) [17]

Laplacian Eigenmaps is another method that preserves the local properties of the manifold to produce a low-dimensional data representation. These local properties are based on the pair-wise distances between k nearest neighbors. Laplacian computes a low-dimensional representation of the data minimizing the distances between a data point and its k nearest neighbors.

The Laplacian Eigenmaps algorithm first constructs a neighborhood graph G in

which every data point x_i is connected to its k nearest neighbors. The weights of the edges on the connected data points of G are computed using the Gaussian kernel function, resulting in an adjacency matrix W . The computation of the degree matrix M and the graph Laplacian L of the graph W allows for the formulation of the minimization problem as an eigenproblem. The low-dimensional representation Y can be found by solving the generalized eigenvalue problem:

$$Lv = \lambda Mv \quad (8.7)$$

For the d smallest nonzero eigenvalues, where d is the number of desired dimensions for Y . The d eigenvectors v_i corresponding to the smallest nonzero eigenvalues form the low-dimensional representation Y . The d eigenvectors v_i corresponding to the smallest nonzero eigenvalues form the low-dimensional representation Y (e.g $y_i = (v_1(i), \dots, v_d(i))$)

Locally-Linear Embedding (LLE) [150]

Similarly to Isomap, LLE constructs a graph representation of the data points and attempts to preserve the local properties of the data allowing for successful embedding of non-convex manifolds. In LLE, the local properties of the data are constructed by writing the local properties of the manifold around a data point x_i by writing the data point as a linear combination W_i of its k nearest neighbors x_{ij} , by fitting a hyperplane through the data point x_i and its nearest neighbors assuming that the manifold is locally linear. If the low-dimensional data representation preserves the local geometry of the manifold, the reconstruction weights W_i that reconstruct data point x_i from its neighbors in the high-dimensional representation also reconstruct data point y_i from its neighbors in the low-

dimensional representation. Therefore finding the d dimensional data representation Y depends on minimizing the cost function:

$$\phi(Y) = \sum_i (y_i - \sum_{j=1}^k w_{ij} y_{ij})^2 \quad (8.8)$$

Where y_i is a data point in the low dimensional representation and w_{ij} is a reconstruction weight.

Dimensionality Thresholding

Thresholding via PCA and SVD variance percentage

For ‘optimal’ dimensional estimation we used the number of dimensions returned by standard PCA and SVDs, setting up a variance threshold between 96 and 99% of the variance. Table 8.1 presents these numbers of dimensions for all three datasets utilized.

Table 8.1. Number of dimensions selected for each dataset

Datasets tested	PCA Variance				SVD Variance			
	96%	97%	98%	99%	96%	97%	98%	99%
Solar	42	46	51	58	58	74	99	143
INDECS	94	106	121	143	215	239	270	319
ImageCLEFmed05	79	89	103	126	193	218	253	307
ImageCLEFmed07	76	87	102	126	192	217	251	304
Pascal2006	77	85	96	114	100	111	125	147
Pascal2008	115	127	141	160	212	239	275	331
Experiment Label	1	2	3	4	5	6	7	8

For our purposes, we think that the number of dimensions for our Solar dataset are safely determined by PCA and SVD with a variance between 96% and 99%. We managed to approach both sides of the peak in classification accuracy in most cases, indicating that PCA approaches the peak (highest accuracy) from the left side (low to high) and SVD behaves the opposite way. The complexity of estimating dimensions this

way in a much larger dataset might render these two techniques highly expensive since they rely on the calculation of eigenvectors, but they seem to be accurate enough compared to running experiments for all possible numbers of dimensions.

Mapping Functions

Since we are planning on applying the best dimensionality reduction method on new data we will be receiving for the next five years, we decided to simulate this scenario with our benchmark dataset.

In our experiments we performed the dimensionality reduction methods on 67% of our data and then map ‘new’ data points (the remaining 33% of our data) into the resulting low-dimensional representation that each dimensionality reduction method produces. We then feed these dimensionally reduced data points respectively as training and test sets to our classification algorithms.

For linear dimensionality reduction methods, the mapping of new data points is very straightforward since, for example, for PCA and SVD you only have to multiply the new data points by the linear mapping matrix V .

As for non-linear dimensionality reduction methods, the mapping of new data points is not straightforward and different techniques have been proposed. For Kernel PCA it is somewhat similar to the original PCA, but requires some additional kernel function computations as presented in [151]. For Isomaps, LLE, and LE, kernel methods have been presented in [152]. Alternative approaches have been presented in [153, 154, 155, 156, 157, 158]

We employed these techniques in order to map the remaining 33% of the data on to the resulting new low-dimensional spaces; mirroring, in a way, how we will be mapping the new solar data we receive after we define our mapping functions. By our comparative analysis using classifiers, we are also able to determine the quality of these mappings.

Evaluation Results

For our evaluation results we utilized the default settings for all classifiers since we are performing a comparative analysis and not a full optimization analysis. We selected 67% of our data as the training set and an ‘unseen’ 33% test set.

Six out of the eight (FA, KernelPCA, Isomap, LLE, LE, LPP) dimensionality reduction methods were investigated using the Matlab Tool box for dimensionality reduction [159]. For the remaining two, PCA and SVD, we utilized the standard Matlab functions. For the non-linear methods that utilize neighborhood graphs we used between 6 and 12 as the number of nearest neighbors, and presented the best results.

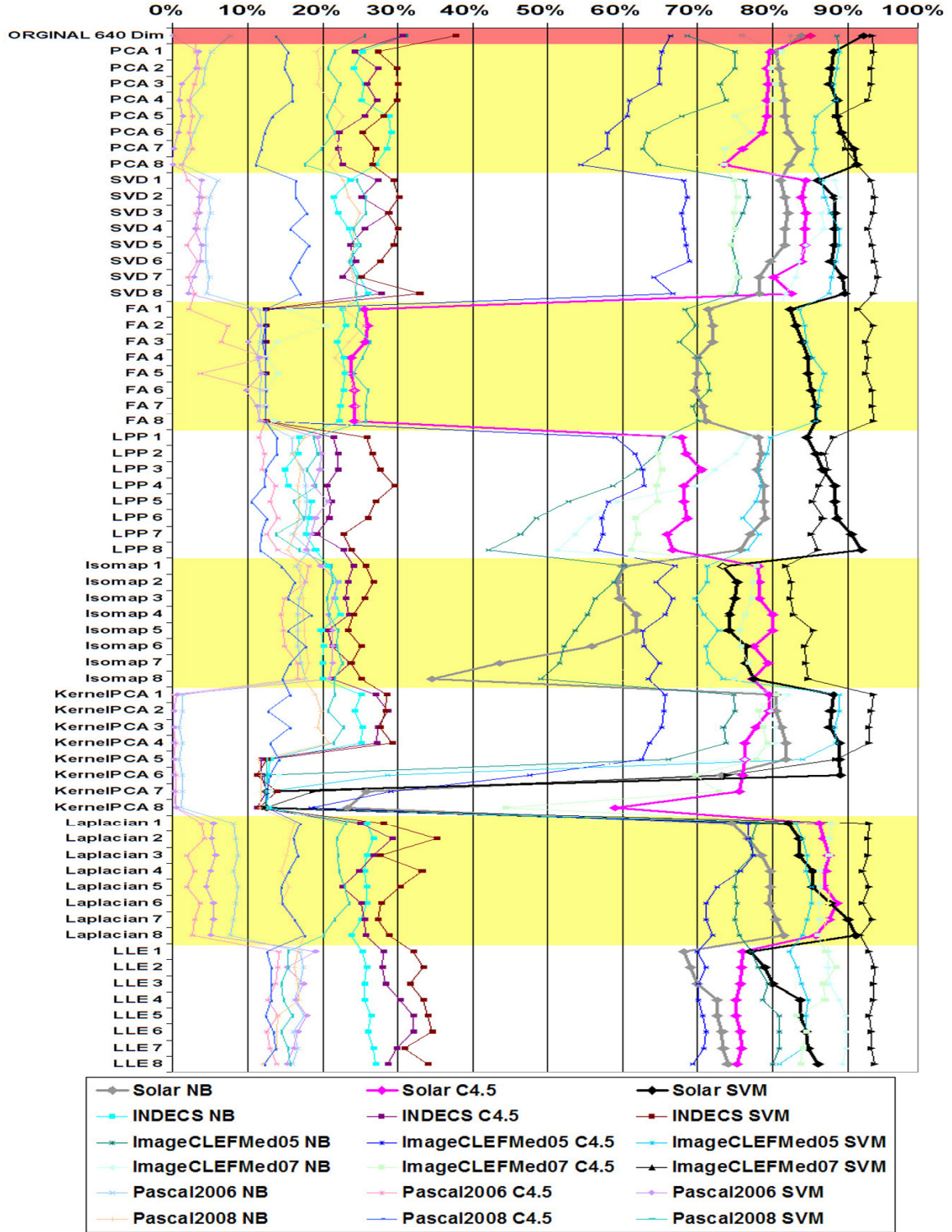


Figure 8.1. Classification accuracy of all dimensionality reduction methods for eight sets of target dimensions for each of the eight dimensionality reduction methods, NB stands for Naïve Bayes, SVM is Support Vector machines. For exact dimensionality of the new dimensional space for each dataset, refer to table 8.1. Experiments use 67% of the data for training and 33% for testing.

Fig. 8.1 shows classification accuracy of our three selected classifiers on the original non-reduced data sets and the 64 dimensionally reduced experiments (from table 8.1, it can be seen that we investigated eight sets of target dimensions for each of the eight dimensionality reduction methods). This figure presents the original data in the first row, then the four linear dimensionality reduction methods (PCA, SVD, FA, LPP) and four non-linear ones (Isomap, KernelPCA, Lapacian, LLE), each with their respected eight targeted number of dimensions.

The first observation we can make is that our image parameters produced very bad results (less than 40% accuracy) for all non-medical and non-solar datasets (INDECS, PASCAL 2006 and 2008) that contain images very different from our own dataset (Solar). However, they produce very good results (between 70% and 90% accuracy) on our solar dataset and the others that contain images similar to ours (ImageCLEFMed 2005 and 2007). This clearly shows that using the right image parameters for a specific type of image is very important. We also observe that SVMs produced most of the highest classification percentages for our solar dataset and the ImageCLEFMed datasets.

In terms of the best dimensionality reduction methods for our solar dataset, we observe that LPP-8, PCA-8, and Laplacian-8 achieve the best results for the solar dataset with over 90% classification accuracy. These methods show a constant improvement with the additional dimensions as the experiments progress and we will latter show that the actual peaks for this methods are achieved in experiment number 8, regardless of the number of additional dimensions added leading up to the original 640 dimensions.

With regards to the best performing classification algorithm, figure 8.1 shows that SVM's achieve all the 90% (and higher) classification accuracy values, constantly performing better than NB and C4.5 by a factor of 5% to 10%, except for the Isomap experiments on the solar dataset. In terms of the most stable dimensionality reduction methods, we found that LPP and Laplacian for all three classifiers behave very nicely as we can see in figure 8.1. From this observation we can determine that the best performing linear dimensionality reduction method for our solar dataset is LPP, and the best non-linear method is the Laplacian Eigenmaps (Laplacian).

When compared against the Medical datasets (ImageCLEFMed05 and ImageCLEFMed07) we have that for most cases the performance of the combinations of classification algorithms and dimensionality reduction methods are very consistent, featuring the same dramatic drops for the FA experiments 1-8 and KernelPCA experiments 6-8. As a side product of our evaluation, we have that ImageCLEFMed07 produced the best overall results with classification accuracy values of up to 95%.

When it comes to the worst performing datasets, we clearly see that the image parameters selected for our solar image data analysis are poor parameter choices for the PASCAL2006, PASCAL2008 and INDECS datasets, and this reflects on classification accuracy values lower than 40% at best. While most of their performance with the linear dimensionality methods is very stable for the majority of the classifier/method combinations, this is a very un-useful result since these classification accuracy numbers are very poor.

An interesting observation we made was that some combinations of classifiers and dimensionality reduction methods (e.g. Laplacian and C4.5) actually produced slightly better results than our original non-reduced dataset. We also see that a few of these combinations produced very bad results (C4.5 and Factor Analysis) and that others dramatically drop in accuracy (KernelPCA for all classifiers after a certain number of dimensions). Figures 8.2 through 8.4 demonstrate the previously mentioned behaviors of our solar dataset. We present detailed classification accuracy results by performing a brute force analysis where we applied the dimensionality reduction methods with target dimensions from 1 through 639 dimensions.

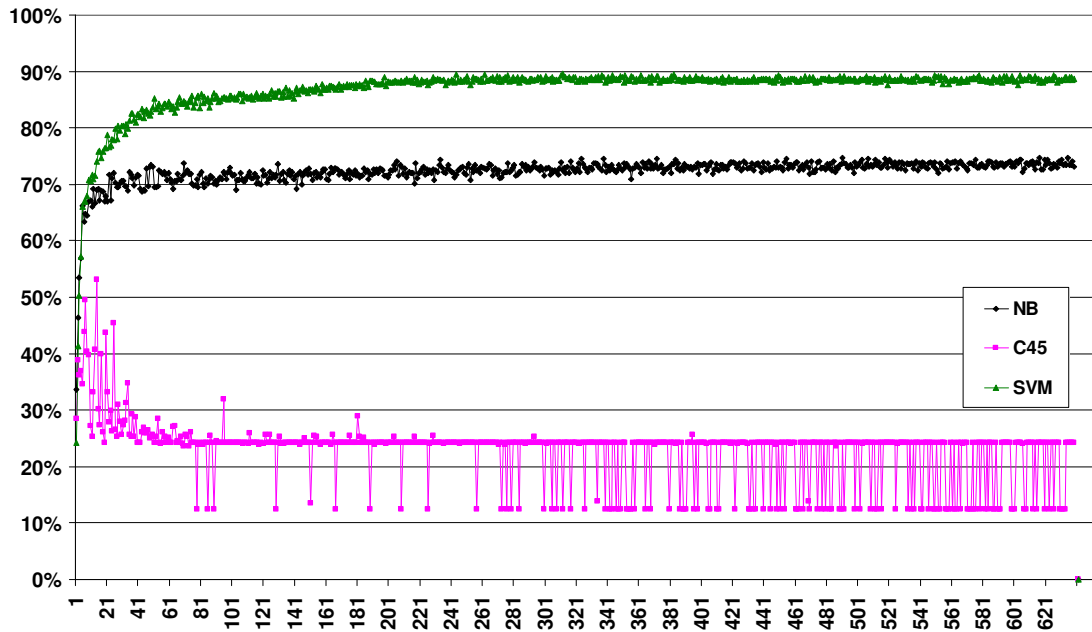


Figure 8.2. Classification accuracy for all Factor Analysis dimensions for the solar dataset. Experiments use 67% of the data for training and 33% for testing

As shown in figure 8.2, the tree based classifier performs very poorly with the Factor Analysis (FA) generated space by making very bad splitting decisions. Since C4.5 is a greedy algorithm (it never re-evaluates its choice of attributes) it results in the

accuracy dropping to 12%, which is equal to a random label assignment since we have 100% over eight classes. We can conclude that if researchers are using FA for dimensionality reduction, it might be a good idea to stay away from decision tree based classifiers.

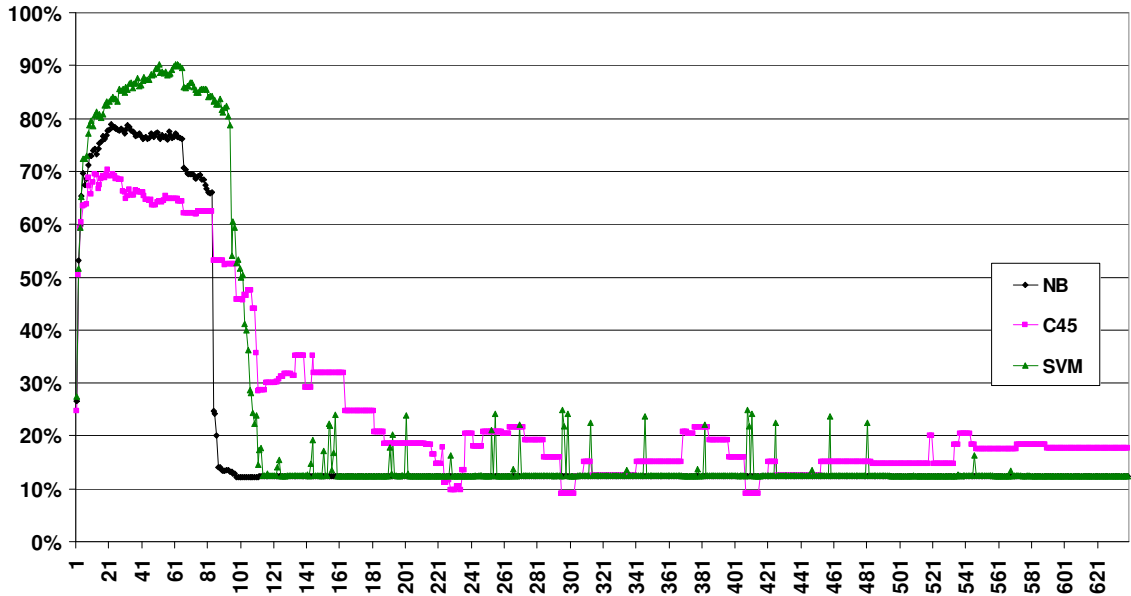


Figure 8.3. Classification accuracy for all KernelPCA dimensions for the solar dataset. Experiments use 67% of the data for training and 33% for testing

After looking at the drops in figure 8.1, KernelPCA showed the most dramatic decrease in accuracy for a higher number of dimensions (experiments 6 to 8). In figure 6, we clearly show this behavior. After a certain number of dimensions, the classification accuracy drops by almost 60% on average. This 12% accuracy is again equivalent to random labeling. We attribute this drop to the partitioned space by Kernel PCA. In a low number of dimensions this method achieves very good representation of our original data space, however, when more dimensions are used, the method just produces more noise for the classifiers and damages their performance considerably. Being a non-linear

method, it is surprising that NB almost achieves 80% accuracy for very few components and that it is also the first classifier to achieve low classification accuracy numbers. We think that this is due to the simple NB model benefiting by the reduced dimensional space. When the dimensionality of our space starts increasing in size, the classifier's accuracy quickly starts dropping. C4.5 takes the longest to hit bottom, showing more resistance to the added noise from the extra dimensions. SVM peaks very quickly with around 60 dimensions and resists the added noise for a bit longer than NB, but it also drops after 200 dimensions.

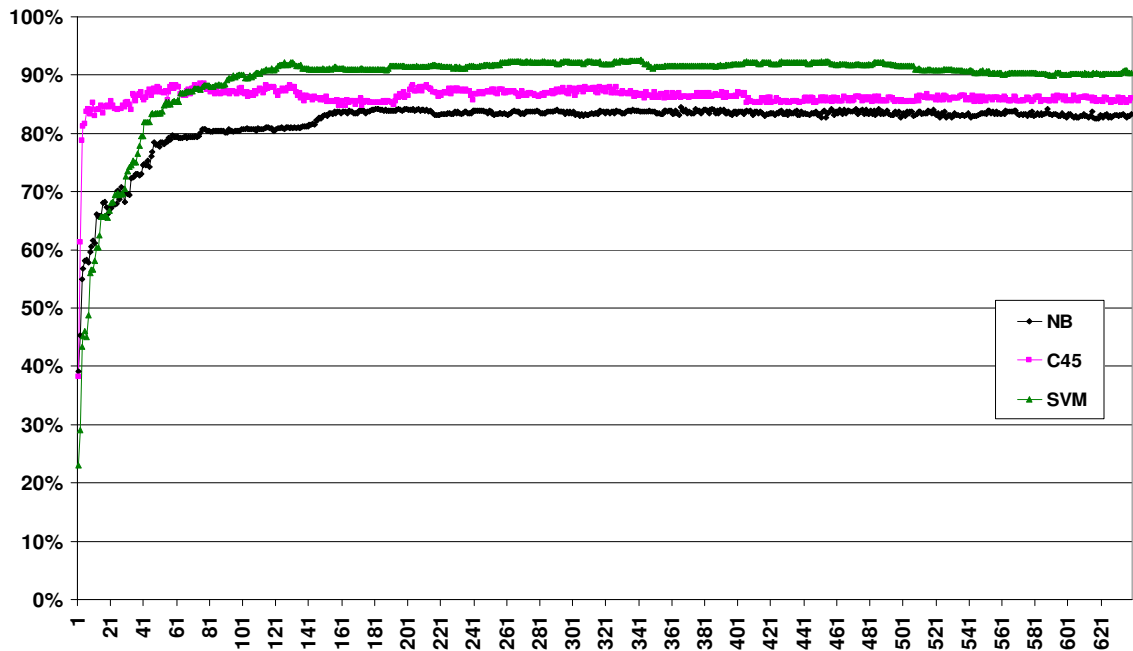


Figure 8.4. Classification accuracy for all Laplacian dimensions for the solar dataset. Experiments use 67% of the data for training and 33% for testing

Laplacian Eigenmaps (Laplacian) offer the most stable results we have seen when applied to our Solar dataset. For all classifiers this dimensionality reduction method provides over 80% accuracy and for SVMs it stays over 90% for the majority of fig. 8.4. The difference between the accuracy of each method is, on average, less than 10%. This

supports the claim of how stable this data representation is. More dimensions, on average, enable an increase in accuracy (especially for NB). But after a certain threshold (140 components) the gain is minimal and not worth the extra computational expense and storage costs.

As we have mentioned, in this chapter we are focusing on achieving dimensionality reduction in order to reduce storage costs. So if we have to sacrifice less than 3% in accuracy (especially at the 90% level) for more than 30% in storage, we are willing to take the hit. Interestingly enough, the tree based classifier for this method quickly increases accuracy with very low number of dimensions. This makes us believe that the Lapacian method is the best for discovering low-dimensional data-spaces when it comes to solar data. This also goes to show that it is very important to analyze how each classifier/dimensionality reduction method behaves before going on and applying whatever you think is the best to your dataset.

As we can see from the upper half of fig. 8.1, the linear methods performed very predictably in terms of stability when the number of components increased. We can say that SVM is the most stable and best performing classification method we utilized. Most of the non-linear dimensionality methods allow the classifiers to perform more consistently between them (bottom half of fig. 8.1). Even when doing badly, the classifier's accuracy stays, on average, within 10% of each other. This better suits these dimensional spaces for classification (on average) than the linear ones. We also show that our way of selecting the number of components provides results on both sides of the fence for non-linear methods.

Out of all of the non-linear dimensionality reduction methods presented in figure 8.1, Laplacian and LLE are the only ones to show constant improvement when compared to linear methods. We theorize that since Laplacian preserves the properties of small neighborhoods around the data points; our benchmark dataset is benefited since many of our data points are highly correlated to each other. This allows Laplacian to preserve their local properties better. We find this very interesting considering that other works have shown that local dimensionality reduction techniques (i.e. Laplacian, LLE), suffer from the intrinsic high dimensionality of the data [78, 142] and these types of methods perform poorly on certain datasets.

In general, as we have seen from our experiments and other works [144], the applicability of dimensionality reduction methods is very dependent on the dataset used. For our purposes, we think that the number of dimensions for our Solar dataset are safely determined by PCA and SVD with a variance between 96% and 99%, and that we manage to approach both sides of the peak in classification accuracy in most cases, indicating that PCA approached the peak (highest accuracy) from the left side (low to high) and SVD behaves the opposite way. The complexity of estimating dimensions this way in a much larger dataset might render these two techniques highly expensive since they rely on the calculation of Eigen vectors, but they seem to be accurate enough when compared to running experiments for all possible numbers of dimensions.

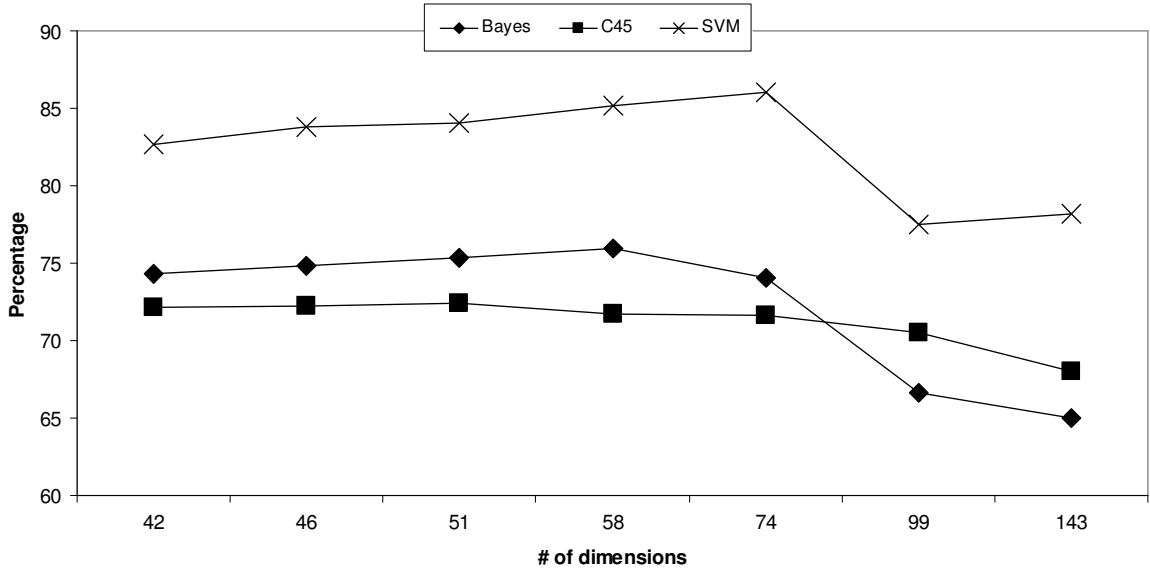


Figure 8.5. Average classification accuracy per number of targeted dimensions for the solar dataset. Experiments use 67% of the data for training and 33% for testing

As a general conclusion for our solar dataset, selecting anywhere between 42 and 74 dimensions provided very stable results for our dataset with all the methods presented in this paper (see figure 8.5). We conclude that for our current benchmark dataset we will be able to reduce our dimensionality around 90% from the originally proposed 640 dimensions. Considering that for the SDO mission we will have to store around 5.27 Gigabytes of data per day and 10,240 dimensions, a 90% reduction would imply savings of up to 4.74 Gigabytes per day. In table 8.2 we show the top ten classification results for the experiments on our solar dataset. From this table we will select the PCA 6 experiment to show the Precision, Recall, F-Measures and ROC curves for the SVM classification experiment.

Table 8.2. Top 10 Classification Results for our solar dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	ORIGINAL	83.86%	Laplacian 6	88.56%	ORIGINAL	92.12%
Top 2	PCA 7	83.49%	Laplacian 3	87.43%	LPP 8	91.74%
Top 3	PCA 8	82.18%	Laplacian 7	87.43%	PCA 8	91.18%
Top 4	PCA 6	81.99%	Laplacian 4	87.05%	Laplacian 8	90.99%
Top 5	SVD 3	81.99%	Laplacian 5	87.05%	PCA 7	90.81%
Top 6	KernelPCA 4	81.80%	Laplacian 2	86.87%	LPP 7	90.43%
Top 7	KernelPCA 5	81.80%	Laplacian 1	86.30%	Laplacian 7	90.06%
Top 8	PCA 4	81.61%	Laplacian 8	85.74%	SVD 8	89.49%
Top 9	PCA 5	81.61%	ORIGINAL	84.99%	SVD 7	89.31%
Top 10	SVD 2	81.61%	SVD 1	84.43%	PCA 6	89.12%

We selected PCA 6 due to the fact that this experiment only loses 3% of classification accuracy and achieves an 87% dimensionality reduction (PCA 6 has 74 dimensions out of 640). The rest of the experiments can easily be reproduced from the files available at [122], and all of the measures can be calculated from them.

Table 8.3. Precision comparison between the SVM original and the PCA 6 experiments on the solar dataset

Class \ Classifier	SVM Original	SVM PCA 6	Improvement
Active Region	0.9650	0.9850	0.0200
Coronal Jet	0.6364	0.8450	0.2086
Emerging flux	0.9643	0.9850	0.0207
Filament	0.7196	0.8380	0.1184
Filament Activation	0.7486	0.9520	0.2034
Filament Eruption	0.5556	0.9400	0.3844
Flare	0.6788	0.7390	0.0602
Oscillation	0.7302	0.8590	0.1288
Average	0.7498	0.8929	0.1431

Table 8.4. Recall comparison between the SVM original and the PCA 6 experiments on the solar dataset

Class \ Classifier	SVM Original	SVM PCA 6	Improvement
Active Region	0.9650	0.9850	0.0200
Coronal Jet	0.7000	0.8960	0.1960
Emerging flux	0.9450	0.9700	0.0250
Filament	0.7700	0.8510	0.0810
Filament Activation	0.6550	0.8940	0.2390
Filament Eruption	0.6750	0.9400	0.2650
Flare	0.4650	0.7730	0.3080
Oscillation	0.7850	0.8210	0.0360
Average	0.7450	0.8913	0.1463

Table 8.5. F-Measure comparison between the SVM original and the PCA 6 experiments on the solar dataset

Class \ Classifier	SVM Original	SVM PCA 6	Improvement
Active Region	0.9650	0.9850	0.0200
Coronal Jet	0.6667	0.8700	0.2033
Emerging flux	0.9545	0.9770	0.0225
Filament	0.7440	0.8440	0.1000
Filament Activation	0.6987	0.9220	0.2233
Filament Eruption	0.6095	0.9400	0.3305
Flare	0.5519	0.7560	0.2041
Oscillation	0.7566	0.8400	0.0834
Average	0.7434	0.8918	0.1484

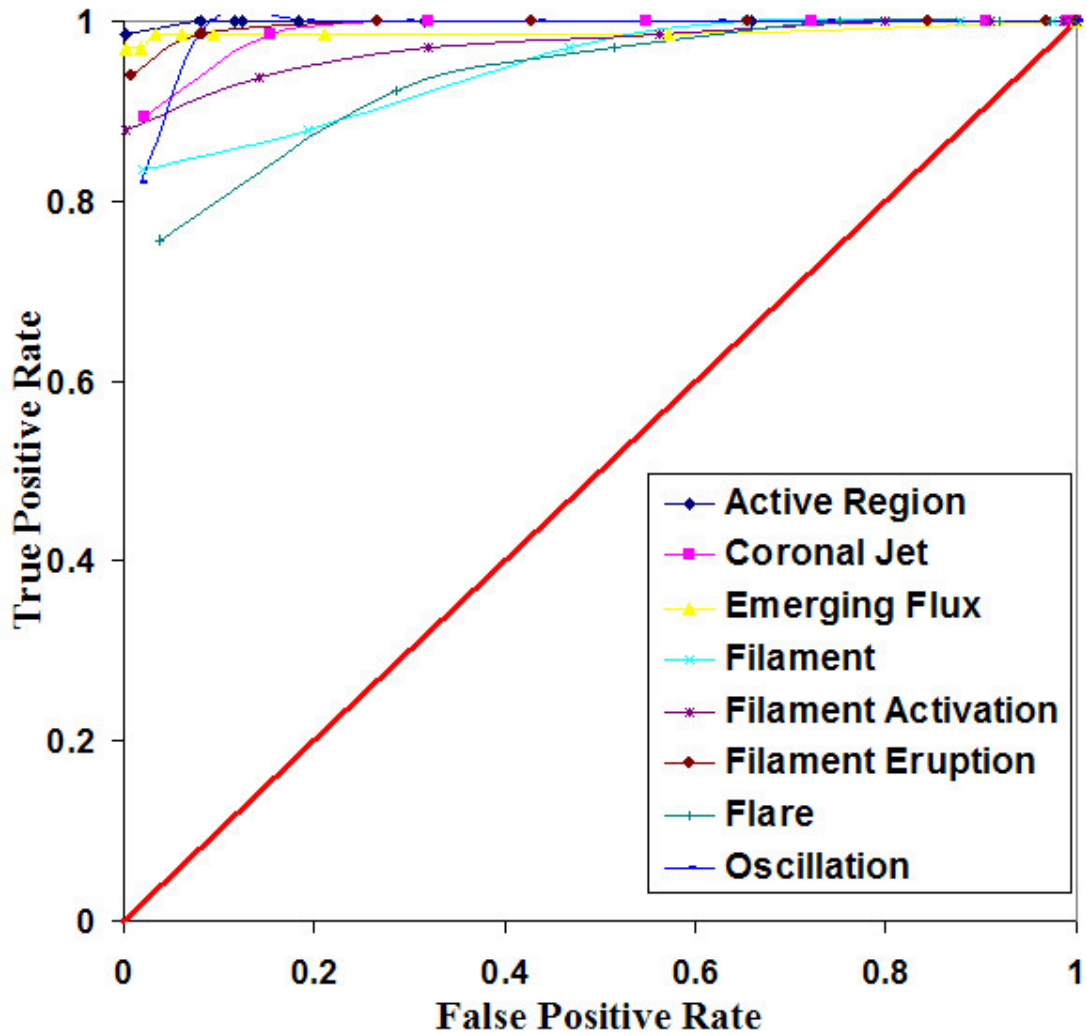


Figure 8.6. PCA 6 experiment SVM ROC Curves for the 8 classes of the solar dataset. Experiments use 67% of the data for training and 33% for testing

For researchers in other fields (i.e. medical images, natural scenes), we present the top 10 classification results for the INDECS, ImageCLEFMED05 and 07, PASCAL 2006 and 2008 datasets in the next few tables. The WEKA files needed to recreate these results and calculate the other measures (recall, precision, etc) can be found at [122]

Table 8.6. Top 10 classification results for the ImageCLEFMED05 dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	LLE 5	80.86%	Laplacian 3	77.49%	PCA 1	88.74%
Top 2	LLE 6	80.86%	Laplacian 1	76.74%	SVD 4	88.74%
Top 3	LLE 7	80.86%	Laplacian 2	76.74%	SVD 5	88.74%
Top 4	LLE 8	79.92%	Laplacian 4	75.61%	KernelPCA 1	88.74%
Top 5	LLE 3	79.36%	Laplacian 5	72.61%	KernelPCA 2	88.74%
Top 6	LLE 4	78.61%	Laplacian 8	72.05%	PCA 2	88.56%
Top 7	LLE 2	78.05%	Laplacian 6	71.11%	ORIGINAL	88.56%
Top 8	Laplacian 1	77.86%	Laplacian 7	71.11%	PCA 4	88.37%
Top 9	Laplacian 3	77.30%	LLE 2	71.11%	SVD 6	88.37%
Top 10	Laplacian 2	77.11%	LLE 6	71.11%	KernelPCA 4	88.37%

Table 8.7. Top 10 classification results for the ImageCLEFMED07 dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	LLE 5	89.68%	LLE 2	88.37%	SVD 7	94.00%
Top 2	LLE 6	89.68%	Laplacian 1	87.62%	SVD 2	93.62%
Top 3	LLE 7	89.68%	Laplacian 2	87.43%	SVD 6	93.62%
Top 4	LLE 8	89.31%	Laplacian 3	87.43%	LLE 2	93.62%
Top 5	Laplacian 3	89.12%	LLE 1	87.05%	LLE 8	93.62%
Top 6	SVD 2	88.74%	LLE 4	86.87%	PCA 1	93.43%
Top 7	LLE 4	88.74%	LLE 3	86.68%	SVD 3	93.43%
Top 8	Laplacian 1	88.56%	Laplacian 4	86.30%	SVD 8	93.43%
Top 9	Laplacian 2	88.56%	Laplacian 5	86.30%	FA 2	93.43%
Top 10	Laplacian 4	88.56%	Laplacian 6	86.30%	FA 8	93.43%

Table 8.8. Top 10 classification results for the INDECS dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	ORIGINAL	30.96%	LLE 5	32.08%	ORIGINAL	37.71%
Top 2	PCA 6	29.08%	LLE 6	32.08%	Laplacian 2	35.08%
Top 3	PCA 5	28.89%	ORIGINAL	30.58%	LLE 6	34.52%
Top 4	PCA 7	28.52%	LLE 4	30.39%	LLE 5	33.96%
Top 5	PCA 8	27.02%	LLE 7	29.83%	LLE 8	33.96%
Top 6	LLE 8	27.02%	Laplacian 2	29.27%	LLE 2	33.40%
Top 7	Laplacian 2	26.83%	SVD 3	28.89%	LLE 4	33.40%
Top 8	LLE 7	26.83%	KernelPCA 2	28.71%	Laplacian 4	33.21%
Top 9	LLE 5	26.45%	LLE 8	28.71%	SVD 8	32.83%
Top 10	SVD 8	26.08%	LLE 3	28.33%	LLE 1	32.08%

Table 8.9. Top 10 classification results for the PASCAL2006 dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	LPP 8	18.39%	Isomap 1	18.20%	Isomap 2	21.95%
Top 2	LPP 5	18.01%	Isomap 2	17.82%	Isomap 3	21.39%
Top 3	LPP 4	17.82%	Isomap 7	16.70%	Isomap 5	21.20%
Top 4	LPP 6	17.82%	Isomap 8	16.70%	Isomap 6	21.20%
Top 5	Isomap 2	17.82%	Isomap 3	15.01%	Isomap 7	21.20%
Top 6	Isomap 8	17.82%	Isomap 6	15.01%	Isomap 8	20.83%
Top 7	LLE 5	17.64%	Isomap 5	14.82%	LPP 5	20.64%
Top 8	LPP 3	17.45%	Isomap 4	14.45%	Isomap 4	20.64%
Top 9	LLE 2	17.45%	LLE 1	14.26%	LPP 2	19.70%
Top 10	Isomap 7	17.26%	LPP 6	14.07%	Isomap 1	19.70%

Table 8.10. Top 10 classification results for the PASCAL2006 dataset for all three classifiers

Classifier	NB		C45		SVM	
Top 1	FA 2	25.52%	Isomap 4	18.57%	FA 3	26.27%
Top 2	SVD 3	24.95%	SVD 5	18.20%	FA 6	26.08%
Top 3	FA 8	24.95%	SVD 3	18.01%	SVD 2	25.70%
Top 4	SVD 8	24.58%	Isomap 6	17.82%	SVD 3	25.70%
Top 5	FA 7	24.58%	Laplacian 8	17.64%	FA 7	25.70%
Top 6	SVD 5	24.20%	SVD 8	17.07%	FA 8	25.70%
Top 7	FA 6	24.20%	Laplacian 1	17.07%	ORIGINAL	25.52%
Top 8	SVD 4	24.02%	SVD 6	16.89%	FA 4	25.33%
Top 9	SVD 7	24.02%	Laplacian 3	16.70%	SVD 5	24.95%
Top 10	SVD 2	23.45%	SVD 1	16.51%	FA 1	24.77%

CHAPTER 9

INDEXING AND RETRIEVAL

Introduction

Once we have carefully selected our image parameters and determined the amount of dimensionality reduction we can achieve; we now need to find an underlying indexing mechanism that will allow us to efficiently store and query our image parameter feature vectors. While this might sound like a trivial task, it is actually one of the most difficult steps in the process of building a CBIR system. In our framework we included six different indexing algorithms that have been widely used by researchers in the fields of CBIR and database systems. As we have seen in the previous chapters, there is not a universal solution for all types of datasets and comprehensive analyses of the indexing methods are needed. The framework proposed in this dissertation allows for experimentation with several state of the art indexes before deciding on which one to implement. We focus the majority of our discussion on the experimental results from solar data, for which we have chosen the single-dimensional indexing algorithms for multi-dimensional data as the best fit. This is mainly due to the fact that our original feature vector has about 640 dimensions, and we will have over 10,000 dimensions for the SDO data. With such dimensionality, we present some of the first, never published before, indexing results for this large number of dimensions.

One of the novel aspects of this work is that we also present the option for the user to take advantage of the dissimilarity measure component in the indexing and query aspect of their CBIR system. With a domain-specific preference for different measures

when it comes to indexing, such as: GIS systems use euclidean-based measures, text-based systems use cosine-based measures, color-photography use the KLD measure, to our knowledge, we present the first tool that enables a comparative analysis of indexing with different dissimilarity measures. With this novelty, we will present very interesting results that will showcase how some different dissimilarity measures perform in our solar dataset and the remaining five experimental datasets we analyzed. We will also take advantage of our dimensionality reduction analysis presented in chapter eight and we will take a look at the best performing dimensionality reduction methods in the context of indexing and retrieval.

We start the discussion of this chapter with a brief description of the indexing algorithms utilized in our framework, and we conclude with an extensive discussion about our experimental results for all datasets proposed in chapter 3. We make a clear separation between multi-dimensional indexing algorithms that index data points in true n -dimensional spaces, versus single-dimensional indexing algorithms for high-dimensional data that map high-dimensional data points in different ways to be able to index them using single-dimensional indexing structures.

Multi-Dimensional Indexing

In this section we present four of the most commonly utilized Multi-Dimensional Indexing algorithms. These algorithms offer great advantages for 2 and 3 dimensional objects but tend to get very inaccurate and computationally expensive to search through as the dimensionality increases [30, 160].

R-trees [47]

R-trees are indexing data structures that are similar to B-trees, but are used for indexing multi-dimensional information in GIS systems. R-trees split space with hierarchically nested, Minimum Bounding Rectangles (MBRs). Overlappings between regions in different branches is allowed, but this causes the search performance to deteriorate, which is a problem that becomes more evident as the number of dimensions increases [161]. The region description of an MBR comprises, for each dimension, a lower and an upper bound. Thus, 2D floating point values are required. An example of an R-tree is presented in figure 9.1.

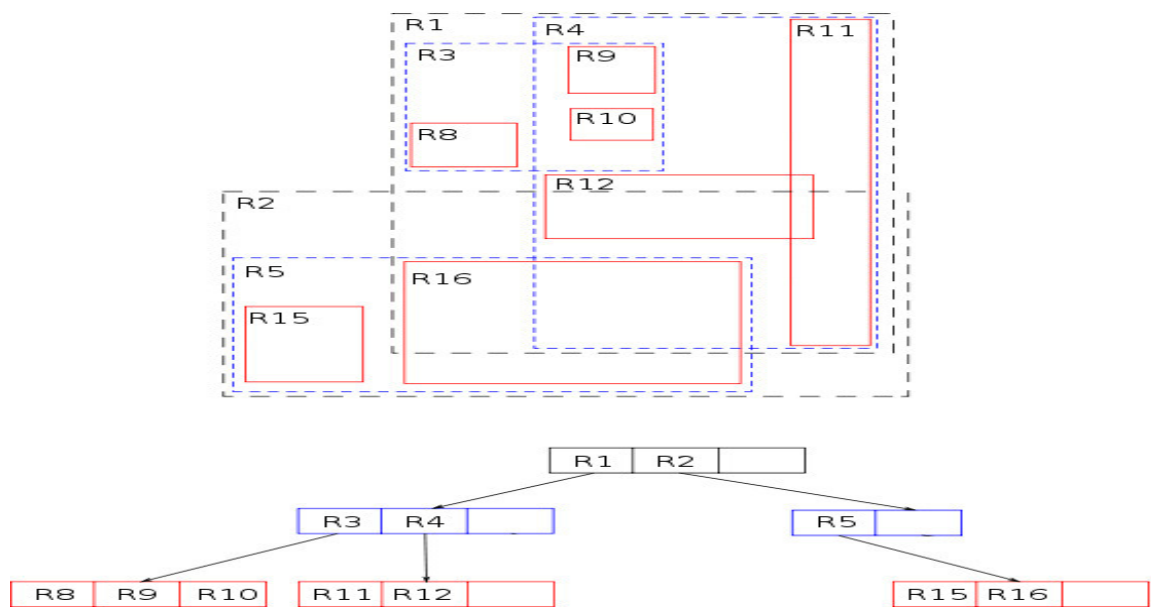


Figure 9.1. 2D R-tree example showing the MBR's and their tree organization, adapted from [162]. Dotted lines indicate nodes in the tree and solid lines represent the MBR's of actual data objects that have references to them placed in tree's leaves

Due to this overlapping problem, several improvements over the original R-tree algorithm have been proposed in order to address these performance issues. We will

discuss two of these improvements (R+-trees and R*-trees) and their advantages in the next two sections (9.2.2 and 9.2.3).

R+-trees [163]

Generally speaking, the R+-tree [163] is an overlap-free variant of the R-tree. In order to guarantee no overlap, the splitting algorithm is modified by a forced-split strategy. In this strategy, the child pages, which are an obstacle in overlap-free splitting of some pages, are simply cut into two pieces at a particular position. This might require that these forced splits must be propagated down until the data page level is reached, allowing the number of pages to exponentially increase from level to level. This extension of the child pages is not much smaller than the extension of the parent page if the dimensions in the data are sufficiently high.

By allowing many forced split operations, the pages which are subject to a forced split, are split, resulting in a page utilization of less than 50%. The more forced splits required, the more the storage utilization of the complete index will deteriorate.

R*-tree [48]

The R*-tree [48] algorithm attempts to reduce both coverage and overlap problems by using a combination of a revised node split algorithm and the concept of forced re-insertion at node overflow. Based on the observation that R-tree structures are highly susceptible to the order in which their entries are inserted, an insertion-built structure is not likely to be optimal. The deletion and re-insertion of entries allows them to locate a position in the tree that may be more optimal than their original location.

When a node overflows, a portion of its entries are removed from the node and re-inserted into the tree. To avoid an indefinite cascade of reinsertions caused by subsequent node overflow, the reinsertion routine may be called only once in each level of the tree when inserting any new entry. This produces more well-clustered groups of entries in nodes and reduces the node coverage. Actual node splits are often postponed, causing the average node occupancy to rise. Re-insertion can be seen as a method of incremental tree optimization triggered on node overflow. This provides a significant improvement over previous R-tree variants, but the overhead comes because of the reinsertion method.

kd-tree [49]

The kd-tree, being a special case of BSP (Binary Space Partition) trees, it organizes points in a k -dimensional space in a binary tree in which every node is a k -dimensional point. Every non-leaf node is determined by implicitly generating a splitting hyper-plane that divides the space into two subspaces. The points found to the left of this hyper-plane generate the left sub-tree of that node and the points right of the hyper-plane generate the right sub-tree. In order to determine the direction of the hyper-plane, we have that every node in the tree is associated with one of the k -dimensions which are associated with the hyper-plane perpendicular to that dimension's axis. This determines that the hyper-plane is directed by the value of said point. In other words, if we select a split on the y -axis, all points with smaller y values go to the left sub-tree and equally all points with larger y values go to the right sub-tree. This results in the hyper-plane being set by this y -value. An example of a two-Dimensional kd-tree is given in figure 9.2.

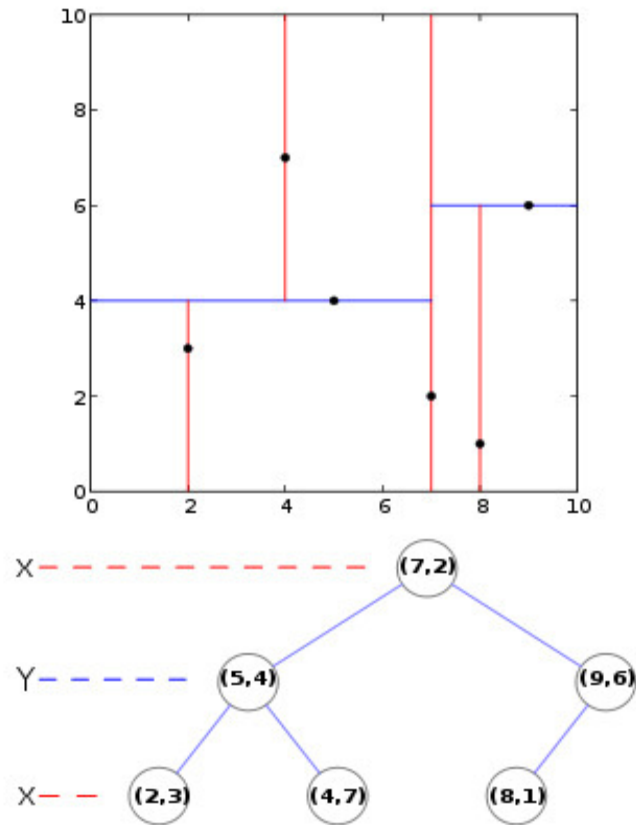


Figure 9.2. 2-D kd-tree example showing the partitions in red and blue, taken from [164]

There are many different ways to choose the axis-aligned splitting hyper-planes, however, the standard method for constructing kd-trees has the following constraints:

- When moving down the tree, cycling through the axes is required in order to select the splitting planes.
- The data points are inserted choosing the median of the points being inserted in the sub-tree, while taking into consideration their coordinates in the axis being used to create the splitting hyper-plane.

These two constraints allow the generation of a balanced kd-tree, in which each leaf node is about the same distance from the root node. The major draw-back from kd-trees is that they are not well suited for efficient nearest neighbor querying in higher

dimensional spaces (over 4 dimensions). Therefore, their performance quite similar to performing an exhaustive search [165].

Empirical analyses [161, 166] show that R-trees, R+-trees, and R*-trees, and even kd-trees [165] do not perform well when indexing high-dimensional data sets (i.e. more than four dimensions). The main problem with R-tree-based indexes is that the overlap of the bounding boxes in the directory increases with growing dimension, quickly creating problems when querying in these overlapping bounding boxes. We will find that this as one of the reasons to not test them using our 640-dimensional solar feature vector and as motivation for using the indexing algorithms described in the next section.

Single-Dimensional Indexing for Multi-Dimensional Data

Due to the nature of these ‘low’ high-dimensional indexing algorithms, researchers worked on creating new indexing algorithms that would accommodate higher dimensions and would allow the efficient indexing of real high-dimensional spaces in creative ways and utilize an underlying B+-tree structure in order to achieve great efficiency in terms of storage and computational expense. Before we continue our discussion, we will define what B+trees are and why they are so useful.

B+ Trees are indexing structures that represent sorted data allowing for efficient insertion, retrieval and deletion of records, each identified by a key. The B+ Tree is dynamic and multilevel, having maximum and minimum bounds on the number of keys in each index segment. In contrast to a B-tree, all records are stored at the leaf level of the

tree and only keys are stored in interior nodes. The notion of the B+ Trees was first brought up by Comer [167], as an interesting alternative for B Trees.

Due to the popularity and efficiency of B+ Trees, they have been implemented in many relational database management systems for table indexes, such as: IBM DB2, Informix, Microsoft SQL Server, Oracle 8, PostgreSQL, MySQL and SQLite.

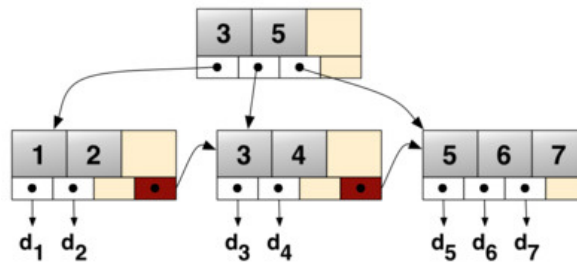


Figure 9.3. B+ tree example, the tree links the keys 1-7 to data values d1-d7. The linked list (red) allows rapid in-order traversal. Image from [168]

B+ Trees have been mainly utilized to index one-dimensional data; however, many of the high-dimensional indexing techniques employ B+ Trees after they have broken down the dimensionality of the dataset into segments that can be indexed by a B+ Tree [50, 52, 168, 170, 171, 240].

iDistance [50, 52]

iDistance is an indexing and query processing technique for k-nearest neighbor (kNN) queries on data points in multi-dimensional metric spaces. The algorithm is designed to process kNN queries in high-dimensional spaces efficiently and it is especially good for skewed data distributions, which usually occurs in real-life data sets.

In order to build the iDistance index, there are two steps involved:

- A certain number of reference points in the original data space are chosen by using cluster centers as reference points [50, 52].
- The distance between a data point and its closest reference point is calculated (using the Euclidean distance in the original implementation), and this distance plus a scaling value is then called the iDistance.

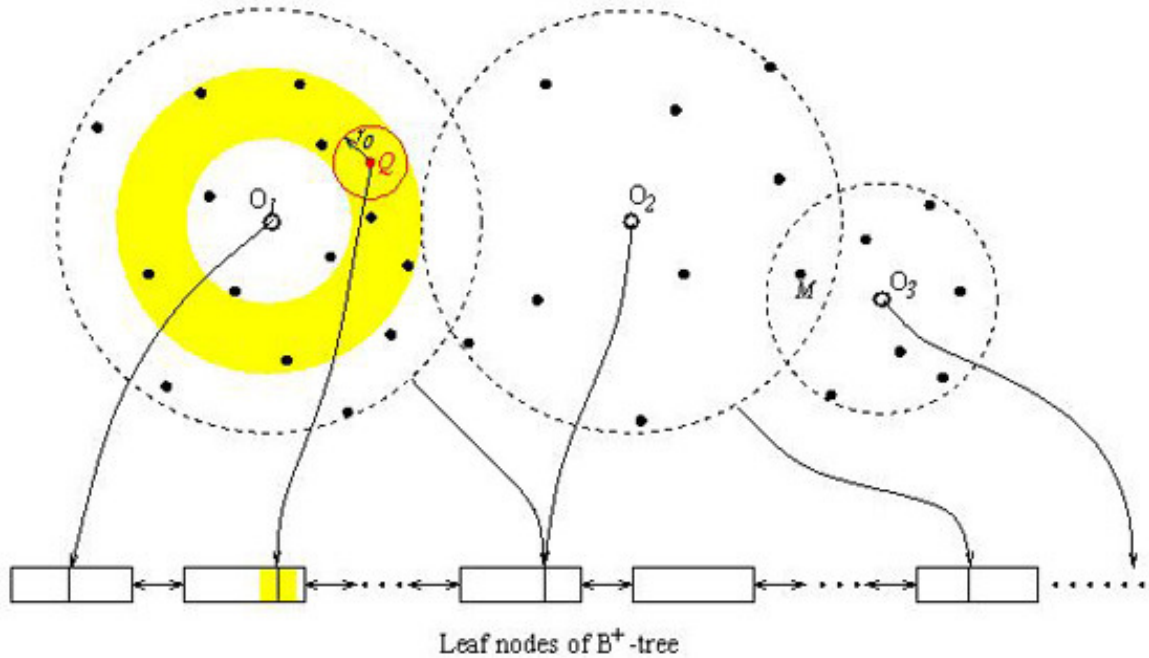


Figure 9.4. Mapping of points via iDistance to a B+-tree. Figure from [52]

In this way, we observe that points in a multi-dimensional space are mapped to one-dimensional values allowing us to use a B+-tree structure to index these points using the iDistance as the key.

To process a kNN query, it is first mapped to several one-dimensional range queries that can be processed efficiently on a B+-tree. A query Q is mapped to a value in the B+-tree while the kNN search sphere is mapped to a range in the B+-tree. The search sphere expands gradually until the kNN's are found corresponding to gradually

expanding range searches in the B+-tree. This iDistance technique can be viewed as a way of accelerating the sequential scan since instead of scanning records from the beginning to the end of the data file, iDistance starts the scan from places where the nearest neighbors can be obtained early with a very high probability.

The authors of iDistance employed the Euclidean distance to calculate the reference point mappings, in our framework we propose a key addition to this algorithm: the usage of our dissimilarity evaluation module. This allows researchers to try 17 different measures with this algorithm and observe how some of their conclusions of the dissimilarity evaluation of their attributes translate to the indexing and retrieval side of things.

Pyramid-trees [240]

Pyramid-trees map a d -dimensional point into a 1 -dimensional space and use a B+-tree to index the one-dimensional space. In Pyramid trees, queries have to be translated in a similar fashion, but in the data pages of the B+-tree, Pyramid-trees stores both the d -dimensional points and the one-dimensional key. This allows Pyramid Trees to disregard the inverse transformation, and the refinement step can be performed without look-ups to another file or structure. This mapping is called Pyramid-mapping and it's based on a partitioning strategy that is optimized for range queries on high-dimensional data. The basic idea of this Pyramid-mapping is to divide the data space in a way that the resulting partitions are shaped like peels of an onion, allowing them to be indexed by B+-trees in a very efficient manner.

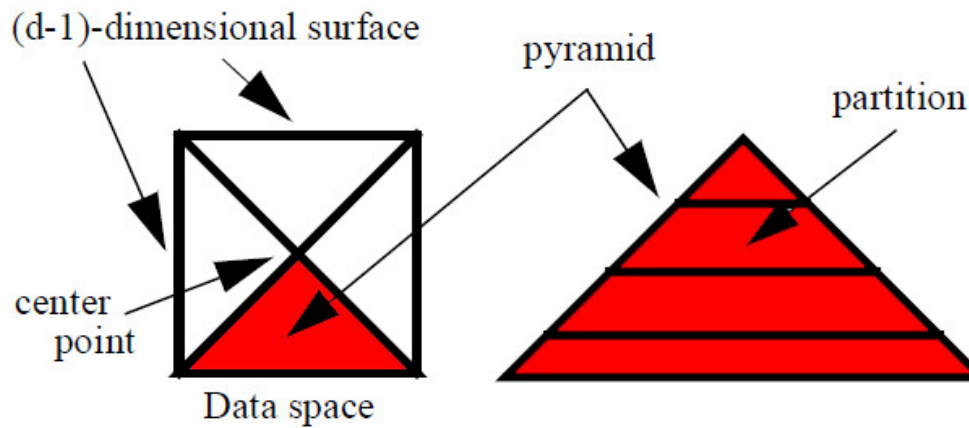


Figure 9.5. Sample breakdown of data points into pyramids. Figure adapted from [240]

Pyramid-trees are one of the only indexing structures known that are not affected by the “curse of dimensionality”, meaning that for uniform data and range queries, the performance of Pyramid-trees gets even better if one increases the dimensionality, and in [240] there is an analytical explanation of this occurrence.

Measures for Evaluation of Retrieval

In order to determine how efficient the indexing algorithms are, we need to query them and measure these results. For these purposes we propose to use a k -Nearest Neighbor (k -NN) search which determines the top k -nearest neighbors to the query. By performing this type of query, we are able to determine how many relevant documents we can retrieve based on our generated index. The following measures are defined to allow us to quantify how effective the index is for k -NN queries based on the image returned. The precision formula is somewhat different from the one defined in chapter five, section five and should not be confused.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \quad (9.1)$$

Precision measures the actual number of correct documents returned by the query, where a precision value of 1 (or 100%) is ideal since it means that all returned documents are relevant to the query.

The other measures we utilize to determine the efficiency of our indexes in terms of querying are:

$$k\text{-}NN\ distance = \text{average distance of the } k\text{-nearest neighbors to the query} \quad (9.3)$$

$$Page\ Reads = \text{number of memory pages of size } n \text{ that are needed to be accessed in order to retrieve each } k\text{-nearest neighbor} \quad (9.4)$$

These two measures will allow us to determine how close the queries are from each other (*k-NN distance*) and how efficient our resulting index is in terms of nodes that need to be accessed to find our queries (*Page Reads*).

Evaluation Results

In the course of this chapter we have presented 6 algorithms for indexing high-dimensional spaces, and in this section we will discuss which algorithms we decided to test on our datasets and discuss some of the most interesting results. We will also present an insight into how some of the other chapters in this dissertation, and modules of our framework, come together.

We begin our discussion by providing rationale into why we decided against using the R-tree based algorithms. As literature shows [161, 166] the R-tree based algorithms do not perform well when indexing high-dimensional data sets because the overlap of the bounding boxes in the directory increases with increased dimensionality, and this quickly creates problems when querying in these overlapping bounding boxes. If we take a look at our 2D and 3D MDS plots in terms of visualization tools, we can see how messy our dimensional space is. As we see in figure 9.6, all of our data points are very close to each other forming a very dense cluster, and this is just shows 2 and 3 dimensions. As the number of dimensions increase, this will become a bigger problem for the R-tree algorithms.

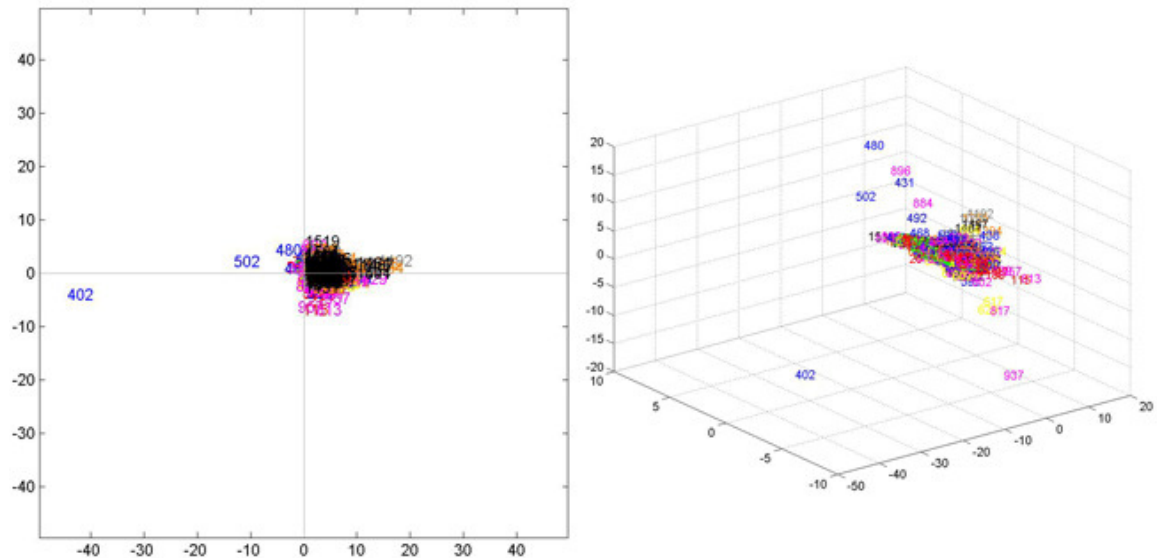


Figure 9.6. 2D and 3D MDS plots that show the cluster formed by our image parameter, this indicates very close similarity between all images of our dataset

With all of these reasons in mind, we will skip experimenting on all 3 types of R-tree algorithms; however, we will present some results for the KD-tree in order to evaluate at least one of the multi-dimensional data indexing algorithms.

As a general setting in our experiments, we treated each of the 1,600 images in our datasets as a query and searched for their 10 Nearest Neighbors in our index. Precision was calculated by determining the number of relevant documents over the number of documents retrieved (10 in this case).

Multi-Dimensional Indexing: kd-tree Experiments

In figure 9.7 we show the retrieval precision values for our 1,600 queries. As we can evidently observe, there are very considerable gaps of precision values of 0 or no relevant documents retrieved.

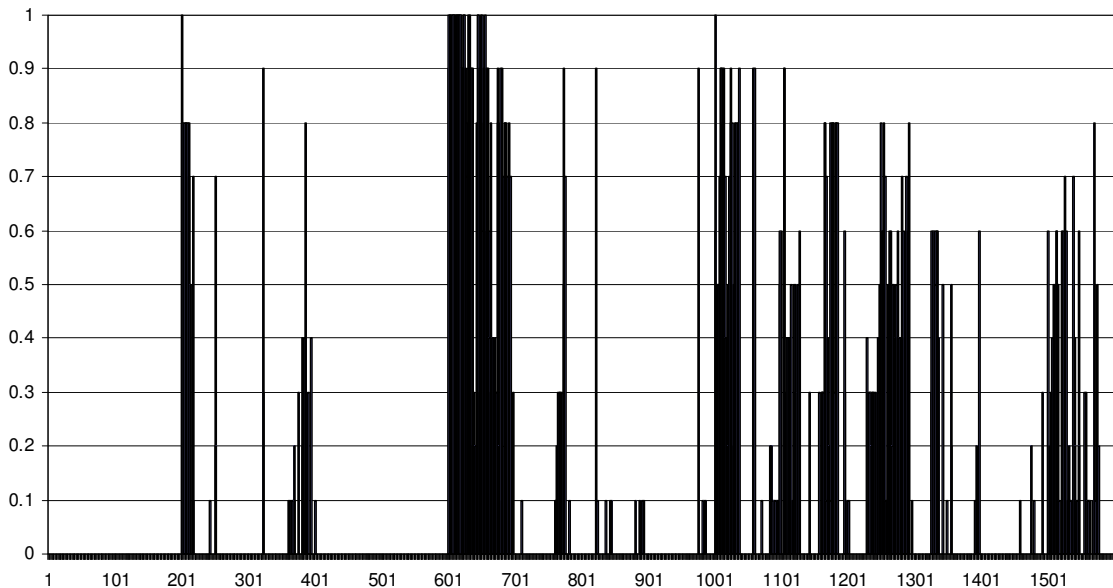


Figure 9.7. kd-tree precision results for each of our 1,600 queries on our solar dataset.
Index built on original 640-dimensional feature vector

The total average precision value for our retrieval on the kd-tree is 0.1114, or in other terms less than 12% accuracy when retrieving with this index. As the literature states and our experimental results show, the kd-trees are clearly not ideal when it comes to indexing a 640 dimensional vector and the retrieval results reflect this. When it comes to the section of queries that have a high precision value (range between 601 and 701 in figure 9.6), we theorize that these queries are the ones closest to our first tree partitions and thus the algorithm returns relevant matches. However, these results when paired with the literature recommendations, this results discourage us from using kd-trees for further analysis on datasets of this high of dimensionality. But note that the kd-trees are still supported in our framework for other researchers using lower dimensional spaces since they might produce good results in those situations.

Single-Dimensional Indexing for Multi-Dimensional Data: Pyramid-tree Experiments

As one of the most promising single-dimensional indexing for multi-dimensional data algorithms, the Pyramid-tree claims to solve the “curse of dimensionality” problem [240] while allowing us to query high-dimensional data in a very efficient manner. To put this claim to the test we will select the 8 different experiments outlined for the dimensionality reduction methods presented in chapter 8 (table 8.1). We will compare these results with the most recent and best performing single-dimensional indexing for multi-dimensional data algorithm: iDistance. For this section we will only use the Euclidean distance since Pyramid-tree only allows the use of this distance. Considerable modifications have to be made to the algorithm to accommodate other distances, but said

modifications are out of the scope of this dissertation and are left for future additions to the framework. We also make clear that we are looking for the best performing algorithm when it comes to indexing and retrieval, so future experiments and analysis beyond this section will be done only on the resulting best performing algorithm. In figure 9.8 we present the number of dimensions for each experiment in the x-axis and the average precision values attained for the 10-NN searches performed on the generated indexes on the y-axis.

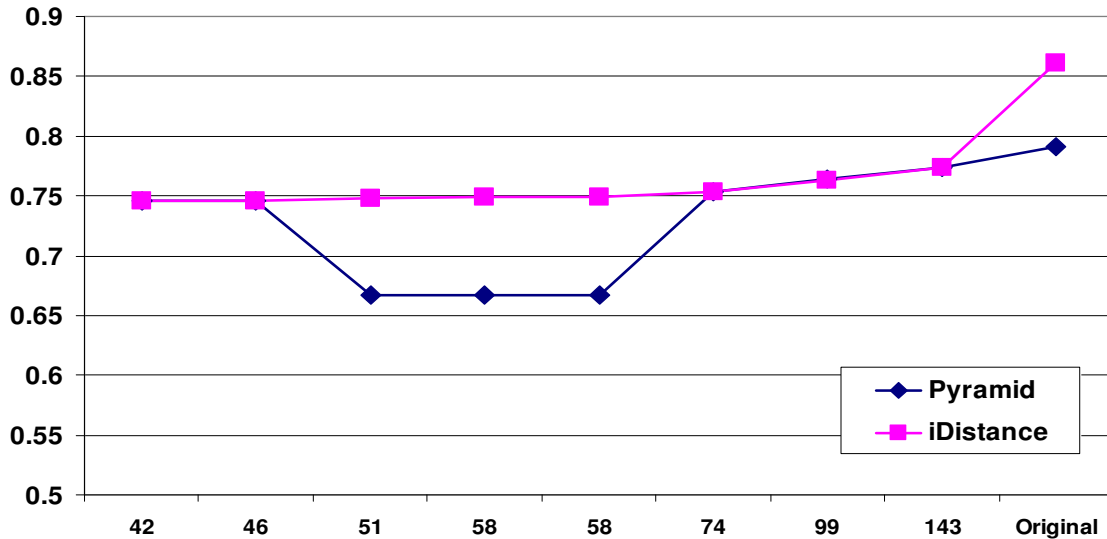


Figure 9.8. Average retrieval precision values for the solar dataset and the pyramid-tree and iDistance indexing algorithms. Indexes built on dimensionally reduced feature vectors and the original feature vector

As figure 9.7 shows, we see that the results for the Pyramid and iDistance algorithms manage to stay very close for 5 out of our 8 experiments. We also see that Pyramid-tree suffers some performance decrease in a few of the targeted dimensions, having a drop of almost 10% (0.10) in retrieval precision, something very considerable when we are trying to find the most stable and best performing indexing algorithm. If we

take into consideration that we are trying to find the best algorithm in terms of handling a high-number of dimension, we see that for the original 640 dimensional dataset the Pyramid-tree comes in behind iDistance by 7% (0.07) accuracy. With this in mind we opted to pursue more experimentation (and to implement different dissimilarity measures) with the best performing algorithm: iDistance. Not only did iDistance retrieval precision values remain very stable, it actually seems to perform very well when the dimensionality increases as seen on figure 9.8. Our framework facilitates experimentation on Pyramid-tree's for future researchers interested in pursuing this algorithm as their indexing mechanism.

Single-Dimensional Indexing for Multi-Dimensional Data: iDistance Experiments

Now that we have determined which single-dimensional indexing for multi-dimensional data algorithm we want to use, we will present our retrieval results for all the dissimilarity measures (table 7.1 in chapter 7) of the dissimilarity measures module implemented for use with the iDistance algorithm. We start our analysis by presenting the retrieval results for all six datasets in their original dimensionality, this will allow us to determine which measures perform well when paired with the iDistance algorithm to be further tested with the dimensionality reduction results of chapter 8.

The labeling on figure 9.9 corresponds to the distance numbers indicated in table 7.1. The D12-13 measures represent the *KLD* measure presented in table 7.1. In our previous experiments we treated the *KLD* measure as a pair of measures ($A-B$ and $B-A$). For iDistance, the measure becomes a one measure since we are only analyzing the

distance from A to B and not the opposite. The other change introduced for our indexing experiments is the elimination of distances D14 and D15, which correspond to the *Minkowski fractional* measure with $p=0.25$ and $p=0.50$. Both of these measures took over 4 hours for index generation, and 16 hours to query 1,600 images making them unreasonable to deploy in a CBIR system that needs to produce results in near real-time.

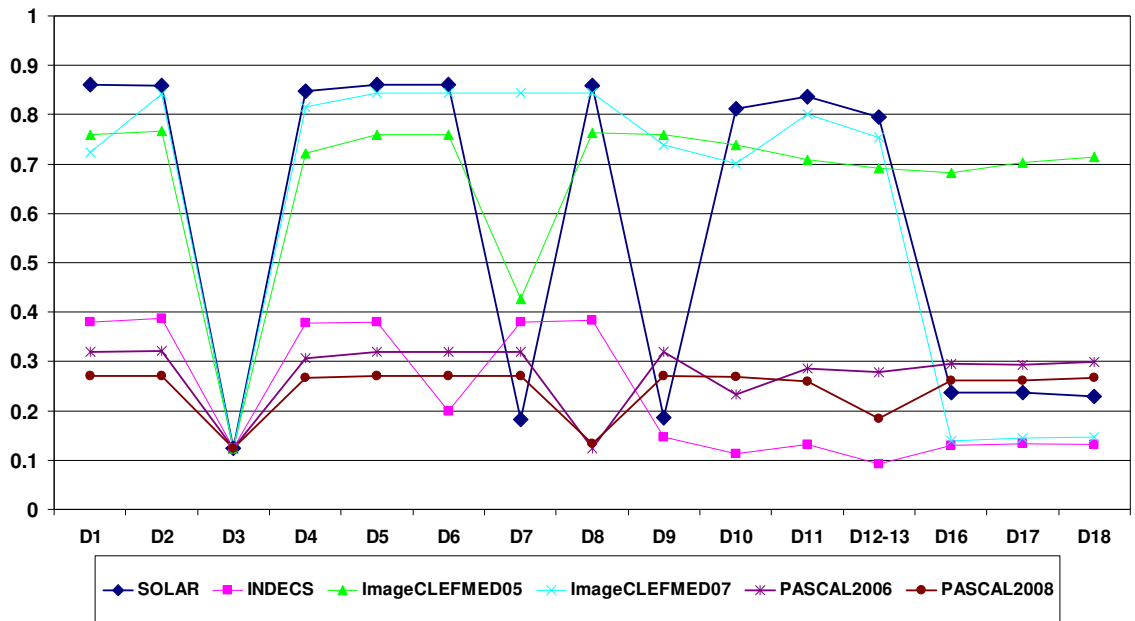


Figure 9.9. Average precision values for the dissimilarity measures implemented on the iDistance algorithm for the 640-dimensional original feature vectors

The results shown in figure 9.9 show very similar trends that our classification results for the dimensionality reduction experiments presented on figure 8.1 in the sense that we have the clusters formed by solar and the ImageCLEFMED05 and ImageCLEFMED07 as the top performing datasets, and then low performing datasets cluster incorporating the INDECS and the PASCAL2006 and 2008 datasets. In this figure we can also observe that most of the dissimilarity measures that perform poorly, do so in a consistent manner for all datasets. These clearly show that our indexing algorithms

produce an effective index that allows for good retrieval results, and when compared to using classifiers we show a great advantage since we are not using a supervised learning approach.

In order to determine the best performing dissimilarity measures for each of the datasets analyzed, we present table 9.1. in which we can observe the top three average retrieval precision results for each dataset and their corresponding dissimilarity measure.

Table 9.1. Top three average precision results for all datasets and dissimilarity measures

Solar		INDECS		ImgCLEFMed05		ImgCLEFMed07		PASCAL2006		PASCAL2008	
D6	0.8612	D2	0.3878	D2	0.7660	D5	0.8438	D2	0.3223	D7	0.2706
D5	0.8611	D8	0.3836	D8	0.7631	D6	0.8438	D1	0.3192	D1	0.2703
D1	0.8609	D5	0.3806	D5	0.7601	D7	0.8438	D9	0.3191	D5	0.2701

As a very interesting observation, we can see that the Euclidean distance (D1) is not the overall best performing measure, validating the need to analyze different alternatives for indexing and retrieval purposes. And while the improvements are in general very small for the datasets we tested (0.3%), these results might change and bigger improvements could be found on other datasets. One thing to mention is that while we had considerable improvements for the KLD (D12-13) measure in chapter 7 for our poorly performing datasets, this is not reflected here due to the fact that we are taking into consideration all 10 image parameters rather than individual combinations of them.

One of the most interesting results of our analysis indicate that the Minkowski based fractional measures actually perform very poorly for almost all datasets. While other researchers have shown that they improve retrieval under certain circumstances [129], we show that for the iDistance algorithm these trends are not followed. We think that this is due to the way that iDistance breaks down the dimensions into segments for

the B+-tree to index. We noticed that as the p gets smaller than one, the distances of the segments increases considerably thus making the search and the retrieval very inefficient. In [53], the authors have shown that by using the *Mahalanobis* measure (D3) they managed to achieve very good results for particular artificial and non-artificial datasets, but in our experimental results we find the opposite. This behavior can again be attributed to the inner workings of the iDistance algorithm and the way it breaks down dimensionality, and another reason is that this distance was not optimal for any of our datasets as we indicated in chapter seven.

Now that we have determined the best dissimilarity measures for the six datasets analyzed, we will combine these results with our dimensionality reduction experiments in order to verify how well the retrieval works for a lower dimensional version of our dataset. In table 9.2 we present the experimental set-up used for the rest of the chapter.

Table 9.2. Experimental set-up for the different dissimilarity measures and iDistance based on the best dimensionality reduction experiments of chapter eight

Dataset	Experiment (dimensions)	Dissimilarity Measure		
		EXP1	EXP2	EXP3
Solar	LPP-8 (143)	D1 (Euclidean)	D6 (Cosine)	D5 (Chebychev)
INDECS	Laplacian-2 (106)	D1 (Euclidean)	D2 (Seuclidean)	D8 (Spearman)
ImgeCLEFMed05	PCA-1 (79)	D1 (Euclidean)	D2 (Seuclidean)	D8 (Spearman)
ImgeCLEFMed07	SVD-7 (251)	D1 (Euclidean)	D5 (Chebychev)	D6 (Cosine)
PASCAL2006	Isomap-2 (85)	D1 (Euclidean)	D2 (Seuclidean)	D9 (Hausdorff)
PASCAL2008	FA-3 (141)	D1 (Euclidean)	D7 (Correlation)	D5 (Chebychev)

Since the standard implementation of iDistance uses the Euclidean distance [50], we present this dissimilarity measure as our first experimental measure (EXP1). For the remaining experiments (EXP2 and EXP3), we selected the top two dissimilarity measures on our indexing and retrieval experiments as presented in table 9.1.. For a direct

comparison to the iDistance's best results we present highest retrieval accuracy values obtained for the original 640-dimensional datasets from all the dissimilarity measures as listed on table 9.1 and we will label them as *Original*.

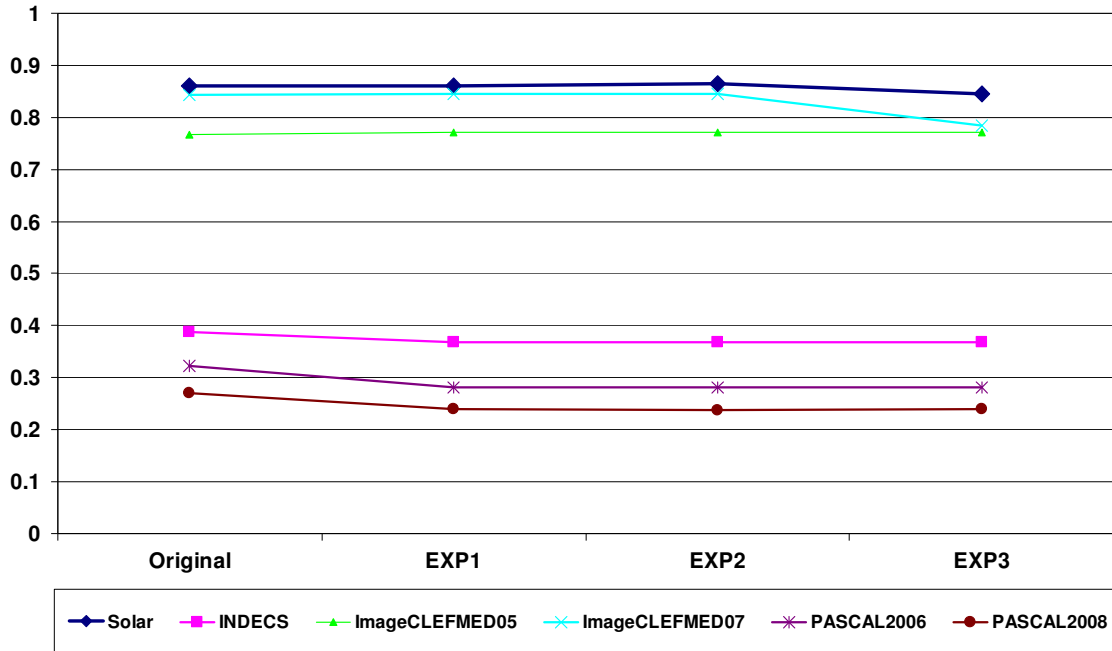


Figure 9.10. Average retrieval precision values for the experimental analysis of the dissimilarity measures outline in table 9.2.

While we do not see any critical improvement on the average retrieval precision in our experimental set up, we have to realize that all our experimental results feature a dimensionality reduction factor of more than half for our ‘biggest’ experiment, which is ImageCLEFMED07 SVD-7 with 251 dimensions. This clearly shows the applicability of dimensionality reduction in an actual indexing and retrieval setting. For our solar dataset, we have a very minimal (less than 1%) improvement for experiments EXP1 and EXP2, and a very small drop (less than 1%) for EXP3. This indicates that even with a 77% of dimensionality reduction, the indexing algorithm creates an index than when queried, returns an average of 86% of the documents retrieved are relevant to the query. Figure

9.10 shows the stability of this indexing algorithm when paired with the correct dissimilarity measures.

In the next figures we will show the average 10-NN distance and the average page reads for our indexing experiments. We present them with the purpose of showing how efficient (or inefficient) these combinations of different dissimilarity measures can be with the iDistance indexing algorithm.

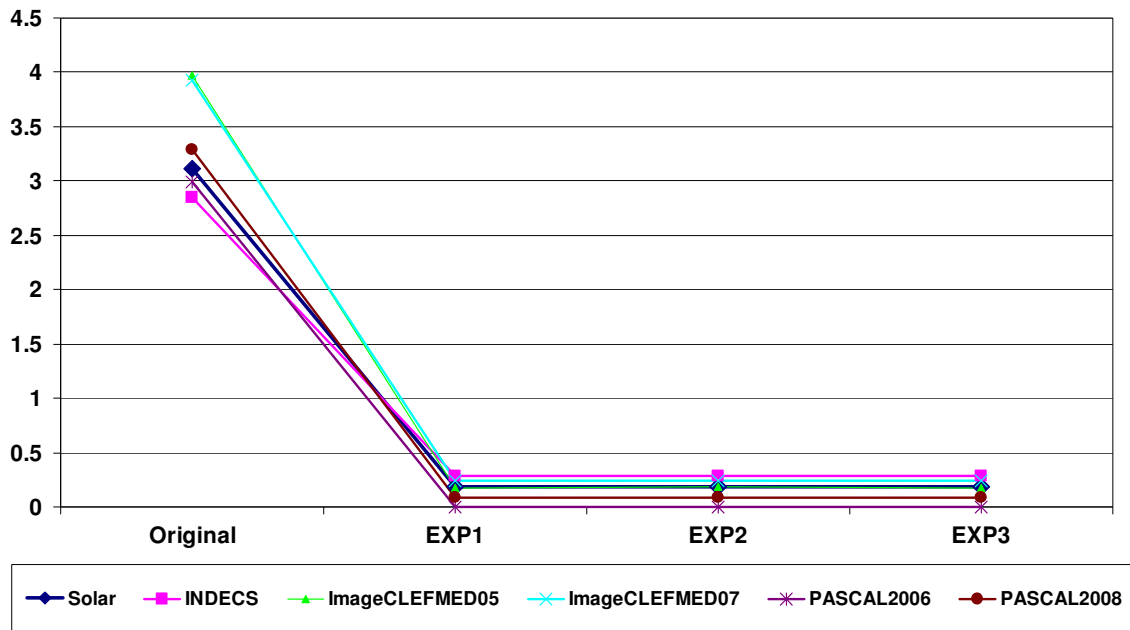


Figure 9.11. Average 10-NN distance for all experiments and all datasets tested compared with the top original non-reduced datasets. Distance and datasets described in table 9.2

Figure 9.11 clearly shows that the average 10-NN distance considerably decreases with the lower dimensionality experiments. We also confirm the very stable results provided by the different dissimilarity measures on the indexing algorithm. In theory, the closest the NN distance is to zero, the stronger it indicates that all the returned results are the actual closest neighbors to the query image. This shows that, for the poorly performing datasets (INDECS, PASCAL2006 and PASCAL2008) the image parameters

and dimensionality reduction methods are not doing a good job in creating feature vectors that are easily discernable from each other, thus achieving very poor precision.

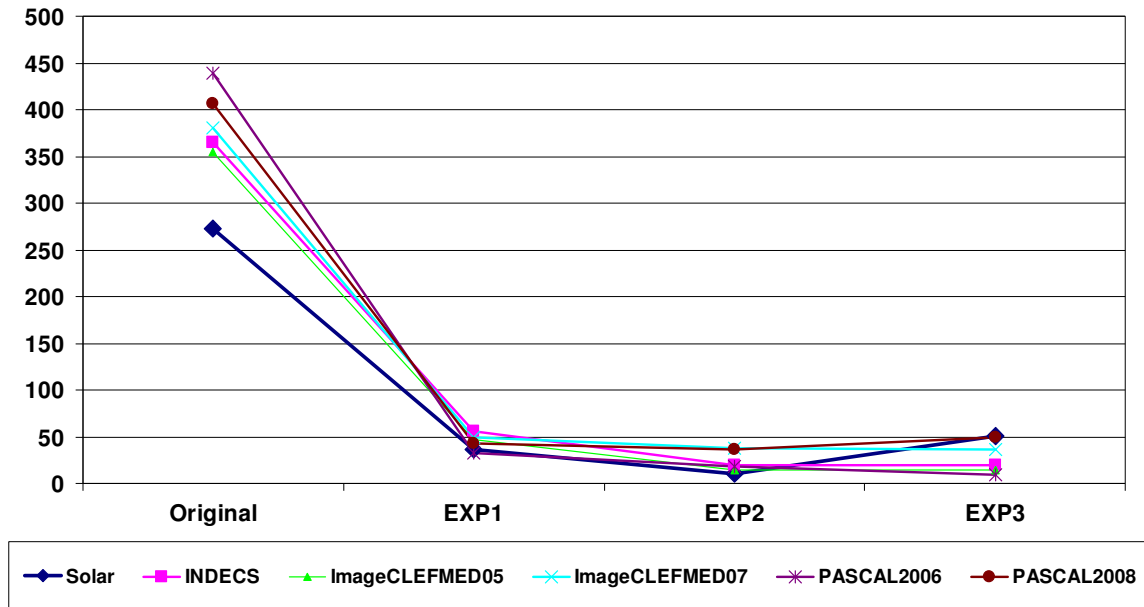


Figure 9.12. Average page reads for all experiments and all datasets tested compared with the original non-reduced datasets. Distance and datasets described in table 9.2

In terms of page reads, figure 9.12 shows that the behavior of fewer page reads for our experiments is not surprising since we are querying a lower dimensional index (fewer data points, fewer pages). But what is valuable here is the fact that we can query the built indexes very efficiently in these low dimensional spaces while still returning around 86% of relevant matches. This translates to a fast responding and efficient retrieval mechanism that will allow the users of the CBIR system to receive relevant results in a timely and efficient manner. We also have that the solar dataset requires fewer page reads for the original dimensionality, EXP1 and EXP2. This indicates that our indexes built for the solar dataset are more efficient than the remaining ones. This behavior can be due to the fact that the image parameter values obtained for the solar

dataset are more efficiently indexed by iDistance, allowing the retrieval to be performed faster while reading fewer pages of the index.

Since we have shown consistent retrieval precision between our original 640 dimensional feature vector and our reduced experimental feature vectors for the solar dataset, we present the figure 9.13, where we analyze our 8 different dimensional targets for the solar dataset (presented in chapter 8, table 8.1) for all dimensionality reduction methods and their performance on our indexing mechanism. We present this evaluation with the purpose of observing the behavior of all of these methods with our indexing algorithm using the Euclidean distance.

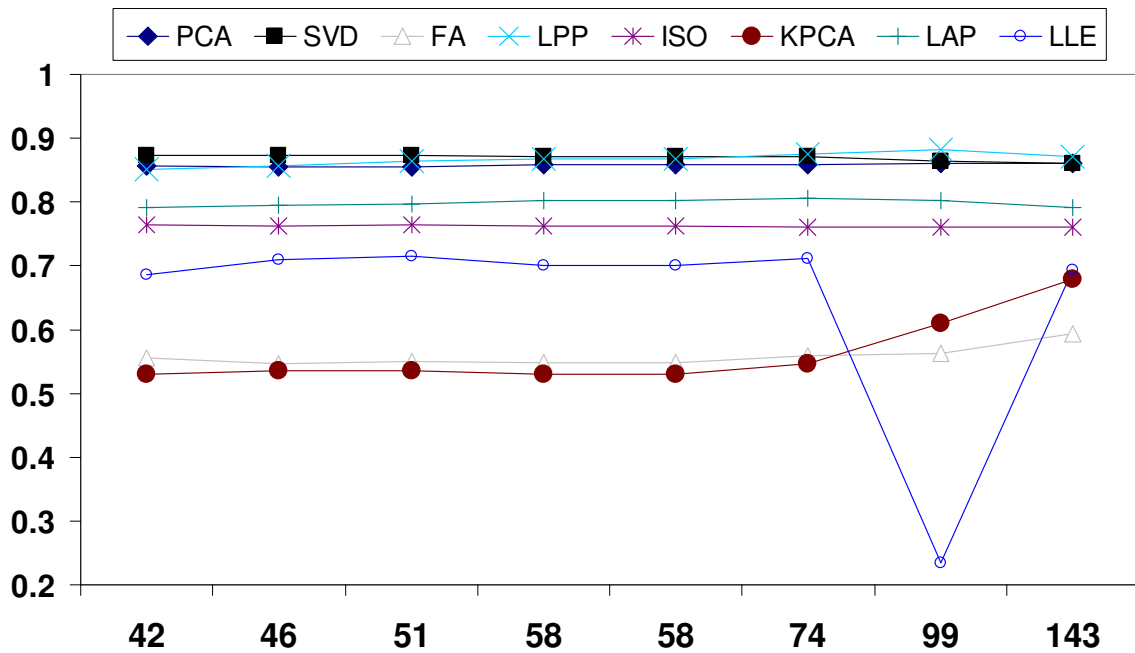


Figure 9.13. Average retrieval precision per number of generated dimensions for the solar dataset using the euclidean distance.

We can observe in figure 9.13 that at least three of our dimensionality reduction methods (SVD, PCA and LPP) are over the 0.86 precision threshold that our original 640-dimensional dataset achieves, this means that the dimensionality reduction provided

by these methods can achieve comparable results on a considerably lower dimensional space, verifying our claim that we can utilize them to reduce storage and retrieval costs significantly.

In this chapter we have presented experimental results that validate our use of iDistance as the single-dimensional indexing algorithm for multi-dimensional data that works the best for our solar dataset. When combining the best dimensionality reduction results with this indexing algorithm, we have demonstrated that our unsupervised evaluation (kNN search on our indexes) of the reduced dimensional feature vectors achieve very similar results to our supervised comparative evaluations presented with the classification algorithms in the previous chapter. This is a very valuable conclusion since we can now determine how much dimensionality reduction we can achieve without considerably affecting the quality of our feature vector while allowing us to save on storage and retrieval costs.

CBIR System - Retrieval Demonstration

Solar Images

For the best performing experiment on our solar dataset: Locality Preserving Projections with 143 dimensions (LPP-8), we will present the demo CBIR system retrieval results for one query image of each of the eight classes available in our solar dataset. We will present the best queries (most similar images) for six classes and we will present poor results for two of the classes for illustrative purposes. Table 9.3 shows the

class names and the image number ranges utilized in our CBIR system retrieval results figures.

Table 9.3. Solar dataset classes and image ranges

Class Label	Image Numbers
Active Region	1 - 200
Coronal Jet	201 - 400
Emerging Flux	401 - 600
Filament	601 - 800
Filament Activation	801 - 1000
Filament Eruption	1001 - 1200
Flare	1201 - 1400
Oscillation	1401 - 1600

Query 87

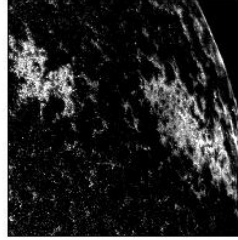


Image 120

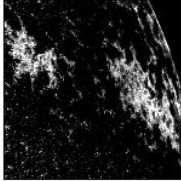


Image 28

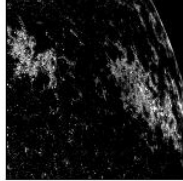


Image 130

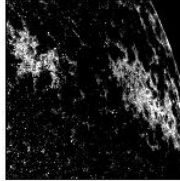


Image 88

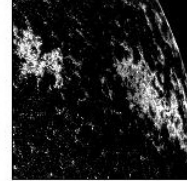


Image 92

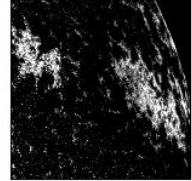


Image 128

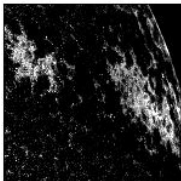


Image 118

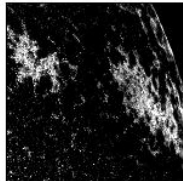


Image 87

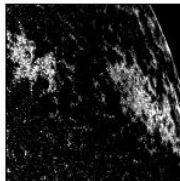


Image 33

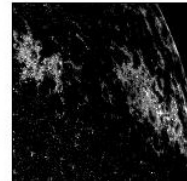


Image 114

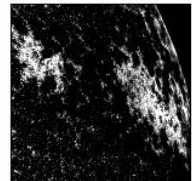


Figure 9.14. Retrieval results for the solar dataset – Active Region class. The query image (#87) is in the top, and the retrieved 10-NN images are in the bottom rows

In figure 9.14 we show the 10-NN results for image 87 as a query. Every image returned with an image number between 1 and 200 (see table 9.3), belongs to the Active Region class. This sample query has a retrieval precision of 1.0 (100%) since all returned images belong to the Active Region class.

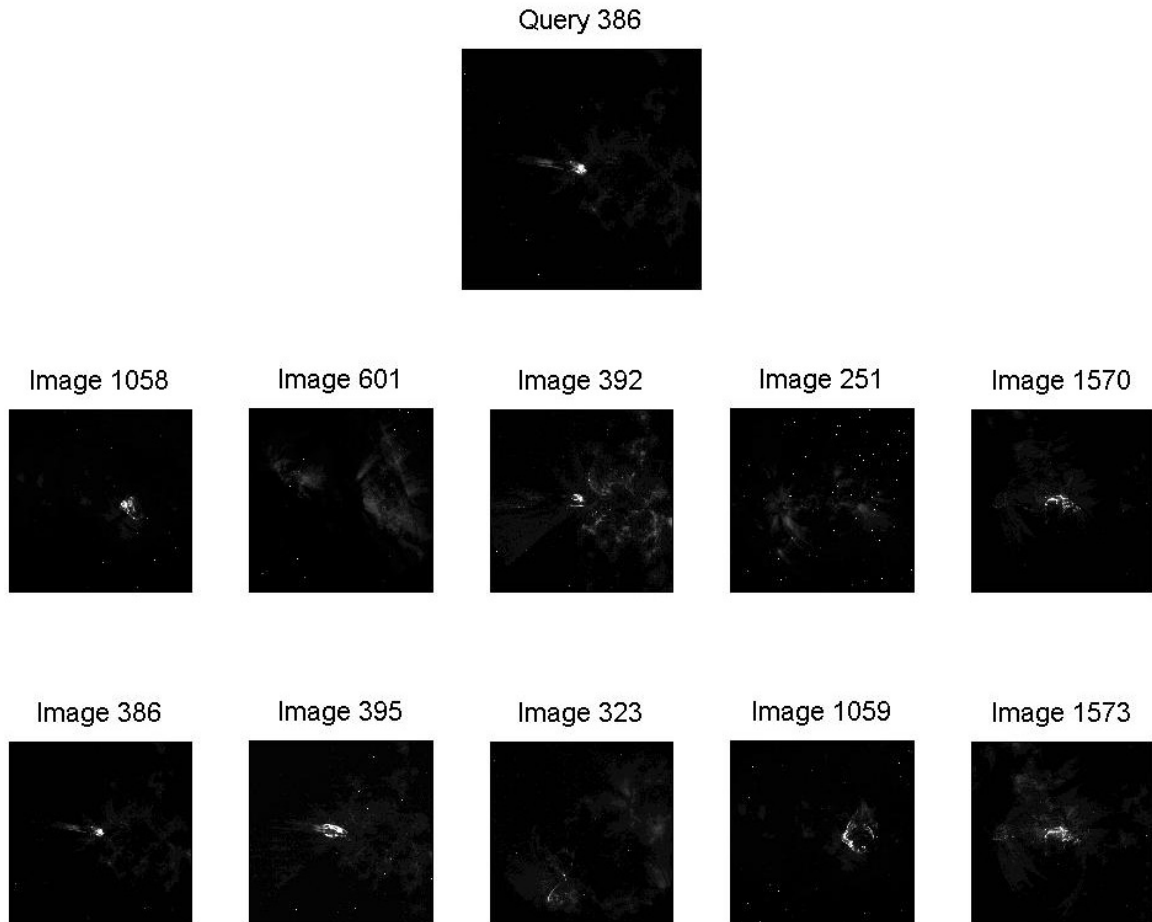


Figure 9.15. Retrieval results for the solar dataset – Coronal Jet class. The query image (#386) is in the top, and the retrieved 10-NN images are in the bottom rows

Figure 9.15, shows a query from the Coronal Jet class that has a retrieval precision of 0.5 or 50%. While, half the images returned are not within the 201-400 range of the Coronal Jet class (which brings the precision values down), there are still a few

images that seem very similar to our query image (#1068 and #1573). In these cases the researchers have to analyze the images to determine if they are valuable for them.

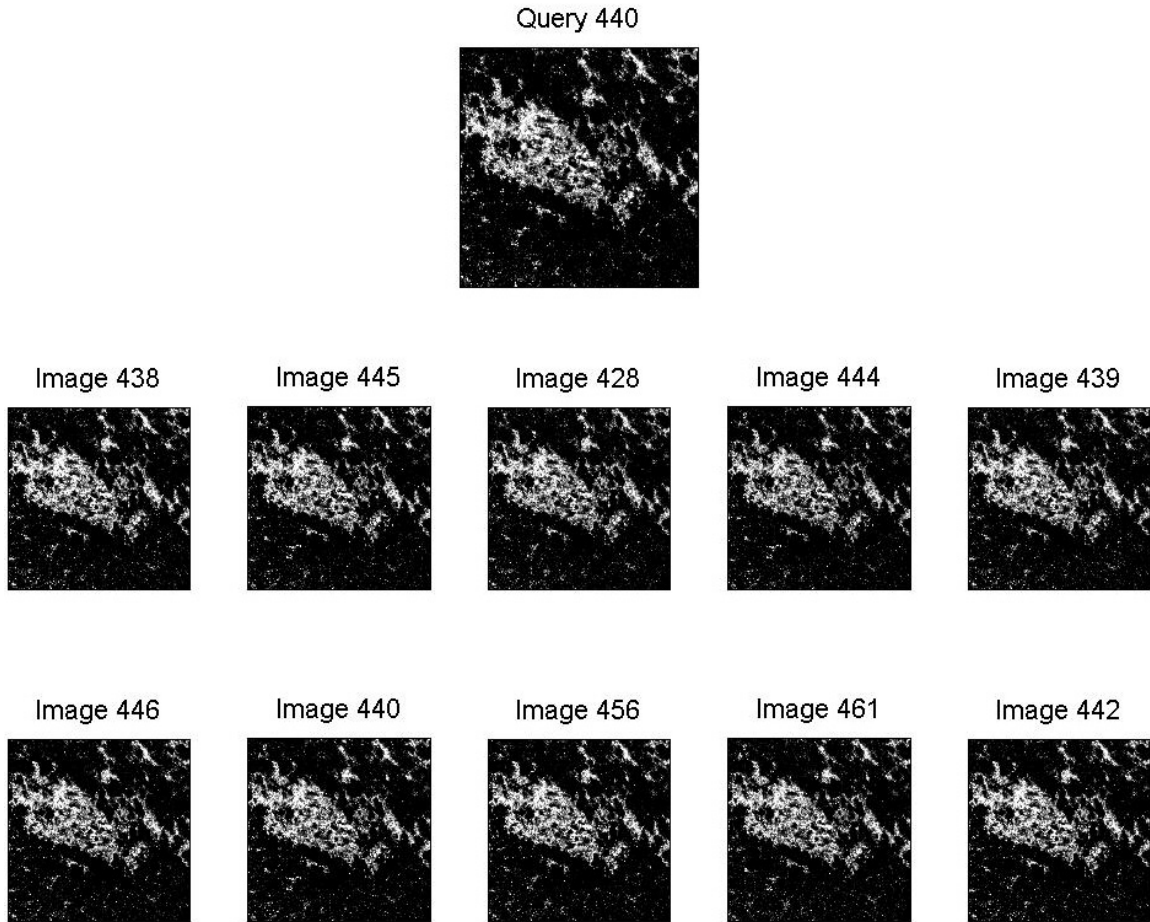


Figure 9.16. Retrieval results for the solar dataset – Emerging Flux class. The query image (#440) is in the top, and the retrieved 10-NN images are in the bottom rows

We again have a query that returns 100% relevant images (retrieval precision of 1.0), it is easy to see in figure 9.16 that all returned images are very similar to each other. This class and the Active Region class are two of the easiest classes to retrieve in our dataset (thus having a high retrieval precision), due to the different wavelengths the images are taken (1600). All remaining classes are from the 171A wavelength.

Query 741

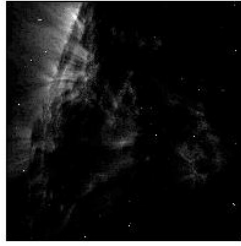


Image 1054

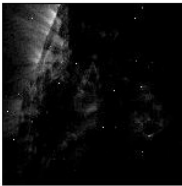


Image 1047

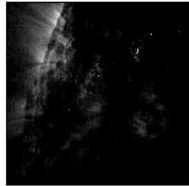


Image 1045

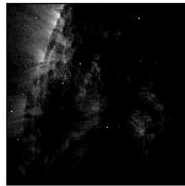


Image 741

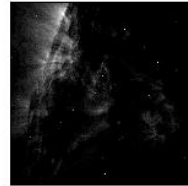


Image 1051

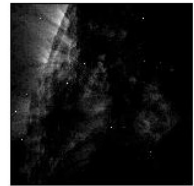


Image 1053

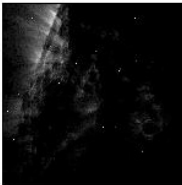


Image 1048

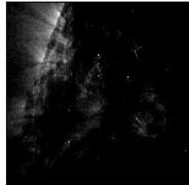


Image 1052

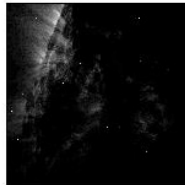


Image 1046

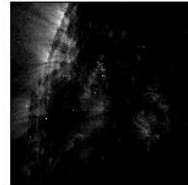


Image 1049

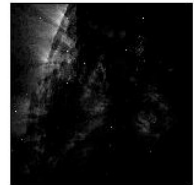


Figure 9.17. Retrieval results for the solar dataset – Filament class. The query image (#741) is in the top, and the retrieved 10-NN images are in the bottom rows

This query image has very interesting results that clearly show the main importance of our CBIR system for solar images. Query #741 has returned a series of images that are very similar but are not found to be included in the Filament class. 9 out of the 10 images belong to the Filament Activation class (image range 1001 – 1200), making this query have a very poor retrieval precision value (0.1 or 10%). However, we just found very similar images that are not in the same class, validating the use of CBIR systems to discover similar images that might have been labeled in correctly or that appear in different classes other than the ones we are querying for.

Query 920

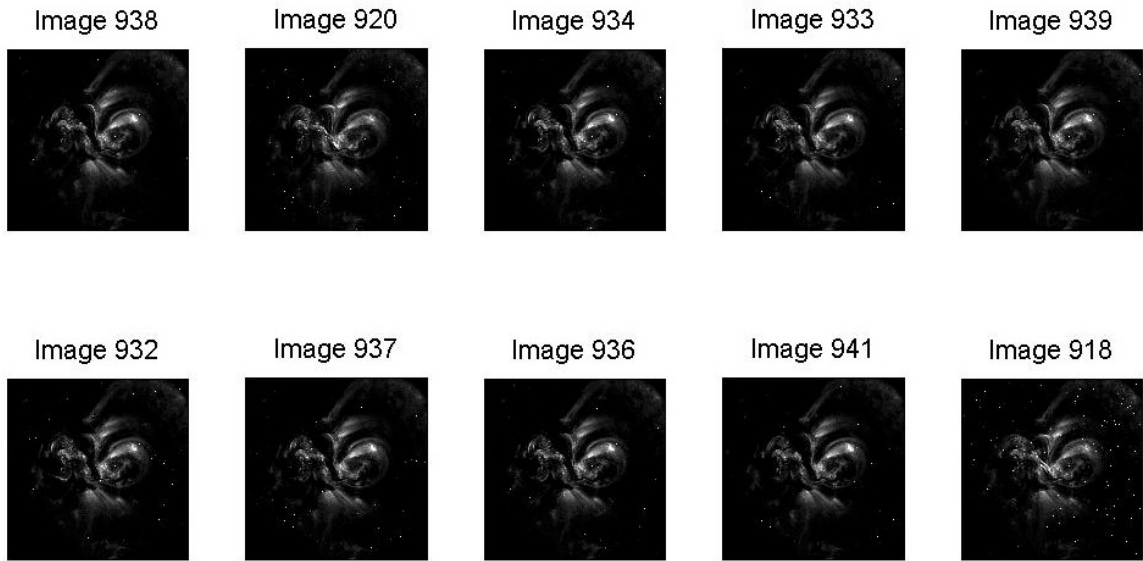
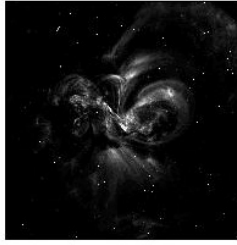


Figure 9.18. Retrieval results for the solar dataset – Filament Activation class. The query image (#920) is in the top, and the retrieved 10-NN images are in the bottom rows

In figure 9.18, we have the Filament Activation class producing very similar results for almost all query images. Our retrieval results for image #920, are very close to each other as we can see from the range of image numbers of the returned images. This retrieval results show the efficiency of our indexing structure, and when considering that we are using a dimensionally reduced version of our feature vector (143 out of 640 dimensions), these results are very good for our SDO CBIR system demo.

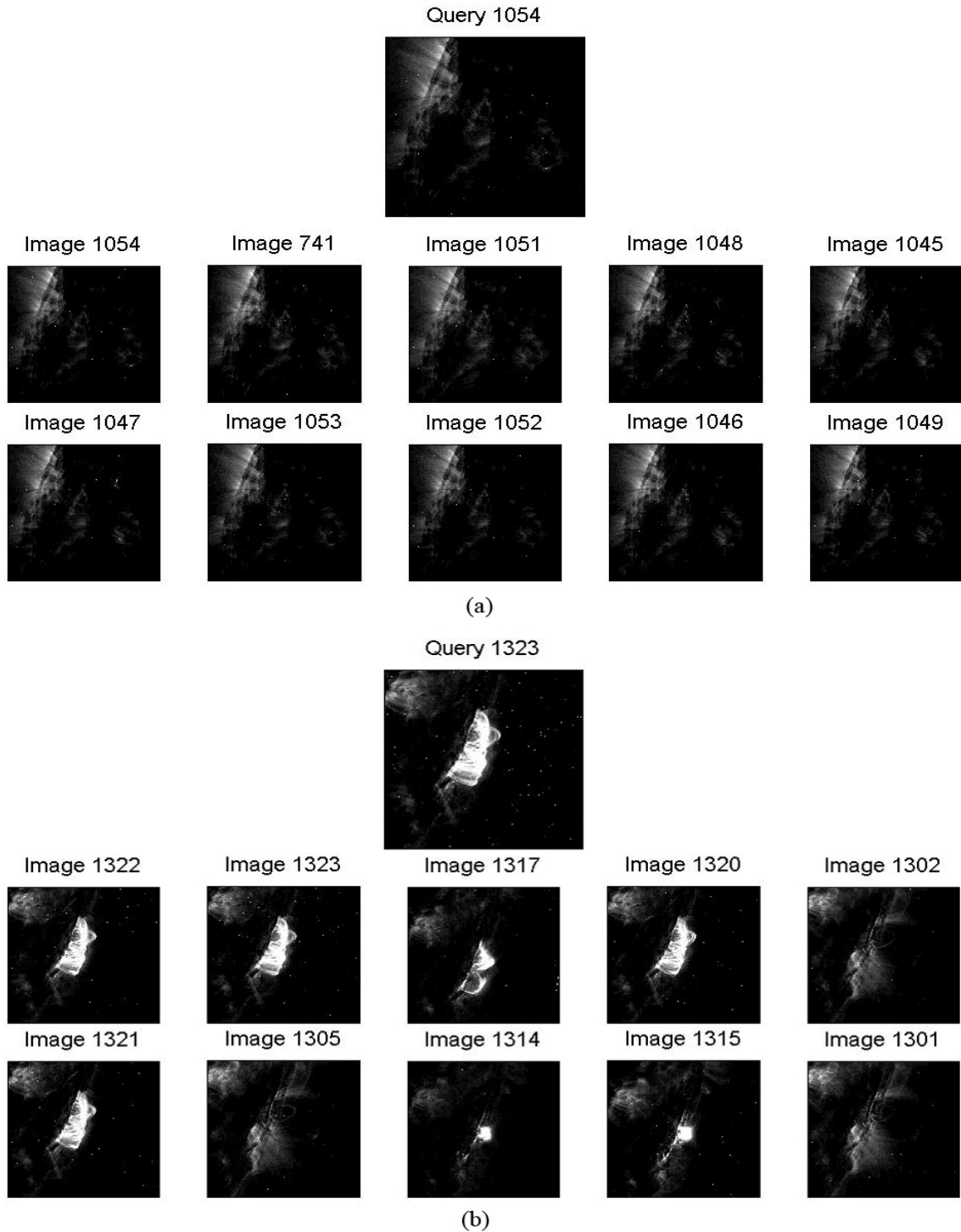


Figure 9.19. (a) Retrieval results for the solar dataset – Filament Eruption class. The query image (#1054) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the solar dataset – Flare class. The query image (#1323) is in the top, and the retrieved 10-NN images are in the bottom rows

In figure 9.19 we have two different queries from two different classes: Filament Eruption (a), and Flare (b). While the Filament Eruption results seem very standard (all returned images are very similar and within the same class, the Flare returned images are the most interesting ones. For the Flare query image #1323, we have very different returned images but all within the same class (comparing #1322 with #1305 and #1317), this shows that our CBIR system is actually allowing us to find similar solar phenomena in terms of finding related images that are not exactly the same or very similar, validating our choice of image parameters and indexing structures at the same time.

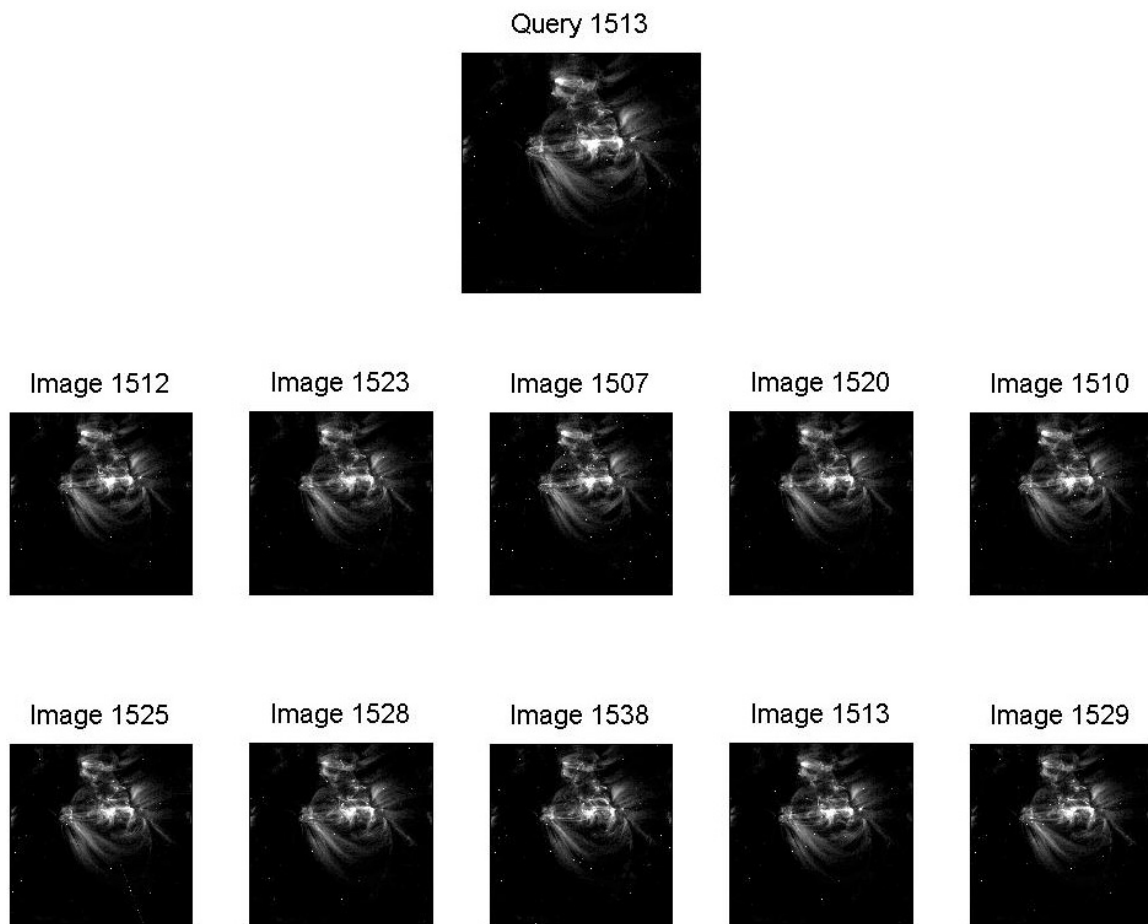


Figure 9.20. Retrieval results for the solar dataset – Oscillation class. The query image (#1513) is in the top, and the retrieved 10-NN images are in the bottom rows

The Oscillation class query image #1513 presented in figure 9.20 shows us the same stable behavior that some of our previous classes. The retrieval results are very accurate since all images are very similar and they all belong to the same class (retrieval precision of 1.0 or 100%).

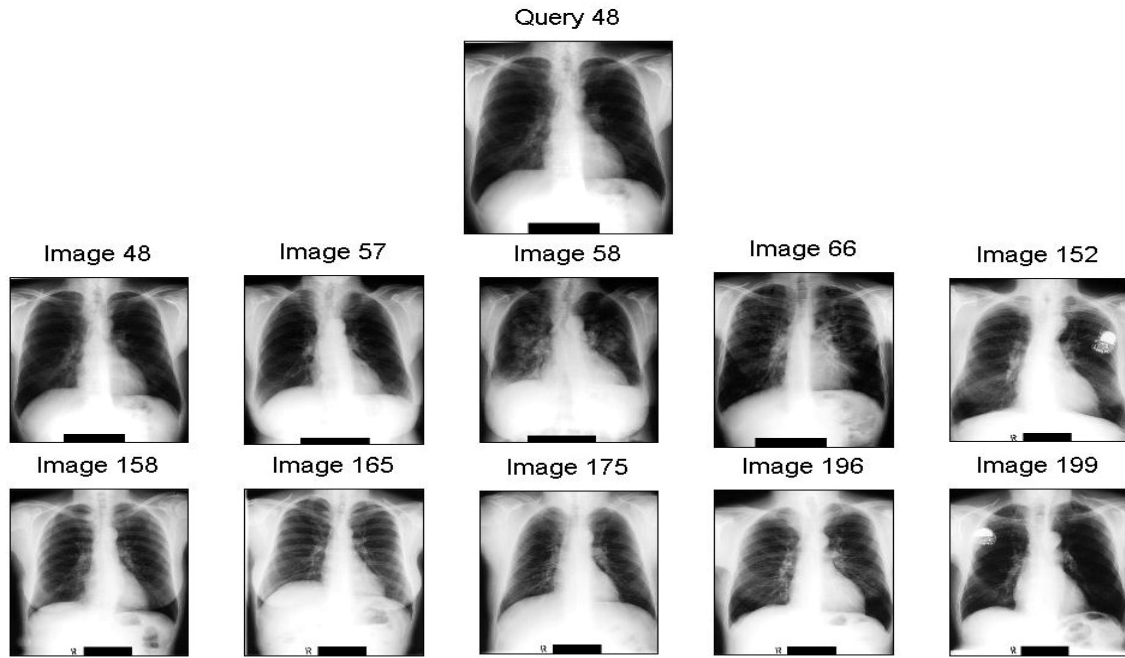
As we have seen in figures 9.13 to 9.20 our SDO CBIR system demo shows the dimensionally reduced dataset index returning very similar (and relevant) results. These results showed the system returning very similar images for most queries presented, even if they are not listed under the same class in our dataset. This behavior will nicely translate into the system returning very similar images in a ‘real life’ setting, allowing researchers to find similar images to the ones they are querying for, thus fulfilling the purpose of our SDO CBIR system. For the remaining 1592 query results, please refer to [122].

ImageCLEFMed07 Images

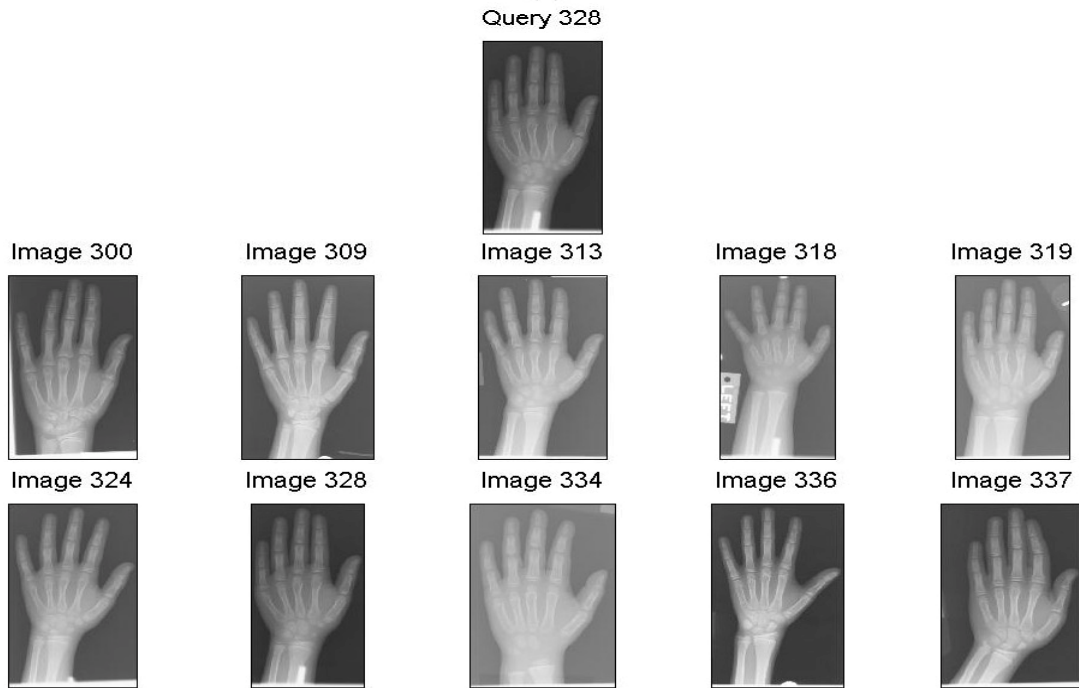
With our CBIR building framework, we managed to develop a CBIR system demo for the ImageCLEFMed datasets. In this section we present our retrieval results for the best performing dimensionality reduction and indexing experiment: Singular Value Decomposition with 251 dimensions (SVD-7) and the Standardized Euclidean dissimilarity measure. With an average retrieval precision of 0.8438 (84%), our demo CBIR system is accurate and produces good results for all eight classes of our subset of the ImageCLEFMed07 dataset. In table 9.4 we present the classes of the dataset and the image number ranges.

Table 9.4. ImageCLEFMed07 dataset classes and image ranges

Class Label	Image Range
Plain Radiography - Chest	1 - 200
Plain Radiography - Hand	201 - 400
Plain Radiography - Cranium	401 - 600
Plain Radiography - Side Chest	601 - 800
Plain Radiography - Trachea	801 - 1000
Plain Radiography - Breast	1001 - 1200
Plain Radiography - Lungs	1201 - 1400
Plain Radiography - Leg Bones	1401 - 1600



(a)



(b)

Figure 9.21. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Chest class. The query image (#48) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Hand class. The query image (#328) is in the top, and the retrieved 10-NN images are in the bottom rows

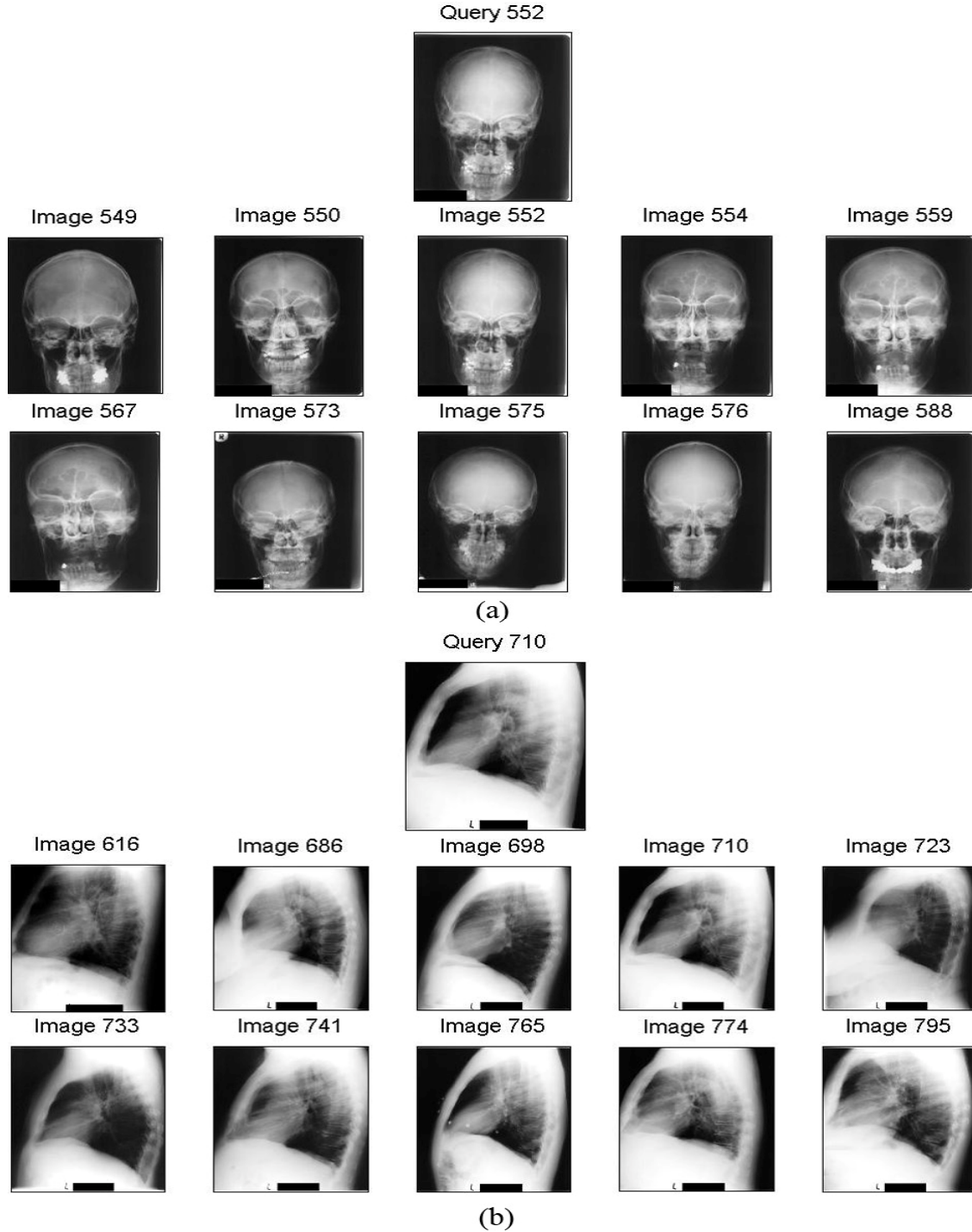


Figure 9.22. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Cranium class. The query image (#552) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Side Chest class. The query image (#710) is in the top, and the retrieved 10-NN images are in the bottom rows

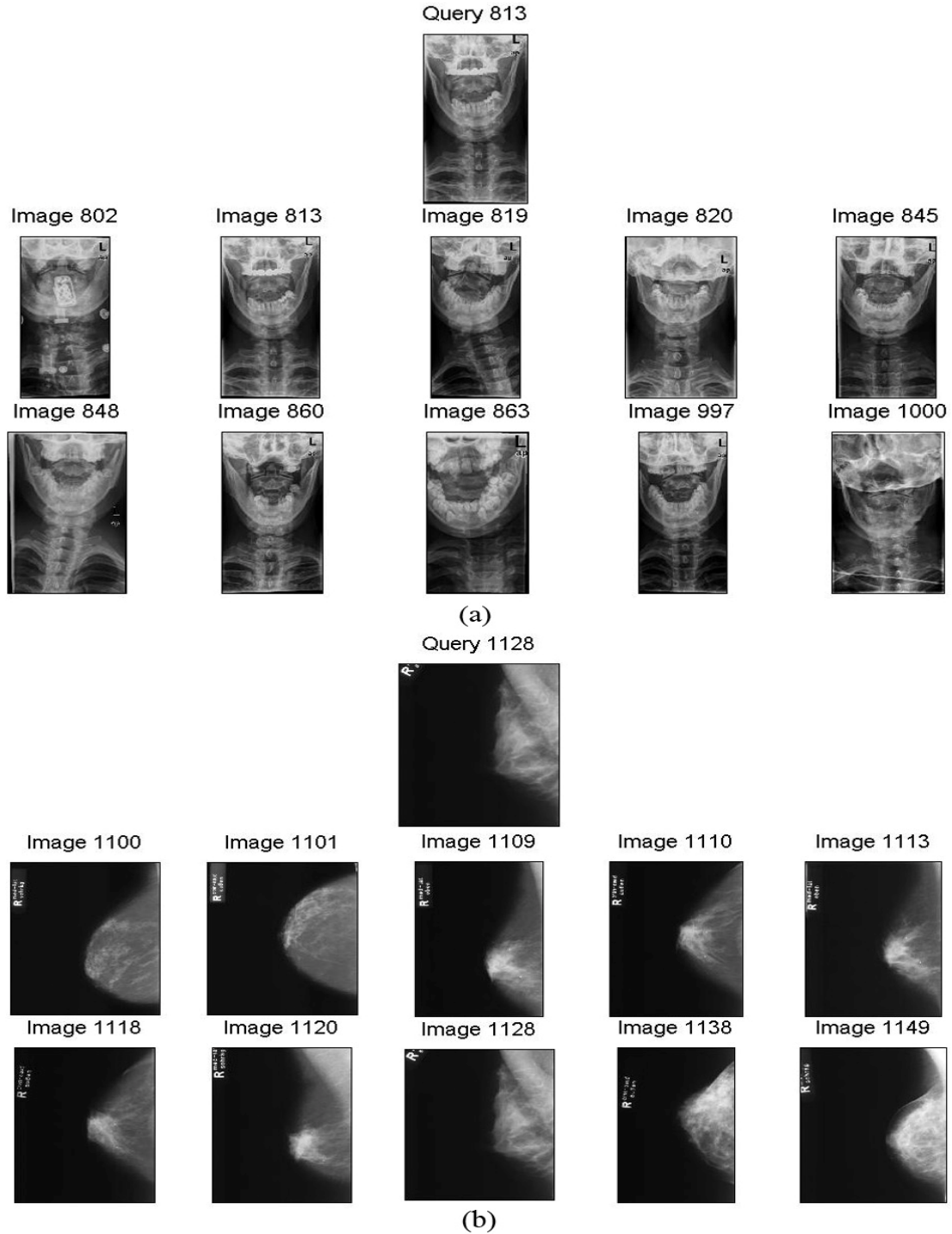


Figure 9.23. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Trachea class. The query image (#813) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Breast class. The query image (#1128) is in the top, and the retrieved 10-NN images are in the bottom rows

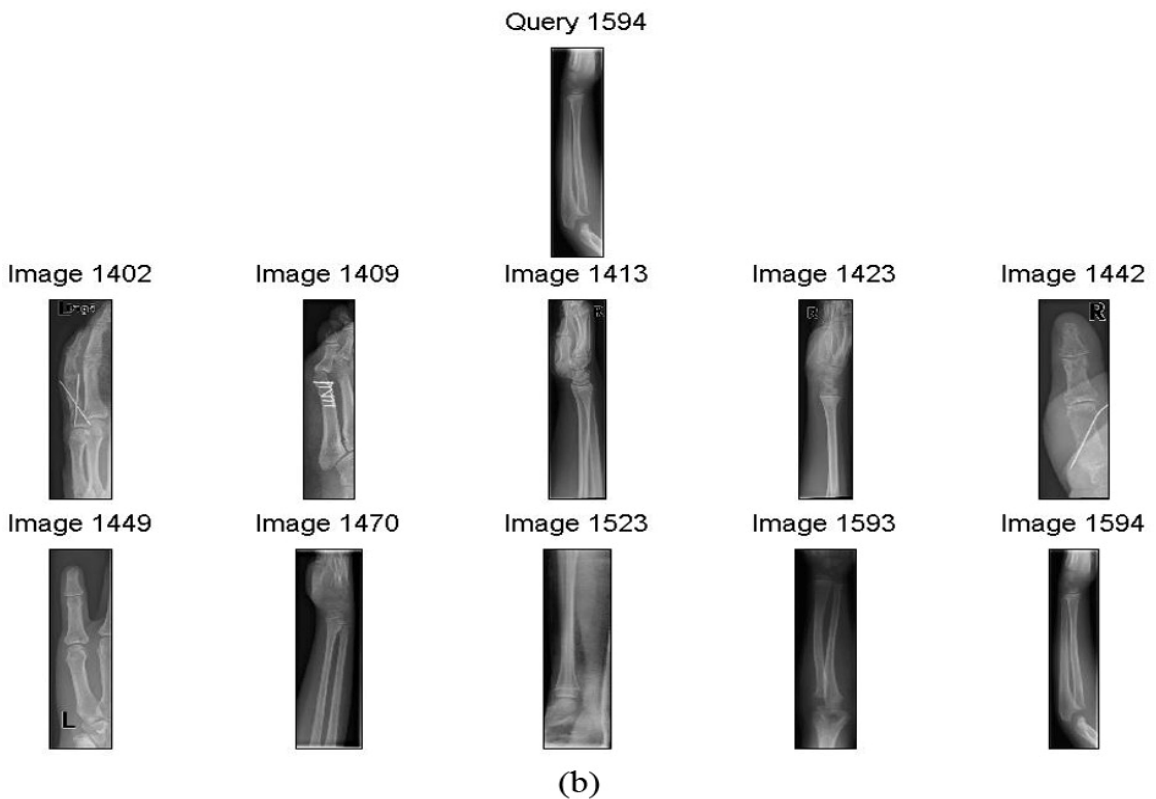
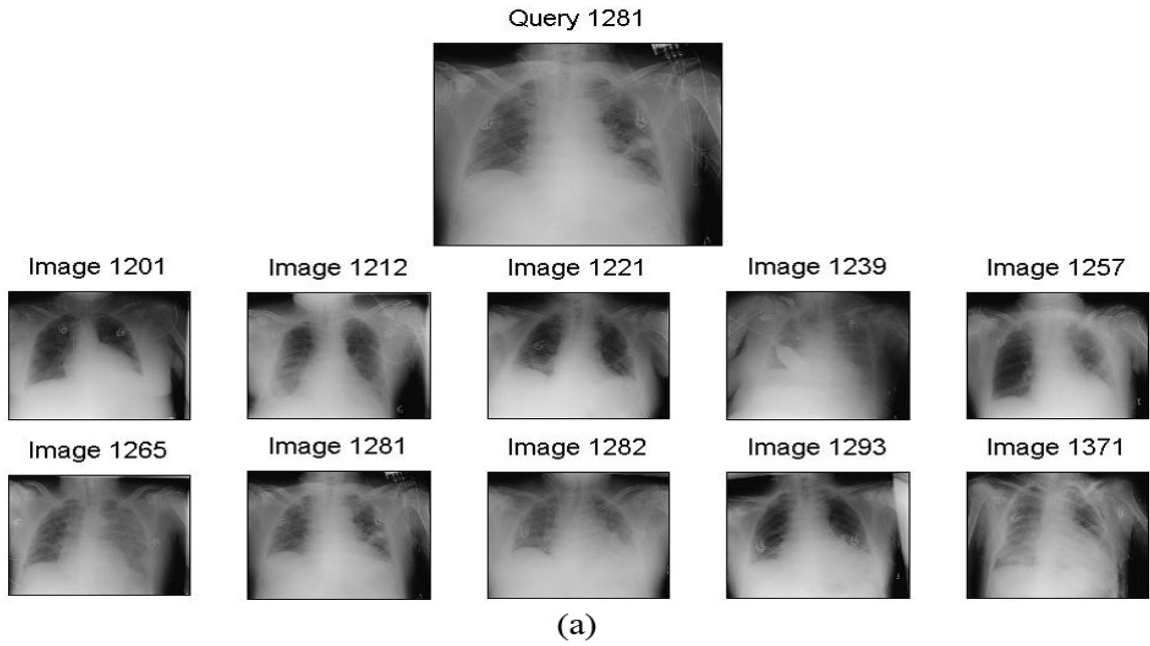


Figure 9.24. (a) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Lungs class. The query image (#813) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the ImageCLEFMed07 dataset – Plain Radiography - Leg Bones class. The query image (#1128) is in the top, and the retrieved 10-NN images are in the bottom rows

PASCAL2008 Images

As one of our poor performing datasets (due to a poor selection of image parameters) we have the PASCAL2008 dataset. Even when using the full 640-dimensional feature vector, we have less than 0.25 (25%) retrieval precision. The following figures are taken from the retrieval results for the 1,600 images and the 10-NN on the full 640-dimensional feature vector using the Euclidean distance. As we can see from the figures, most of the returned results that actually fit within the image range of the query images are random at best since the similarity within the images (besides the query image) is almost non-existent from the visual perspective of the researcher.

Table 9.5. PASCAL2008 dataset classes and image ranges

Class Label	Image Range
Car	1 - 200
Bicycle	201 - 400
Bottle	401 - 600
Boat	601 - 800
Dog	801 - 1000
Tv Monitor	1001 - 1200
Person	1201 - 1400
Cat	1401 - 1600

184

Query 8



Image 342

Image 8



Image 167



Image 352



Image 364



Image 827



Image 1337



Image 1342



Image 1373



Image 1448



(a)

Query 240



Image 240



Image 360



Image 363



Image 425



Image 1161



Image 1205



Image 1447



Image 1535



Image 1560



Image 1579



(b)

Figure 9.25. (a) Retrieval results for the PASCAL2008 dataset – Car class. The query image (#8) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Bicycle class. The query image (#240) is in the top, and the retrieved 10-NN images are in the bottom rows

Query 453



Image 252



Image 345



Image 359



Image 453



Image 954



Image 971



Image 983



Image 984



Image 1467



Image 1474



(a)

Query 737



Image 233



Image 422



Image 537



Image 591



Image 737



Image 812

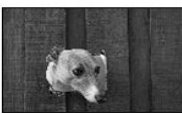


Image 1062



Image 1112



Image 1495



Image 1503



(b)

Figure 9.26. (a) Retrieval results for the PASCAL2008 dataset – Bottle class. The query image (#453) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Boat class. The query image (#737) is in the top, and the retrieved 10-NN images are in the bottom rows

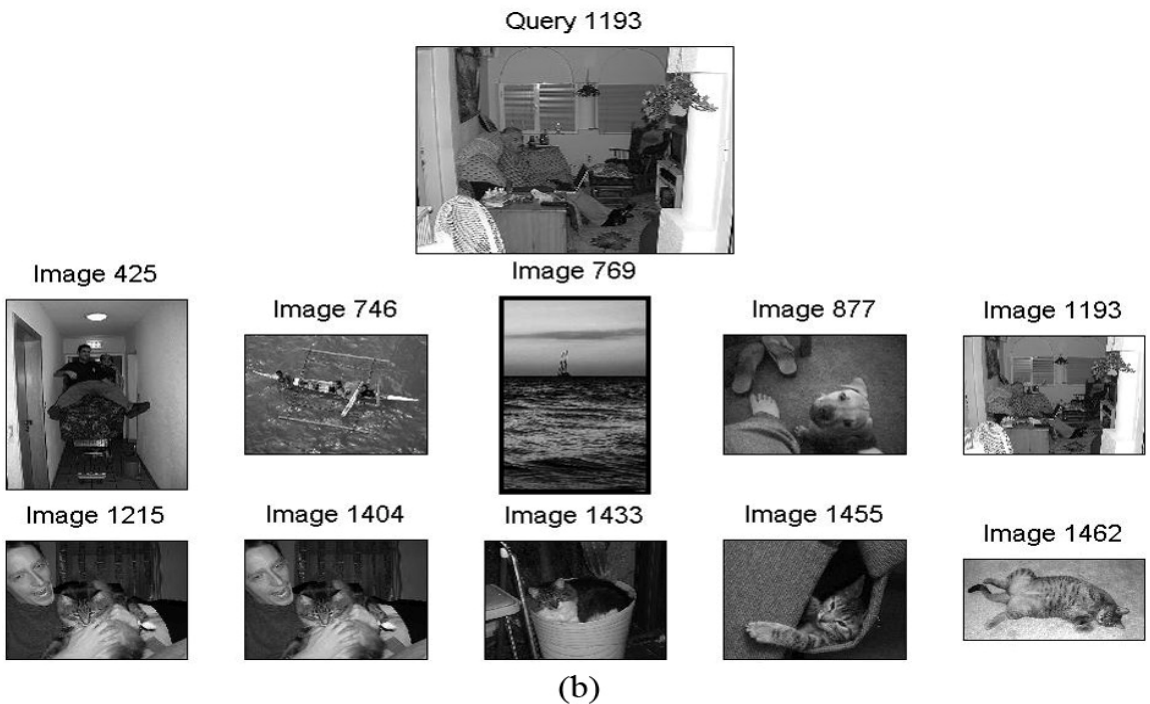
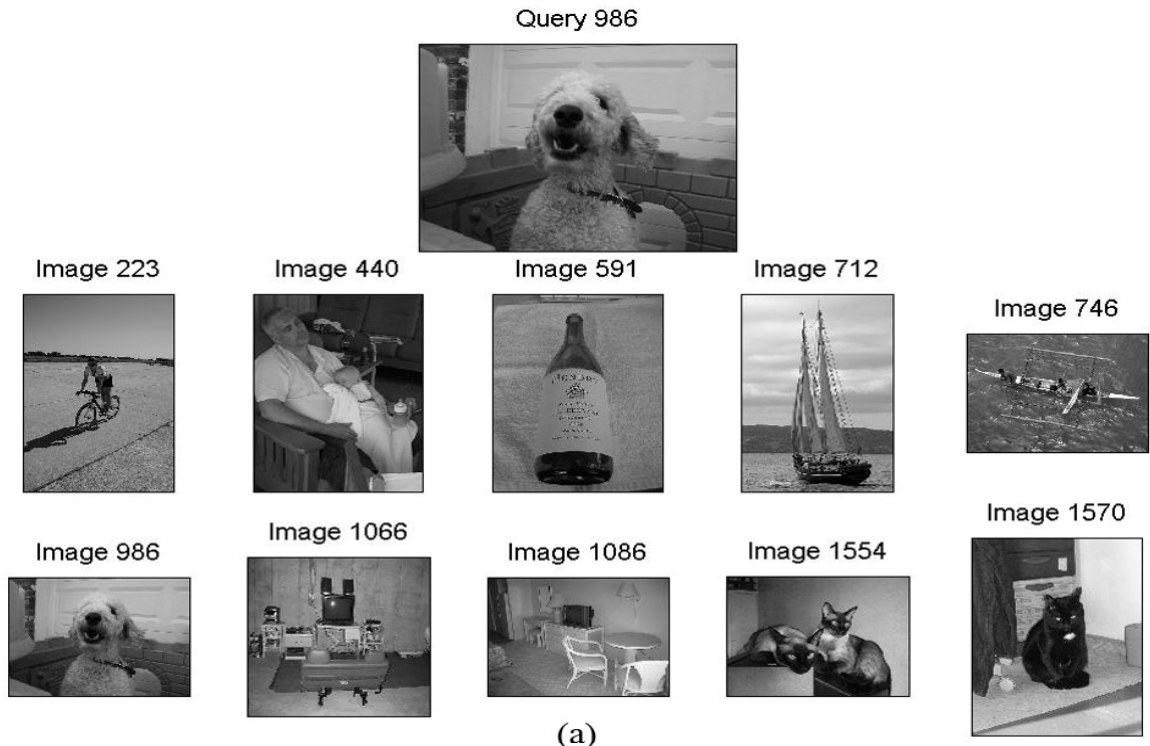


Figure 9.27. (a) Retrieval results for the PASCAL2008 dataset – Dog class. The query image (#986) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – TV Monitor class. The query image (#1193) is in the top, and the retrieved 10-NN images are in the bottom rows

187

Query 1324



Image 252



Image 162



Image 258



Image 292



Image 349



Image 365



Image 753



Image 983



Image 1324



Image 1474



(a)

Query 1517



Image 116



Image 398



Image 474



Image 574



Image 591



Image 764



Image 940



Image 1241



Image 1484

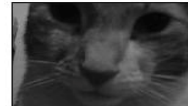


Image 1517



(b)

Figure 9.28. (a) Retrieval results for the PASCAL2008 dataset – Person class. The query image (#1324) is in the top, and the retrieved 10-NN images are in the bottom rows. (b) Retrieval results for the PASCAL2008 dataset – Cat class. The query image (#1517) is in the top, and the retrieved 10-NN images are in the bottom rows

CHAPTER 10

GENERAL CONCLUSIONS

Closing Remarks

This work has introduced the reader to the current state of CBIR systems and the area of image processing for solar physics, in order to put in perspective the need to construct a CBIR system for solar physics, in particular Solar Dynamics Observatory mission. During this process, the idea of creating a framework capable of providing researchers with facilities that would allow them to create or tweak domain-specific CBIR systems was founded.

In chapter one, we presented a full outline of the components of our proposed framework in order to familiarize the reader with the steps performed to build our CBIR system for the SDO mission. After our literature review, we presented the solar dataset that was created to aid us in our ambitious task, as well as the five different datasets that will be used to present comparative analyses of the results provided in each module of our system.

A discussion of image parameters for image retrieval was presented in chapter four. Here we first determined which image parameters to utilize on our CBIR system and implement in our framework. Since this early stage of our work we started observing the high domain dependency of image parameters and the need for further analysis to be performed to determine how useful they are in different domain-specific datasets. Such analysis for our domain is carried out in chapters six and seven.

By performing supervised and un-supervised attribute evaluation we are able to determine which image parameters are most useful, and we can start considering the possibility of achieving dimensionality reduction by removal of not useful image parameters. By paying attention to the unsupervised attribute evaluation, we introduced our dissimilarity measures module that includes over 15 different measures to determine dissimilarity. With this analysis we found some of our most interesting results, and we managed to considerably improve the classification of some of the datasets that originally performed very poorly. With these interesting results we justify the analysis of all these different dissimilarity measures in order to find interesting combinations of image parameters and dissimilarity measures that would help us to classify the images in our datasets.

Chapter eight we showed that the original dimensionality of our solar dataset can be greatly reduced (by at least 77%) and still maintain a fairly good representation of our original feature vector. In order to put these results to the test, we went on to find an indexing algorithm that could handle our high dimensional data and was able to be efficiently queried in order to complete our CBIR system.

In our indexing and retrieval chapter we introduced our dissimilarity measure module into one of the best performing single-dimensional indexing algorithms for high dimensional data. Besides presenting some of the first indexing and retrieval results for a feature vector of over 200 dimensions, we managed to show that several dissimilarity measures performed equally well compared to the Euclidean distance for this indexing algorithm. Other results showed different tendencies on what previous authors have

stated, and can surely lead to future discussions about these behaviors. However, the most important achievement of this section was to demonstrate that our underlying indexing structure was able to achieve similar retrieval performance for the dimensionally reduced datasets, providing us with good retrieval results (86% precision accuracy) for unsupervised (not using the image labels) analysis of our data. As we stated in chapter nine, this greatly resembles the actual usage of our CBIR system when dealing with new incoming (and un-labeled) data and responding to researcher queries for similar images.

As a whole, our CBIR building framework contains all the modules needed to perform an extensive analysis on a dataset and guide researchers in the quest of building and tweaking their domain-specific dataset. As a product of the framework, we have all the necessary elements needed for the SDO mission CBIR system. By providing extensive results and a comparative analysis with other datasets, we justified the need for each of the modules included in our framework. As one of the key principles behind this dissertation, all results are easily reproducible and all the proper files to generate them chapter by chapter are freely available [122].

Ideas For Future Work

Our outlined framework has unlimited potential for growth and as new image parameters are being developed, they can be implemented and added to our framework to create a bigger repository of available parameters for evaluation. This same principle applies to new dissimilarity measure being developed, new dimensionality reduction algorithms and new indexing mechanisms. Besides this growth potential, the framework

also can be adapted as a teaching tool for image retrieval, content-based retrieval and other courses since it allows a quick analysis of image datasets providing intermediate files between modules to be used in many other ways (different attribute evaluation methods, application of clustering algorithms, etc).

In terms of solar physics, the need to identify phenomena that occur over time is a potential area of future research from the perspective of CBIR systems since these systems are geared towards the analysis of static images. We theorize that new image parameters will have to be developed that can take into consideration the changes certain solar phenomena show over time before they can be fully detected. We believe that new image parameters derived from video compression algorithms could be another way to look at these types of phenomena.

In terms of CBIR systems for high-dimensional data, we see great potential in developing better and more efficient techniques to handle high dimensional data. We also observe that while most researchers focus their efforts based on algorithms that depend on the Euclidean distance, there have been researchers that have managed to show (including chapter six of this dissertation) that it is worth analyzing other dissimilarity measures in traditional settings (e.g. correlation analysis) to discover the relationships between images, data points, etc., in different ways. This is one of the main purposes of including the dissimilarity measure module in our framework. Of particular interest is the modification of the Pyramid-tree indexing algorithm to be able to take advantage of different dissimilarity measures and present a comparison similar to the iDistance comparison in chapter nine.

The door is opening for CBIR systems in the solar physics and astrophysics communities. With this system for the SDO mission, we expect many others to follow for different missions and different images for different planets, constellations, galaxies, and many other possibilities. With the exuberant amounts of new images being generated by the SDO mission, the Hubble telescope, and many other satellites, telescopes and observatories, the need to be able to query and retrieve all these images in an efficient manner is evident. There are many other fields taking advantage of the conveniences of CBIR systems (e.g. medical systems for cancer diagnosis, etc), and it is about time that other fields start finding ways of organizing and retrieving the increasingly growing repositories of large-scale image data.

REFERENCES CITED

- [1] R. Datta, D. Joshi, J. Li and J. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age", *ACM Computing Surveys*, vol. 40, no. 2, article 5, pp. 1-60, 2008.
- [2] Y. Rui, T.S. Huang, S. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues". *Journal of Visual Communication and Image Representation* 10, pp. 39–62, 1999.
- [3] H. Müller, N. Michoux, D. Bandon, A. Geissbuhler, "A review of content-based image retrieval systems in medical applications: clinical benefits and future directions". *International journal of medical informatics*, Volume 73, pp. 1-23, 2004
- [4] Y.A Aslandogan, C.T Yu, "Techniques and systems for image and video retrieval" *IEEE Transactions on Knowledge and Data Engineering*, Vol: 11 1 , Jan.-Feb. 1999.
- [5] A. Yoshitaka, T. Ichikawa "A survey on content-based retrieval for multimedia databases" *IEEE Transactions on Knowledge and Data Engineering*, Vol: 11 1 , Jan.-Feb. 1999.
- [6] T. Deselaers, D. Keysers, and H. Ney, "Features for Image Retrieval: An Experimental Comparison", *Information Retrieval*, Vol. 11, issue 2, The Netherlands, Springer, pp. 77-107, 2008.
- [7] H. Müller, A. Rosset, J-P. Vallée, A. Geissbuhler, Comparing feature sets for content-based medical information retrieval. *SPIE Medical Imaging*, San Diego, CA, USA, February 2004.
- [8] S. Antani, L.R. Long, G. Thomas. "Content-Based Image Retrieval for Large Biomedical Image Archives" *Proceedings of 11th World Congress on Medical Informatics (MEDINFO) 2004 Imaging Informatics*. September 7-11 2004; San Francisco, CA, USA. 829-33. 2004.
- [9] R. Lamb, "An Information Retrieval System For Images From The Trace Satellite," M.S. thesis, Dept. Comp. Sci., Montana State Univ., Bozeman, MT, 2008.
- [10] V. Zharkova, S. Ipson, A. Benkhalil and S. Zharkov, "Feature recognition in solar images," *Artif. Intell. Rev.*, vol. 23, no. 3, pp. 209-266. 2005.

- [11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. "The WEKA Data Mining Software: An Update" *SIGKDD Explorations*, Volume 11, Issue 1, 2009
- [12] K. Yang, J. Trewn. *Multivariate Statistical Methods in Quality Management*. McGraw-Hill Professional; pp. 183-185. 2004.
- [13] J. Lin. "Divergence measures based on the shannon entropy". *IEEE Transactions on Information Theory* 37 (1): pp. 145–151. 2001.
- [14] S. Kullback, R.A. Leibler "On Information and Sufficiency". *Annals of Mathematical Statistics* 22 (1): pp. 79–86. 1951.
- [15] J. Munkres. *Topology (2nd edition)*. Prentice Hall, pp 280-281. 1999.
- [16] K. Pearson, "On lines and planes of closest fit to systems of points in space" . *Philosophical Magazine* 2 (6) 1901, pp 559–572.
- [17] M. Belkin and P. Niyogi. "Laplacian Eigenmaps and spectral techniques for embedding and clustering". In *Advances in Neural Information Processing Systems*, volume 14, pp. 585–591, Cambridge, MA, USA. The MIT Press. 2002.
- [18] L.K. Saul, K.Q. Weinberger, F. Sha, J. Hamm and D. D. Lee. Spectral methods for dimensionality reduction. In *Semi-supervised Learning*, 2006. pp. 293-308
- [19] T. Etzold, A. Ulyanov, P. Argos. "SRS: information retrieval system for molecular biology data banks". *Methods Enzymol.* pp. 114–128. 1999
- [20] D. S. Raicu, J. D. Furst, D. Channin, D. H. Xu, & A. Kurani, "A Texture Dictionary for Human Organs Tissues' Classification", *Proceedings of the 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004)*, Orlando, USA, in July 18-21, 2004.
- [21] P. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast and effective retrieval of medical tumor shapes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 6, pp.889-904. 1998.

- [22] T. Kato, "Database architecture for content-based image retrieval". In *Image Storage and Retrieval Systems*, pp. 112-123, 2002
- [23] Flickr [Online] . Available: <http://blog.flickr.net/en/2010/09/19/50000000000/> [Accessed: March 20, 2011]
- [24] J.Z. Wang, N. Boujemaa, A. Del Bimbo, D. Geman, A. Hauptmann, and J. Tesic. "Diversity in multimedia information retrieval research". In *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR) at the International Conference on Multimedia*. 2006
- [25] P. Aigrain, H. Zhang, and D. Petkovic. "Content-based representation and retrieval of visual media:A review of the state-of-the-art". *Multimed. Tools Appl.* 3, 3, pp. 179–202. 1996
- [26] Y. Rui, T. Huang, and S.-F. Chang. "Image retrieval: Current techniques, promising directions and open issues". *J. Visual Commun. Image Represent.* 10, 1, pp. 39–62. 1999
- [27] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. "Content-based image retrieval at the end of the early years". *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 12, pp 1349–1380. 2000
- [28] C.G.M. Snoek and M. Worring. "Multimodal video indexing: A review of the state-of-the-art". *Multimed. Tools Appl.* 25, 1, pp. 5–35. 2005
- [29] H. Muller, T. Pun, and D. Squire. "Learning from user behavior in image retrieval: Application of market basket analysis". *Int. J. Comput. Vision* 56, 1–2, pp. 65–77. 2004
- [30] C. Bohm, S. Berchtold, and D.A. Keim. "Searching in high-dimensional space index structures for improving the performance of multimedia databases". *ACM Comput. Surv.* 33, 3, pp 322–373. 2001
- [31] S.-C. Zhu and A. Yuille. "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation". *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 9, pp 884–900. 1996

- [32] J. Shi and J. Malik. "Normalized cuts and image segmentation". *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8, pp 888–905. 2000
- [33] E. Petrakis and A. Faloutsos. "Similarity searching in medical image databases". *IEEE Trans. Knowl. Data Eng.* 9, 3, pp 435–447. 1997
- [34] D. Swets and J. Weng. "Using discriminant eigenfeatures for image retrieval". *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 8, pp 831–836. 1996
- [35] K.S. Fu and J.K. Mui, "A survey on image segmentation". *Pattern Recognition*, Volume 13, Issue 1, pp 3-16, 1981
- [36] D. Zhang and G. Lu. "Segmentation of moving objects in image sequence: A review". *Circuits, systems, and signal processing* V. 20 N. 2 pp 143-183, 2001
- [37] J.B. Maintz and M.A. Viergever. "A survey of medical image registration". *Medical image analysis* V. 2 N. 1, pp 1-36, 1998
- [38] M. Swain and B. Ballard. "Color indexing". *Int. J. Comput. Vision* 7, 1, pp 11–32. 1991
- [39] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker. "Query by image and video content: The QBIC system". *IEEE Comput.* 28, 9, pp 23–32. 1995
- [40] T. Gevers and A. Smeulders. "Pictoseek: Combining color and shape invariant features for image retrieval". *IEEE Trans. Image Process*, 9, 1, pp 102–119. 2000
- [41] J. Smith and S.-F. Chang. "Visualeek: A fully automated content-based image query system". In *Proceedings of the ACM International Conference on Multimedia*. 1997
- [42] G. Finlayson. "Color in perspective". *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 10, pp 1034–1038. 1996
- [43] J. Huang, S. Ravi Kumar, M. Mitra, W.-J. Zhu and R. Zabih. "Spatial color indexing and applications". *Int. J. Comput. Vision* 35, 3, pp 245–268. 1999

- [44] J. Wang, G. Wiederhold, O. Firschein, and S. Wei. "Content-Based image indexing and searching using Daubechies' wavelets". *Int. J. Digital Librar.* 1, 4, pp 311–328. 1998
- [45] B. Manjunath, W.-Y. Andma. "Texture features for browsing and retrieval of image data". *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 8, pp 837–842. 1996
- [46] W.-Y. Andma and B. Manjunath. "Texture thesaurus for browsing large aerial photographs". *J. Amer. Soc. Inf. Sci.* 49, 7, pp 633–648. 1998
- [47] A. Guttman. "R-trees: A Dynamic Index Structure for Spatial Searching", *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Boston, MA, pp. 47-57. 1984.
- [48] N. Beckmann, H.P. Kriegel, R. Schneider, B. Seeger. "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City, NJ, pp. 322-331. 1990.
- [49] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509-517, 1975.
- [50] C. Yu, B. C. Ooi, K. Tan and H. V. Jagadish. "Indexing the distance: an efficient method to KNN processing", *Proceedings of the 27th international conference on Very large data bases*, Roma, Italy, 421-430, 2001.
- [51] S. Berchtold, C. Böhm, and H. Kriegel. "The pyramid-technique: towards breaking the curse of dimensionality". In *Proceedings of the 1998 ACM SIGMOD international Conference on Management of Data (Seattle, Washington, United States, June 01 - 04, 1998)*. *SIGMOD '98*. ACM, New York, NY, pp. 142-153. 1998.
- [52] H. V. Jagadish, B. C. Ooi, K. Tan, C. Yu and R. Zhang "iDistance: An Adaptive B+-tree Based Indexing Method for Nearest Neighbor Search", *ACM Transactions on Data Base Systems (ACM TODS)*, 30, 2, pp. 364-397, 2005.
- [53] H.T. Shen, X. Zhou, and A. Zhou, "An adaptive and dynamic dimensionality reduction method for high-dimensional indexing", *presented at VLDB J.*, 2007, pp.219-234.
- [54] R. Wilson and E. Hancock. "Structural matching by discrete relaxation". *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 6, pp 634–648. 1997

- [55] A. Del Bimbo, and P. Pala. "Visual image retrieval by elastic matching of user sketches". *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 2, pp 121–132. 1997
- [56] Y. Wu, Q Tian and T.S. Huang. "Discriminant-EM algorithm with application to image retrieval". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2000
- [57] M. Webe, M. Welling and P. Perona. "Unsupervised learning of models for recognition". In *Proceedings of the European Conference on Computer Vision (ECCV)* 2000
- [58] J.Z. Wang, J.Li and G. Wiederhold. "SIMPLiCity: Semantics-Sensitive Integrated Matching for Picture Libraries". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947-963, 2001.
- [59] Z. Wang, Z. Chi, and D. Feng. "Fuzzy integral for leaf image retrieval". In *Proceedings of the IEEE International Conference on Fuzzy Systems*. 2002
- [60] T.H. Painter, J. Dozier, D.A. Roberts, R.E. Davis, and R.O. Green. "Retrieval of subpixel snowcovered area and grain size from imaging spectrometer data". *Remote Sens. Env.* 85, 1, pp 64–77. 2003
- [61] M. Schroder, H. Rehrauer, K. Seidel and M. Anddatcu. "Interactive learning and probabilistic retrieval in remote sensing image archives". *IEEE Trans. Geosci. Remote Sens.* 38, 5, pp 2288–2298. 2000
- [62] The Best Airplane Information, Aviation Photos and Aviation News! [Online], Available: <http://www.airliners.net> [Accessed: March 20, 2011]
- [63] CLUE (Cluster-based Image Retrieval) System [Online], Available: http://wang14.ist.psu.edu/cgi-bin/yixin/spectralsearch_show.cgi [Accessed: March 20, 2011]
- [64] FIRE -- Flexible Image Retrieval Engine [Online], Available: <http://thomas.deselaers.de/fire/> [Accessed: March 20, 2011]
- [65] V. Zharkova and V. Schetin, "Filament recognition in solar images with the neural network technique," *Solar Physics*, vol. V228, no. 1, 2005, pp. 137-148. 2005.

- [66] V. Delouille, J. Patoul, J. Hochedez, L. Jacques and J.P. Antoine ,“Wavelet spectrum analysis of eit/soho images," *Solar Physics*, vol. V228, no. 1, 2005, pp. 301-321. 2005.
- [67] A. Irbah, M. Bouzaria, L. Lakhal, R. Moussaoui, J. Borgnino, F. Laclare and C. Delmas, “Feature extraction from solar images using wavelet transform: image cleaning for applications to solar astrolabe experiment.” *Solar Physics*, Volume 185, Number 2, April 1999 , pp. 255-273(19). 1999.
- [68] K. Bojar and M. Nieniewski. “Modelling the spectrum of the fourier transform of the texture in the solar EIT images”. *MG&V* 15, 3, pp. 285-295. 2006.
- [69] S. Christe, I. G. Hannah, S. Krucker, J. McTiernan, and R. P. Lin. “RHESSI Microflare Statistics. I. Flare-Finding and Frequency Distributions”. *ApJ*, 677 pp. 1385–1394. 2008.
- [70] P. N. Bernasconi, D. M. Rust, and D. Hakim. “Advanced Automated Solar Filament Detection And Characterization Code: Description, Performance, And Results”. *Sol. Phys.*, 228. pp. 97–117, 2005.
- [71] A. Savcheva, J. Cirtain, E. E. Deluca, L. L. Lundquist, L. Golub, M. Weber, M. Shimojo, K. Shibasaki, T. Sakao, N. Narukage, S. Tsuneta, and R. Kano. “A Study of Polar Jet Parameters Based on Hinode XRT Observations”. *Publ. Astron. Soc. Japan*, 59:771–+. 2007.
- [72] I. De Moortel and R. T. J. McAteer. “Waves and wavelets: An automated detection technique for solar oscillations”. *Sol. Phys.*, 223. pp. 1–2. 2004.
- [73] R. T. J. McAteer, P. T. Gallagher, D. S. Bloomfield, D. R. Williams, M. Mathioudakis, and F. P. Keenan. “Ultraviolet Oscillations in the Chromosphere of the Quiet Sun”. *ApJ*, 602, pp. 436–445. 2004.
- [74] V. Barra, V. Delouille, J.F. Hochedez, P. Chainais, "Segmentation of EIT images using fuzzy clustering: a preliminary study". In *The Dynamic Sun: Challenges for Theory and Observations*, SP-600, ESA, Noordwijk, 71 – 80.
- [75] J. M. Banda and R. Anrgyk “An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image Categorization”. *FLAIRS-23*:

- Proceedings of the twenty-third international Florida Artificial Intelligence Research Society conference*, Daytona Beach, Florida, USA, May 19–21 2010. 2010.
- [76] W. Hersh, H. Müller, J. Kalpathy-Cramer, E. Kim, X. Zhou, “The consolidated ImageCLEFmed Medical Image Retrieval Task Test Collection”, *Journal of Digital Imaging*, volume 22(6), 2009, pp 648-655.
- [77] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman, A. "The PASCAL Visual Object Classes (VOC) Challenge" *International Journal of Computer Vision*, 88(2), 303-338, 2010
- [78] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. “A discriminative approach to robust visual place recognition”. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, Beijing, China, 2006.
- [79] Heliophysics Event Registry [Online], Available: <http://www.lmsal.com/~cheung/hpkb/index.html> [Accessed: Sep 24, 2010]
- [80] TRACE On-line (TRACE) [Online], Available: <http://trace.lmsal.com/>. [Accessed: Sep 29, 2010]
- [81] TRACE Dataset (MSU) [Online], Available: <http://www.cs.montana.edu/angryk/SDO/data/TRACEbenchmark/> [Accessed: Sep 29, 2010]
- [82] J.M Banda and R. Angryk “On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images” *Proceedings of the 18th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '09)*, Jeju Island, Korea, August 2009, pp. 2019-2024.
- [83] J.M Banda and R. Angryk “Selection of Image Parameters as the First Step Towards Creating a CBIR System for the Solar Dynamics Observatory”. *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. Sydney, Australia, December 1-3, 201. pp. 528-534.
- [84] The INDECS Database [Online], Available: <http://cogvis.nada.kth.se/INDECS/> [Accessed: Sep 29, 2010]

- [85] Cross Language Evaluation Forum [Online] , Available: <http://www.clef-campaign.org/> [Accessed: Sep 29, 2010]
- [86] S. Kulkarni, B. Verma, "Fuzzy Logic Based Texture Queries for CBIR," *Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03)*, pp.223, 2003
- [87] H Lin, C Chiu, and S. Yang, "LinStar texture: a fuzzy logic CBIR system for textures", In *Proceedings of the Ninth ACM international Conference on Multimedia (Ottawa, Canada). MULTIMEDIA '01*, vol. 9. ACM, New York, NY, pp 499-501. 2001.
- [88] S. Thumfart, W. Heidl, J. Scharinger, and C. Eitzinger. "A Quantitative Evaluation of Texture Feature Robustness and Interpolation Behaviour". In *Proceedings of the 13th international Conference on Computer Analysis of Images and Patterns*. 2009.
- [89] J. Muwei, L. Lei, G. Feng, "Texture Image Classification Using Perceptual Texture Features and Gabor Wavelet Features," *Asia-Pacific Conference on Information Processing* vol. 2, pp.55-58, 2009.
- [90] E. Cernadas, P. Carrión, P. Rodriguez, E. Muriel, and T. Antequera. "Analyzing magnetic resonance images of Iberian pork loin to predict its sensorial characteristics" *Comput. Vis. Image Underst.* 98, 2 pp. 345-361. 2005.
- [91] S.S. Holalu and K. Arumugam "Breast Tissue Classification Using Statistical Feature Extraction Of Mammograms", *Medical Imaging and Information Sciences*, Vol. 23 No. 3, pp. 105-107. 2006
- [92] S. T. Wong, H. Leung, and H. H. Ip, "Model-based analysis of Chinese calligraphy images" *Comput. Vis. Image Underst.* 109, 1 (Jan. 2008), pp. 69-85. 2008.
- [93] V. Devendran, T. Hemalatha, W. Amitabh "SVM Based Hybrid Moment Features for Natural Scene Categorization," *International Conference on Computational Science and Engineering* vol. 1, pp.356-361, 2009.
- [94] B. B. Chaudhuri, Nirupam Sarkar, "Texture Segmentation Using Fractal Dimension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 72-77, Jan. 1995

- [95] C. Wen-lun, S. Zhong-ke, F. Jian, "Traffic Image Classification Method Based on Fractal Dimension," *IEEE International Conference on Cognitive Informatics* Vol. 2, pp.903-907, 2006.
- [96] A.P Pentland, "Fractal-based description of natural scenes", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6 pp. 661-674, 1984.
- [97] H.F. Jelinek, D.J. Cornforth, A.J. Roberts, G. Landini, P. Bourke, and A. Iorio, "Image processing of finite size rat retinal ganglion cells using multifractal and local connected fractal analysis", In *17th Australian Joint Conference on Artificial Intelligence, volume 3339 of Lecture Notes in Computer Science*, pages 961--966. Springer--Verlag Heidelberg, 2004
- [98] N. Orlov, L. Shamir, T. Macura, J. Johnston, D.M. Eckley, I.G. Goldberg. "WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern Recognition Letters*. 29(11): 1684-93, 2008.
- [99] R. C. Gonzalez and R. E. Woods "*Digital Image Processing (3rd Edition)*". Prentice-Hall, Inc. Upper Saddle River, pp. 100-120, 2006.
- [100] M. Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. New York: W. H. Freeman, pp. 41-45, 1991.
- [101] H. Tamura, S. Mori, T. Yamawaki. "Textural Features Corresponding to Visual Perception". *IEEE Transaction on Systems, Man, and Cybernetics* 8(6): pp. 460--472. 1978.
- [102] T. Deselaers. *Features for image retrieval. Master's thesis*, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, 2003.
- [103] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and efective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231-262, July 1994.
- [104] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *International Conference on Computer Vision*, volume 2, pages 1165-1173, Corfu, Greece, Sept. 1999.

- [105] B. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703{715, June 2001.
- [106] Z. Yang and C. Kuo. Survey on image content analysis, indexing, and retrieval techniques and status report of MPEG-7. *Tamkang Journal of Science and Engineering*, 3(2):101{118, 1999.
- [107] T. Sikora. "The MPEG-7 visual standard for content description - an overview". *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696-702, 2001.
- [108] J.-R. Ohm. The MPEG-7 visual description framework { concepts, accuracy and applications. In *CAIP 2001, number 2124 in Lecture Notes in Computer Science*, pages 2{10, 2001.
- [109] H. Eidenberger. How good are the visual MPEG-7 features? In *Proceedings SPIE Visual Communications and Image Processing Conference*, volume 5150, pages 476{488, Lugano, Italy, 2003.
- [110] D.G. Lowe. "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision*. 2. pp. 1150–1157, 1999
- [111] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008
- [112] L. Bretzner and T. Lindeberg. "Feature Tracking with Automatic Selection of Spatial Scales". *Computer Vision and Image Understanding* 71: pp 385–392, 1998
- [113] J. Matas, O. Chum, M. Urba, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." *Proc. of British Machine Vision Conference*, pages 384-396, 2002.
- [114] M. Minsky. "Steps toward Artificial Intelligence." *Proceedings of the IRE* 49(1):8-30, 1961.
- [115] J. Quinlan. "Induction of Decision Trees". *Mach. Learn.* 1, 1 (Mar. 1986), 81-106.

- [116] J. Quinlan. "*C4.5: Programs for Machine Learning*". Morgan Kaufmann: San Mateo, CA., 1993
- [117] C. Cortes and V. Vapnik. "Support-Vector Networks", *Machine Learning*, 20, 1995
- [118] L. Breiman. "Random Forests". *Machine Learning* 45 (1): 5–32, 2001.
- [119] D. Hand, and K. Yu. "Idiot's Bayes-not so stupid after all?" *International Statistical Review* 69(3): 385-398. 2001
- [120] S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, *Informatica* 31(2007) 249-268, 2007
- [121] Adapted from the Wikimedia Commons file "File:Svm_separating_hyperplanes.png" [Online] Available: http://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes.png [Accessed: March 20, 2011]
- [122] Juan M. Banda - Dissertation Materials [Online] Available: <http://www.jmbanda.com/Dissertation/> [Accessed: March 27, 2011]
- [123] J. Muwei, L. Lei and G. Feng Texture Image Classification Using Perceptual Texture Features and Gabor Wavelet Features. *Asia-Pacific Conference on Information Processing* vol. 2, 55-58. 2009
- [124] N. Vasconcelos, M. Vasconcelos. Scalable Discriminant Feature Selection for Image Retrieval and Recognition. In *CVPR 2004*. (Washington, DC 2004), 770–775. 2004.
- [125] I. Borg, and P. Groenen. "*Modern multidimensional scaling: theory and applications*" (2nd ed.), Springer-Verlag New York, 2005
- [126] G. D Guo, A.K. Jain, W.Y Ma, H.J Zhang, "Learning similarity measure for natural image retrieval with relevance feedback". *IEEE Transactions on Neural Networks*. Volume 13 (4): 811--820, 2002
- [127] R. Lam, H. Ip, K. Cheung, L. Tang, R. Hanka, "Similarity Measures for Histological Image Retrieval," *15th International Conference on Pattern Recognition (ICPR'00)* - Volume 2: 2295. 2000.

- [128] T. Ojala, M. Pietikainen, and D. Harwood. "A comparative study of texture measures with classification based feature distributions." *Pattern Recognition*, 29(1):51–59. 1996.
- [129] C. Aggarwal, A. Hinneburg, D. A. Keim. "On the surprising behavior of distance metrics in high dimensional space." *Database Theory—ICDT 2001*: 420-434.
- [130] D. Francois, V. Wertz, M. Verleysen. "Non-Euclidean metrics for similarity search in noisy datasets." *ESANN 2005*:27-29.
- [131] M. Beatty, B. S. Manjunath, "Dimensionality Reduction Using Multi-Dimensional Scaling for Content-Based Retrieval," *Image Processing, International Conference on, International Conference on Image Processing (ICIP'97) - Volume 2*: 835, 1997.
- [132] Y. Rubner, L.J. Guibas, and C. Tomasi, C. "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval" *Proceedings of the ARPA Image Understanding Workshop* :661—668, 1997
- [133] K. Yang, J. Trewin. *Multivariate Statistical Methods in Quality Management*. McGraw-Hill Professional; February 24, 2004 pp. 183-185.
- [134] P.-N. Tan, M. Steinbach & V. Kumar, "*Introduction to Data Mining*", Addison-Wesley 500, 2005.
- [135] C. Spearman, "The proof and measurement of association between two things" *Amer. J. Psychol.*, V 15 :72–101. 1904
- [136] A. Shahrokni. "Texture Boundary Detection for Real-Time Tracking" *Computer Vision - ECCV 2004*: 566-577. 2004.
- [137] F. Moisy "EzyFit 2.30" [Online], Available: <http://www.mathworks.com/matlabcentral/fileexchange/10176> [Accessed: Nov 12, 2010]
- [138] R.N. Shepard, "Multidimensional scaling, tree-fitting, and clustering" *Science* Vol. 210 (4468): 390—398, 1980.

- [139] P. Moravec, and V. Snasel, "Dimension reduction methods for image retrieval". In. *ISDA '06. IEEE Computer Society*, Washington, DC, pp. 1055-1060.
- [140] J. Ye, R. Janardan, and Q. Li, "GPCA: an efficient dimension reduction scheme for image compression and retrieval". In *KDD '04*. ACM, New York, NY, pp. 354-363.
- [141] E. Bingham, and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data". In *KDD '01*. ACM, New York, NY, pp. 245-250.
- [142] A. Antoniadis, S. Lambert-Lacroix, F. Leblanc, F. "Effective dimension reduction methods for tumor classification using gene expression data". *Bioinformatics*, V. 19, 2003, pp. 563–570
- [143] J. Harsanyi and C.-I Chang, "Hyper-spectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sensing*, vol. 32, July 1994, pp. 779–785.
- [144] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. "Dimensionality reduction: a comparative review". *Tilburg University Technical Report*, TiCC-TR 2009-005, 2009.
- [145] C. Eckart, G. Young, "The approximation of one matrix by another of lower rank", *Psychometrika* 1 (3): 1936, pp 211–218.
- [146] D. N. Lawley, and A. E. Maxwell. "*Factor analysis as a statistical method*". 2nd Ed. New York: American Elsevier, 1971.
- [147] X. He and P. Niyogi, "Locality Preserving Projections," In *NIPS 2003*. V 16. pp 153-160.
- [148] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [149] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *ICANN97*, Springer Lecture Notes in Computer Science, pp 583, 1997.

- [150] L.K. Saul, K.Q. Weinberger, F. Sha, J. Hamm and D. D. Lee. Spectral methods for dimensionality reduction. In *Semi-supervised Learning*, 2006. pp. 293-308
- [151] B. Schölkopf, A.J. Smola, and K.-R. Muller. “Nonlinear component analysis as a kernel eigenvalue problem”. *Neural Computation*, 10(5):1299–1319, 1998.
- [152] Y. Bengio, J.-F. Paiement, P. Vincent . “Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering”. In *NIPS 2004* V 16.
- [153] N.M. Rajpoot, M. Arif, and A.H. Bhalerao. Unsupervised learning of shape manifolds. In *BMVC 2007*.
- [154] H. Choi and S. Choi. “Robust kernel Isomap”. *Pattern Recognition*, 40(3):853–862, 2007.
- [155] V. de Silva and J.B. Tenenbaum. “Global versus local methods in nonlinear dimensionality reduc. In *NIPS 2003*, V15, pp 721–728
- [156] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura. “On the use of Kernel PCA for feature extraction in speech recognition”. *IEICE Transactions on Information Systems*, E87-D(12):2802–2811, 2004.
- [157] K.Q. Weinberger, F. Sha, Q. Zhu et and L.K. Saul “Graph Laplacian regularization for large-scale semi-definite programming”. In *NIPS 2007*, V19. pp 1487-1495
- [158] M. Welling, M. Rosen-Zvi and G. Hinton. “Exponential family harmoniums with an application to IR”. In *NIPS 2004*, V 17, pp1481–1488.
- [159] Matlab Toolbox for Dimensionality Reduction [Online] Available: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html [Accessed: June 22, 2010]
- [160] Christos Faloutsos, ”*Searching Multimedia Databases by Content*”, Springer; 1 edition (September 30, 1996)
- [161] S. Berchtold, D. Keim, H.P. Kriegel. “The X-Tree: An Index Structure for High-Dimensional Data”, *22nd Conf. on Very Large Databases*, Bombay, India, pp. 28-39. 1996.

- [162] Adapted from the Wikimedia Commons file "File:R-tree.svg" [Online] Available: <http://en.wikipedia.org/wiki/File:R-tree.svg> [Accessed: March 20, 2011]
- [163] T. Sellis, N. Roussopoulos, C. Faloutsos. "The R+-Tree: A Dynamic Index for Multi-Dimensional Objects", *Proc. 13th Int. Conf. on Very Large Databases*, Brighton, England, pp. 507-518. 1987.
- [164] Adapted from the Wikimedia Commons file "File:Kdtree_2d.svg" [Online] Available: http://en.wikipedia.org/wiki/File:Kdtree_2d.svg [Accessed: March 20, 2011]
- [165] J.E. Goodman, J. Rourke and P. Indyk "Chapter 39 : Nearest neighbours in high-dimensional spaces". *Handbook of Discrete and Computational Geometry (2nd ed.)*. 2004
- [166] D.A White, R. Jain. "Similarity indexing with the SS-tree", *Proc. 12th Int. Conf on Data Engineering*, New Orleans, LA, 1996.
- [167] D. Comer. "Ubiquitous B-Tree.", *ACM Comput. Surv.* 11, 2 (Jun. 1979), pp. 121-137. 1979
- [168] Adapted from the Wikimedia Commons file "File:Bplustree.png" [Online] Available: <http://en.wikipedia.org/wiki/File:Bplustree.png> [Accessed: March 20, 2011]
- [169] B. C. Ooi, K. L. Tan, C. Yu, and S. Bressan. "Indexing the edge: a simple and yet efficient approach to high-dimensional indexing". In *Proc. 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 166-174. 2000.
- [170] V. Markl. "MISTRAL: Processing Relational Queries using a Multidimensional Access Technique". Ph.D Thesis. Der Technischen Universität München. 1999.
- [171] R. Zhang, P. Kalnis, B. C. Ooi, K. Tan. "Generalized Multi-dimensional Data Mapping and Query Processing". *ACM Transactions on Data Base Systems (TODS)*, 30(3): pp. 661-697, 2005.
- [172] D. Hoiem, A. Efros, and M. Hebert. "Putting objects in perspective". In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

- [173] I. Laptev. "Improvements of object detection using boosted histograms". In *Proceedings of the 17th British Machine Vision Conference (BMVC)*, 2006.
- [174] M. Mahdavian. "Fast object class recognition". BSc. Thesis, University of British Columbia, 2005.
- [175] M. Marszalek and C. Schmid. "Spatial weighting for bag-of-features". In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [176] K. Mikolajczyk, B. Leibe, and B. Schiele. "Multiple object class detection with a generative model". In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [177] F. Moosmann, D. Larlus, and F. Jurie. "Learning saliency maps for object categorization". In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, 2006.
- [178] P.B Thawari and N. J. Janwe, "CBIR Based on Color and Texture", *International Journal of Information Technology and Knowledge Management*, January-June 2011, Volume 4, No. 1, pp. 129-132
- [179] G-H. Cha and C-W. Chung, "Primary Title: A New Indexing Scheme for Content-Based Image Retrieval". In *Multimedia Tools and Applications* V 6-3 pp. 263-288, 1998.
- [180] A. Ahmadian, A. Mostafa, M.D. Abolhassani, Y.Salimpour, "A texture classification method for diffused liver diseases using Gabor wavelets", In: *Engineering in Medicine and Biology Society*, 2005. IEEE-EMBS 2005, pp.1567-1570. 2005

APPENDICES

APPENDIX A

PUBLICATIONS AND PERMISSIONS

Publications contained in this dissertation are as follows. In most cases this dissertation extends these publications.

- J.M Banda and R. Angryk “Selection of Image Parameters as the First Step Towards Creating a CBIR System for the Solar Dynamics Observatory”. International Conference on Digital Image Computing: Techniques and Applications (DICTA). Sydney, Australia, December 1-3, 2010. pp. 528-534.
- J.M Banda and R. Angryk “Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis”. NASA Conference of Intelligent Data Understanding (CIDU 2010). Computer History Museum, Mountain View, CA October 5th - 6th, 2010. pp. 189-203.
- J.M Banda and R. Angryk “An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image categorization” Proceedings of the twenty-third international Florida Artificial Intelligence Research Society conference (FLAIRS-23), Daytona Beach, Florida, USA, May 19–21 2010. pp. 380-385.
- J.M Banda and R. Angryk “On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images” Proceedings of the 18th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '09), Jeju Island, Korea, August 2009, pp. 2019-2024.

APPENDIX B

SOURCE CODE

Source code for all chapters in this dissertation can be found at:

<http://www.jmbanda.com/Dissertation/>