
**MULTIPLE INSTANCE CHOQUET INTEGRAL
FOR MULTIRESOLUTION SENSOR FUSION**

A Dissertation presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
XIAOXIAO DU
Dr. Alina Zare, Dissertation Supervisor
DECEMBER 2017

© Copyright by XIAOXIAO DU 2017

All Rights Reserved

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

MULTIPLE INSTANCE CHOQUET INTEGRAL
FOR MULTIRESOLUTION SENSOR FUSION

presented by Xiaoxiao Du,

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Alina Zare

Dr. James Keller

Dr. Dominic Ho

Dr. Marjorie Skubic

Dr. Mihail Popescu

To my family, all my love.

To Dr. Alina Zare, with warmest wishes.

ACKNOWLEDGMENTS

I would like to thank my dissertation adviser and committee chair, Dr. Alina Zare, for her wholehearted guidance, support, and all the opportunities she provided me throughout my studies and research. I am inspired by her dedication, passion, and creativity in research and teaching; I admire and respect her confidence, intelligence, and strength of character.

I would also like to thank my doctoral committee members, Dr. James Keller, Dr. Dominic Ho, Dr. Marjorie Skubic, and Dr. Mihail Popescu, for their help and valuable suggestions. I gained knowledge and experience in the lectures, seminars, and discussions, for which I am grateful.

Thank you to those who shed light on the field of multiple instance learning, computational intelligence, and sensor fusion. Thank you to those who make the data sets used in this dissertation available.

I am thankful for the University of Missouri and Zhejiang University, for preparing me for the journey. I am thankful for my current and former teachers and professors, for their help and support along the way.

I am thankful for my labmates, for the times they provided insightful comments as well as encouragements in my studies and in life. I am grateful for my friends, for all the music and laughter.

Finally, a warm thank you goes to my parents and all of my family, for their love, support, understanding, and inspiration, as always.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	vi
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS AND ACRONYMS	xvii
LIST OF SYMBOLS AND NOTATIONS	xix
ABSTRACT	xx
CHAPTER	
1 Introduction	1
2 Literature Review	6
2.1 Multiple Instance Classification	6
2.2 Multiple Instance Regression	13
2.3 Fuzzy Measure and Choquet Integral	19
2.3.1 Fuzzy Measure	20
2.3.2 Choquet Integral	22
2.3.3 Learning The Fuzzy Measure	24
2.4 Sensor Fusion	32
2.4.1 Co-registration	33
2.4.2 Multi-resolution Fusion	38
2.4.3 Fusion of Mixed Data Types	40

2.5	Summary and Discussion of Literature Review	42
3	Multiple Instance Choquet Integral	45
3.1	Noisy-or Model	46
3.2	Min-Max Model	47
3.3	Generalized Mean Model	48
3.4	Optimization	50
3.4.1	Measure Initialization	50
3.4.2	Evaluation of Valid Intervals	52
3.4.3	Mutation	53
3.4.4	Selection	54
4	Multiple Instance Choquet Integral Regression	56
5	Multi-Resolution Multiple Instance Choquet Integral	58
6	Experimental Results	62
6.1	Classification	62
6.1.1	Synthetic 3-Source Classification Data Set	63
6.1.2	Synthetic Lane-Based Target Detection Data Set	64
6.1.3	Synthetic 5-Source Classification Data Set For Varying Parameters .	67
6.1.4	MUUFL Gulfport Target Detection	69
6.2	Multi-Resolution Fusion Data Set	104
6.2.1	Synthetic Multi-Resolution Fusion Data Set	104
6.2.2	MUUFL Gulfport Building Detection – Sub-image	107
6.2.3	MUUFL Gulfport Scene Understanding: Building, Sidewalk and Road	108

6.2.4	Soybean and Weed Data Set	115
6.3	Regression Data Set	137
6.3.1	Synthetic Regression Data Set	137
6.3.2	Crop Yield Data Set	139
6.4	Discussion on Optimization Schemes	144
6.4.1	“Top-Down” and “Bottom-Up” Initialization	144
6.4.2	Sampling according to measure element used	145
6.4.3	Using a binary fuzzy measure	148
7	Conclusion	157
APPENDIX		
A	Truncated Gaussian Sampling Method	159
BIBLIOGRAPHY		162
VITA		202

LIST OF TABLES

Table	Page
6.1 Mean and standard deviation of estimated and true measure element values learned by MICI noisy-or model for synthetic 3-source MICI classification data set over three runs.	65
6.2 Mean and the standard deviation (in parentheses) of estimated measure element values learned for synthetic 5-source lane-based classification data set over five runs.	68
6.3 The positive detection and false alarm rate of the synthetic lane-based target detection data set after five-fold cross validation across five runs.	70
6.4 Relative error versus contamination for synthetic classification data set for MICI noisy-or model across five runs.	70
6.5 Relative error versus contamination for synthetic classification data set for MICI min-max model across five runs.	71
6.6 Relative error versus contamination for synthetic classification data set for MICI generalized mean model across five runs.	71
6.7 The AUC results at on un-normalized MUUFL Gulfport data across five runs.	77

6.8	The AUC results at on normalized MUUFL Gulfport data across five runs. Normalized by dividing over the norm of the data.	77
6.9	The AUC results at on normalized MUUFL Gulfport data across five runs. Normalized by unity-based normalization.	78
6.10	The AUC results at on normalized MUUFL Gulfport data across five runs. Normalized by the mean and standard deviation.	79
6.11	Running time (seconds) and number of iterations until convergence for MICI models comparison.	79
6.12	One example estimated measure element values learned for synthetic 5- source multi-resolution classification data set after one run.	106
6.13	The AUC results of building, sidewalk, and road detection using MUUFL Gulfport HSI and LiDAR data.	114
6.14	The RMSE results of MICI and MR-MICI on building, sidewalk, and road detection.	114
6.15	The AUC and RMSE results of MICI and MR-MICI on building detection, scored on edges. Train on campus 1 and test on campus 2.	115
6.16	The AUC and RMSE results of MICI and MR-MICI on building detection, scored on edges. Train on campus 2 and test on campus 1.	115
6.17	Relative error versus percentage of primary instances for synthetic regres- sion data set for MICI Regression model across five runs.	138
6.18	Relative error versus SNR for synthetic regression data set MICI Regres- sion model across five runs.	139
6.19	Number of counties (bags) with both corn and wheat yield in the crop yield data set.	140

6.20 RMSE error for CA corn and wheat yield, Training on Years 2001-2004, Test on Year 2005.	140
6.21 RMSE error for KS corn and wheat yield, Training on Year 2001-2004, Test on Year 2005.	141
6.22 Running time (seconds) and number of iterations until convergence for op- timization schemes comparison.	147
6.23 Running time (seconds) and number of iterations until convergence for MICI models comparison with binary measures.	152

LIST OF FIGURES

Figure	Page
2.1 Illustration of bags in multiple instance learning.	7
2.2 Illustration of standard supervised classification, multiple instance learning classification and embedded multiple instance learning classification.	8
2.3 An illustration for the subset and superset relationships between fuzzy measure elements given four sources.	21
5.1 Illustration for HSI and LiDAR fusion.	60
6.1 Synthetic 3-source dataset and results for MICI classifier fusion model. . . .	64
6.2 Colorbar.	65
6.3 One example of the synthetic lane-based target detection data set.	66
6.4 RX detection output (plotted horizontally) of the synthetic lane-based target detection data set.	67
6.5 Relationship of fitness values vs. number of iterations in the synthetic lane-based target detection experiment.	69
6.6 The RGB image from MUUFL Gulfport “campus 3” data set.	72
6.7 The spectral signatures for brown, dark green, FVG, and pea green targets in the MUUFL Gulfport data set. The unit for the wavelength is nanometers.	72

6.8 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were un-normalized.	80
6.9 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were un-normalized.	81
6.10 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were un-normalized.	82
6.11 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were un-normalized.	83
6.12 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were un-normalized.	84
6.13 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were un-normalized.	85
6.14 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by dividing over the norm of the data.	86
6.15 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by dividing over the norm of the data.	87
6.16 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by dividing over the norm of the data.	88
6.17 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by dividing over the norm of the data.	89

6.18 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by dividing over the norm of the data.	90
6.19 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by dividing over the norm of the data.	91
6.20 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by unity-based normalization.	92
6.21 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by unity-based normalization.	93
6.22 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by unity-based normalization.	94
6.23 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by unity-based normalization.	95
6.24 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by unity-based normalization.	96
6.25 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by unity-based normalization.	97

6.26 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by the mean and standard deviation method.	98
6.27 ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by the mean and standard deviation method.	99
6.28 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by the mean and standard deviation.	100
6.29 ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by the mean and standard deviation.	101
6.30 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by the mean and standard deviation method.	102
6.31 ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by the mean and standard deviation method.	103
6.32 Groundtruth for synthetic 5-source dataset for MR-MICI fusion experiments.	104
6.33 One example for synthetic 5-source dataset for MR-MICI fusion experiments.	105
6.34 Three subimages of buildings in MUUFL Gulfport campus 1 data set. . . .	108
6.35 Results of building classification, train on sub-image 1, test on sub-image 1.	117
6.36 ROC curve results of building classification results, train on sub-image 1, test on sub-image 1.	118

6.37 Results of building classification, train on sub-image 1, test on sub-image 3.	119
6.38 ROC curve results of building classification results, train on sub-image 1, test on sub-image 3.	120
6.39 Four LiDAR lines in MUUFL Gulfport data, shown in Google Earth.	121
6.40 RGB image of MUUFL Gulfport data.	121
6.41 Scatterplot of LiDAR line 1 point cloud in MUUFL Gulfport campus 1 data.	122
6.42 Raster image of the first return MUUFL Gulfport LiDAR data.	122
6.43 Open Street Map imagery over MUUFL Gulfport campus 1.	123
6.44 The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for building detection.	124
6.45 The building signature for ACE detector and the ACE detection map for the MUUFL Gulfport HSI data.	124
6.46 The histogram and peaks of the LiDAR values of building points.	125
6.47 The LiDAR confidence maps for building detection in the MUUFL Gulf- port HSI data.	125
6.48 The fusion test confidence maps for building detection in the MUUFL Gulf- port HSI data for SVM and min methods. Train on campus 1 and test on campus 2.	126
6.49 The fusion test confidence maps for building detection in the MUUFL Gulf- port HSI data for taking the max and mean of the sources. Train on campus 1 and test on campus 2.	126
6.50 The fusion test confidence maps for building detection in the MUUFL Gulf- port HSI data for the mi-SVM and CI-QP methods. Train on campus 1 and test on campus 2.	127

6.51 The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for the proposed MICI and MR-MICI methods. Train on campus 1 and test on campus 2.	127
6.52 The Overall ROC curve for building detection for MUUFL Gulfport data.	128
6.53 The difference map in the MUUFL Gulfport HSI data between LiDAR points picked by MR-MICI and mean of the LiDAR points versus rasterized LiDAR imagery.	129
6.54 The difference map in the MUUFL Gulfport HSI data between min and max of the LiDAR points and rasterized LiDAR imagery.	129
6.55 The ROC curve for building detection for MUUFL Gulfport data, scored on the difference map between LiDAR edge map and mean maps. Train on Campus 1, test on Campus 2.	130
6.56 The ROC curve for building detection for MUUFL Gulfport data, scored on the difference map between min and max maps. Train on Campus 1, test on Campus 2.	130
6.57 The ROC curve for building detection for MUUFL Gulfport data, scored on the difference map between LiDAR edge map and mean maps. Train on Campus 2, test on Campus 1.	131
6.58 The ROC curve for building detection for MUUFL Gulfport data, scored on the difference map between min and max maps. Train on Campus 2, test on Campus 1.	131
6.59 The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for sidewalk detection.	132

6.60 The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for road detection.	132
6.61 The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for the proposed MICI and MR-MICI methods. Train on campus 1 and test on campus 2.	133
6.62 The Overall ROC curve for sidewalk detection for MUUFL Gulfport data. .	133
6.63 The Overall ROC curve for road detection for MUUFL Gulfport data. . .	134
6.64 The RGB image of weed in the soybean-weed data.	134
6.65 The height map of the soybean-weed data.	134
6.66 The L-band image of the soybean-weed data.	134
6.67 The B-band image of the soybean-weed data.	135
6.68 The Gabor filtered image of the soybean-weed data height map.	135
6.69 The Ground Truth map of weed in the soybean-weed data.	135
6.70 The SLIC segmentation map of the soybean-weed data.	135
6.71 The ROC curve for weed detection in the soybean-weed data.	136
6.72 The confidence map obtained from MICI fusion for the soybean-weed data. .	136
6.73 The confidence map obtained from the MR-MICI fusion for the soybean-weed data.	136
6.74 Contamination data set for MICI Regression model experiments.	138
6.75 The relationship between the Gaussian kernel width and RVM RMSE. . . .	142
6.76 The relationship between the Gaussian kernel width and Aggregate MIR RMSE.	143
6.77 The relationship between the cluster number and Cluster MIR RMSE. . . .	144

6.78 Comparison of ROC curve performance using top-down initialization and bottom-up initialization.	153
6.79 Comparison of the two optimization schemes: sampling by measure element or sampling according to the valid intervals.	154
6.80 An illustration for the subset and superset relationships between binary fuzzy measure elements given four sources.	155
6.81 Relationship of fitness values vs. number of iterations for MICI binary measure models.	156

LIST OF ABBREVIATIONS AND ACRONYMS

AHI	Advanced Himawari Imager
BFM	binary fuzzy measure
CI	Choquet integral
DD	Diverse Density
DEM	digital elevation model
DSM	digital surface model
EM	Expectation-Maximization
EMI	electro-magnetic induction (sensor)
FAR	false alarm rate
GA	genetic algorithm
GIS	geographic information systems
GPR	ground penetrating radar
GPS	Global Positioning System
GSD	ground sample distance
HSI	hyperspectral image
IR	infrared (sensor)
kNN	K -nearest neighbor
LiDAR or LIDAR	Light Detection and Ranging
LCFI	linguistic Choquet fuzzy integral
MAP	maximum a posteriori probability
MI	Mutual Information
MICI	Multiple Instance Choquet Integral

MIL	Multiple Instance Learning
MILES	Multiple-Instance Learning via Embedded Instance Selection
MIR	Multiple Instance (Multiple) Regression
MKL	Multiple Kernel Learning
MR-MICI	Multi-Resolution Multiple Instance Choquet Integral
MS	multispectral (imagery)
PAN	panchromatic (imagery)
PSO	particle swarm optimization
RFC-MIR	Robust Fuzzy Clustering Multiple Instance Regression
ROC	receiver operating characteristic (curve)
SAR	synthetic aperture radar
SVM	support vector machine

LIST OF SYMBOLS AND NOTATIONS

g	A fuzzy measure
B	Total number of bags
m	Total number of sources (e.g. classifiers) to be fused
N	Total number of data points
N_h	Total number of source combinations in multiresolution fusion
$C_g(\cdot)$	The Choquet integral output on an input computed with fuzzy measure g
P	Measure population size in the evolutionary algorithm
I	Maximum number of iterations in the evolutionary algorithm
\mathbf{F}_P^0	Fitness values for all measures in the initial population in the evolutionary algorithm
\mathbf{F}_P^t	Fitness values for all measures in Iteration t
F^*	Best (highest) current fitness value
g^*	Best current measure with the highest fitness value
\mathcal{G}	All measures in the current measure population
$\mathcal{G}\{p\}$	The p^{th} measure in measure population \mathcal{G}
η	Rate of small-scale mutation

ABSTRACT

Imagine you are traveling to Columbia, MO for the first time. On your flight to Columbia, the woman sitting next to you recommended a bakery by a large park with a big yellow umbrella outside. After you land, you need directions to the hotel from the airport. Suppose you are driving a rental car, you will need to park your car at a parking lot or a parking structure. After a good night's sleep in the hotel, you may decide to go for a run in the morning on the closest trail and stop by that recommended bakery under a big yellow umbrella. It would be helpful in the course of completing all these tasks to accurately distinguish the proper car route and walking trail, find a parking lot, and pinpoint the yellow umbrella.

Satellite imagery and other geo-tagged data such as Open Street Maps provide effective information for this goal. Open Street Maps can provide road information and suggest bakery within a five-mile radius. The yellow umbrella is a distinctive color and, perhaps, is made of a distinctive material that can be identified from a hyperspectral camera. Open Street Maps polygons are tagged with information such as “parking lot” and “sidewalk.” All these information can and should be fused to help identify and offer better guidance on the tasks you are completing.

Supervised learning methods generally require precise labels for each training data point. It is hard (and probably at an extra cost) to manually go through and label each pixel in the training imagery. GPS coordinates cannot always be fully trusted as a GPS device may only be accurate to the level of several pixels. In many cases, it is practically infeasible to obtain accurate pixel-level training labels to perform fusion for all the imagery and maps available.

Besides, the training data may come in a variety of data types, such as imagery or as

a 3D point cloud. The imagery may have different resolutions, scales and, even, coordinate systems. Previous fusion methods are generally only limited to data mapped to the same pixel grid, with accurate labels. Furthermore, most fusion methods are restricted to only two sources, even if certain methods, such as pan-sharpening, can deal with different geo-spatial types or data of different resolution. It is, therefore, necessary and important, to come up with a way to perform fusion on multiple sources of imagery and map data, possibly with different resolutions and of different geo-spatial types with consideration of uncertain labels.

I propose a Multiple Instance Choquet Integral framework for multi-resolution multi-sensor fusion with uncertain training labels. The Multiple Instance Choquet Integral (MICI) framework addresses uncertain training labels and performs both classification and regression. Three classifier fusion models, i.e. the noisy-or, min-max, and generalized-mean models, are derived under MICI. The Multi-Resolution Multiple Instance Choquet Integral (MR-MICI) framework is built upon the MICI framework and further addresses multi-resolution in the fusion sources in addition to the uncertainty in training labels. For both MICI and MR-MICI, a monotonic normalized fuzzy measure is learned to be used with the Choquet integral to perform two-class classifier fusion given bag-level training labels. An optimization scheme based on the evolutionary algorithm is used to optimize the models proposed. For regression problems where the desired prediction is real-valued, the primary-instance assumption is adopted.

The algorithms are applied to target detection, regression and scene understanding applications. Experiments are conducted on the fusion of remote sensing data (hyperspectral and LiDAR) over the campus of University of Southern Mississippi - Gulfpark. Cloth-panel sub-pixel and super-pixel targets were placed on campus with varying levels of oc-

clusion and the proposed algorithms can successfully detect the targets in the scene. A semi-supervised approach is developed to automatically generate training labels based on data from Google Maps, Google Earth and Open Street Map. Based on such training labels with uncertainty, the proposed algorithms can also identify materials on campus for scene understanding, such as road, buildings, sidewalks, etc. In addition, the algorithms are used for weed detection and real-valued crop yield prediction experiments based on remote sensing data that can provide information for agricultural applications.

Chapter 1

Introduction

Multi-sensor fusion methods aim to combine and integrate information obtained from multiple sensor sources while reducing uncertainties in the data and providing more detailed information [1, 2]. Each of the sensor sources may provide complementary and reinforcing information that is helpful in applications such as target detection, classification, or scene understanding [3]. Take remote sensing applications for example, imagery can be obtained from multiple sensors such as hyperspectral and LiDAR (light detection and ranging). Hyperspectral imaging sensors can provide spectral information about materials in the scene across a wide range of wavelengths while LiDAR data can provide height information. If a road and a building rooftop are built with the same material (say, asphalt), hyperspectral information alone may not be sufficient to tell them apart. However, LiDAR data provides height information and can easily distinguish the two. On the other hand, a highway and a biking trail can be at the same elevation and using LiDAR data alone may not be sufficient to distinguish the two types of roads, but hyperspectral sensor can be very helpful in identifying the distinctive spectral characteristics between a highway, which is likely to

be primarily asphalt, and a biking trail, which is likely to be covered in dirt. It would be, thus, valuable to take advantage and fuse the information provided by multiple sensors in order to produce more detailed information, such as a better classification result or a more comprehensive understanding of the scene [2, 4–6].

The Choquet integral (CI) has a long history of being an effective aggregation operator for non-linear fusion [3, 7–10]. A discrete Choquet integral integrates the input sources with respect to a fuzzy measure [11]. Compared with commonly used aggregation operators such as weighted arithmetic means [12], the Choquet integral is able to model the relationship amongst the combinations of the sources and can flexibly represent a wide variety of aggregation operators [7, 13]. In this dissertation, the monotonic normalized discrete Choquet integral [14] is used as the aggregation operator for sensor fusion.

The standard Choquet integral fusion method assumes that (1) the data to be fused are homogeneous (of the same data type and with the same resolution) and (2) there are training labels available for each data point. It is generally assumed that the data provided by multiple sources for fusion must be of the same type, of the same resolution, on the same grid, or be possible to match and link individual data points together if the data types are arbitrary/heterogeneous [15]. That is to say, the standard CI fusion method requires data from m different sensors produce data that has a one-to-one correspondence, or that some form of pre-processing is needed to transform all sources to the same resolution and perform matching. In addition, the data has to be on the same grid or scale, so pixel (i, j) in sensor image data 1 should correspond to pixel (i, j) in sensor image data m exactly. The image data values X_{ij}^k , $k = \{1, \dots, m\}$ as well as its classification or regression labels l_{ij} are also assumed to be known for each training pixel/data point in the image [16].

This assumption raises two problems. First, the assumption about homogeneous data

source types does not generally hold in real applications for sensor fusion. Existing optical sensors operate on varying spatial and spectral and temporal resolutions [17] and it is not necessarily feasible to convert all data to the same resolution or map to the same grid. Techniques such as rasterization and image registration [18, 19] have been proposed to perform co-registration of heterogeneous geospatial information sources. However, rasterized images may lose some of its original information. Suppose a hyperspectral imaging (HSI) camera scans the scene and provides an image with 1-meter ground sample distance (GSD) [20]. That means each pixel in the hyperspectral image covers $1 \times 1m^2$ area. LiDAR technology, on the other hand, produces point clouds by densely sample the surface of the earth in the scene. There can be several LiDAR data points inside an $1 \times 1m^2$ area. One way to rasterize LiDAR data is by projecting the LiDAR data points into the pixel coordinate plane of the HSI image [21] and obtain one single LiDAR value (Z coordinate value, usually height information) for each pixel. Then, data fusion between the HSI and rasterized LiDAR data can be performed pixel-by-pixel on the image grid. However, the dense LiDAR point cloud data can offer higher geographic accuracy [22] as the data does not depend on grid size. Each LiDAR data point provides a certain degree of information and it would be nice to take all available information into account. Besides, most image registration methods rely heavily on both the accuracy of input images and registration parameters [18, 23] and can also bring in another layer of uncertainty regarding geometric misalignment and mismatch [15, 21].

Second, even assuming that homogeneous data is available for fusion or that there is a noiseless way to transform heterogeneous data, standard supervised learning methods require accurate labels for each training data point. However, data-point specific labels are often unavailable or difficult and expensive to obtain [10]. For target detection in real

remote sensing applications, for example, some Global Positioning System (GPS) device may only be accurate to the level of several meters [24]. Depending on the GSD of the imagery, the target ground truth locations in the scene measured by a GPS can only be accurate to the level of several pixels. It is, thus, difficult to pinpoint accurate pixel-level target locations and provide accurate training labels.

To address these two problems, this dissertation proposes a Multiple Instance Choquet Integral (MICI) framework for both multi-sensor classifier fusion and regression that can deal with uncertainty in training labels. Three variations of the objective functions based on three models, i.e. noisy-or, min-max and generalized-mean models, were derived for the MICI classifier fusion framework. A monotonic normalized fuzzy measure is learned to be used with the Choquet integral to perform two-classs classifier fusion given bag-level training labels. An optimization scheme using an evolutionary algorithm is used to optimize the models proposed. For regression problems where the desired prediction is real-valued, this dissertation adopts the “primary-instance” assumption that there is one primary instance responsible for the label for each bag [25] and proposes a Multiple Instance Choquet Integral Regression model that can fuse multiple sources with real-valued label as well as taking into account the uncertainties in the label.

Furthermore, this dissertation proposes a Multi-Resolution Multiple Instance Choquet Integral (MR-MICI) framework that takes in heterogeneous data (for example, images at different resolutions or data of different geospatial types) from multiple sensors with uncertainty in training labels and perform fusion. The proposed MR-MICI algorithm works under the assumption that at least one point in each bag from each data source is accurate but the remaining data points in the bags can have uncertain labels. The proposed MR-MICI algorithm also learns a monotonic normalized fuzzy measure to be used with the

Choquet integral to perform fusion on heterogeneous data sources.

Experiments were conducted based on the proposed algorithms on both synthetic data and real applications such as target detection and scene understanding in remote sensing imagery. Results indicate that the proposed MICI framework can successfully learn a set of fuzzy measure to be used with the Choquet integral to effectively perform classifier fusion and regression while dealing with uncertainties in training labels. Results also indicate that the proposed MR-MICI framework can successfully perform classifier fusion and yield better classification accuracy on heterogeneous data (multi-resolution data and data of different geospatial types such as HSI and LiDAR) while taking into account uncertain labels.

Chapter 2

Literature Review

This chapter provides a literature review on Multiple Instance Learning frameworks including MIL classification and MIL regression. This chapter also provides definitions for fuzzy measures and the Choquet integral and reviews previous methods in learning a fuzzy measure. This chapter also reviews the existing literature in sensor fusion, especially fusion with the Choquet integral and fusion of heterogeneous data types and multiresolution fusion.

2.1 Multiple Instance Classification

The Multiple Instance Learning (MIL) framework was first proposed in [26] to address uncertainty and inaccuracy in labeled data in supervised learning. In the MIL framework, training labels are associated with sets of data points (“bags”) instead of each data point (“instance”). In the scenario of two-class classification, the standard MIL assumes that a bag is labeled positive if at least one instance in the bag is positive and a bag is labeled

negative if all the instances in the bag are negative. Figure 2.1 shows the illustration of MIL bags. Assuming the red-colored points are positive instances and blue-colored points are negative instances. The two bags on the top row are, therefore, negative bags, as all the instances in the bags are negative. The three bags on the bottom row are positive bags, as there is at least one positive instance in the bags. A more generalized view of the MIL was also introduced in the literature [27]. The generalized MIL does not follow the restriction of the standard MIL assumption [28, 29]. The standard MIL assumes that a bag’s label depends on the labels of the instances in the bag, while the generalized MIL assumes that a bag’s label depends on the “interaction” between instances in the bag [28, 29]. In this dissertation, only the standard MIL assumption is discussed as it fits the task of sensor fusion at hand.

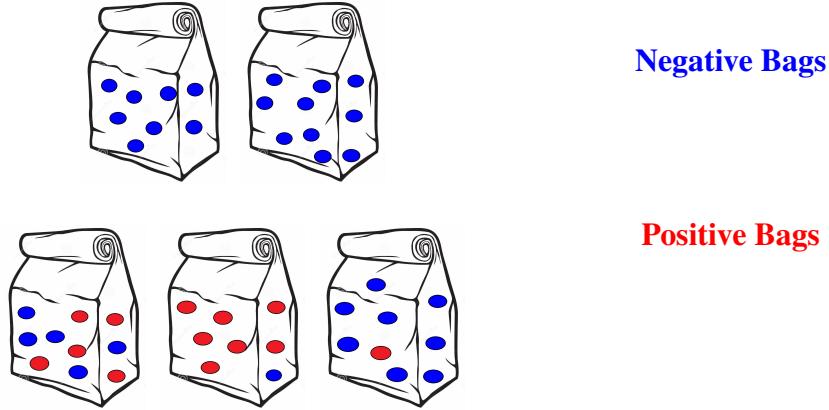


Figure 2.1: Illustration of bags in multiple instance learning. Red color marks positive instances and blue color marks negative instances. The two bags on the top row are negative bags and the three bags on the bottom row are positive bags.

Figure 2.2 illustrates the differences between standard supervised classification, multiple instance learning classification and embedded multiple instance learning classification based on the paradigms discussed in [30]. The standard supervised classification learns a classifier based on a set of training feature vectors, where each feature vector has an asso-

ciated class label (such as marked in orange or blue color in Figure 2.2) [30]. The instance-space paradigm classifies instances based on their values at the instance level and draws a decision boundary between classes. The bag-space paradigm discriminates information at the bag-level. Distances between bags are computed and a standard distance-based classifier may be applied such as the K -nearest neighbor (KNN) classifier [30, 31]. In the embedded-space paradigm, each bag is often mapped into a high-dimensional space. Each feature vector in the high-dimensional space for each bag represents information from the entire bag and all the instances in the bag. A discriminative classifier is then applied to the feature vectors in the embedded space; thus classifying entire bags. Selected notable MIL classification methods are discussed in detail as follows.

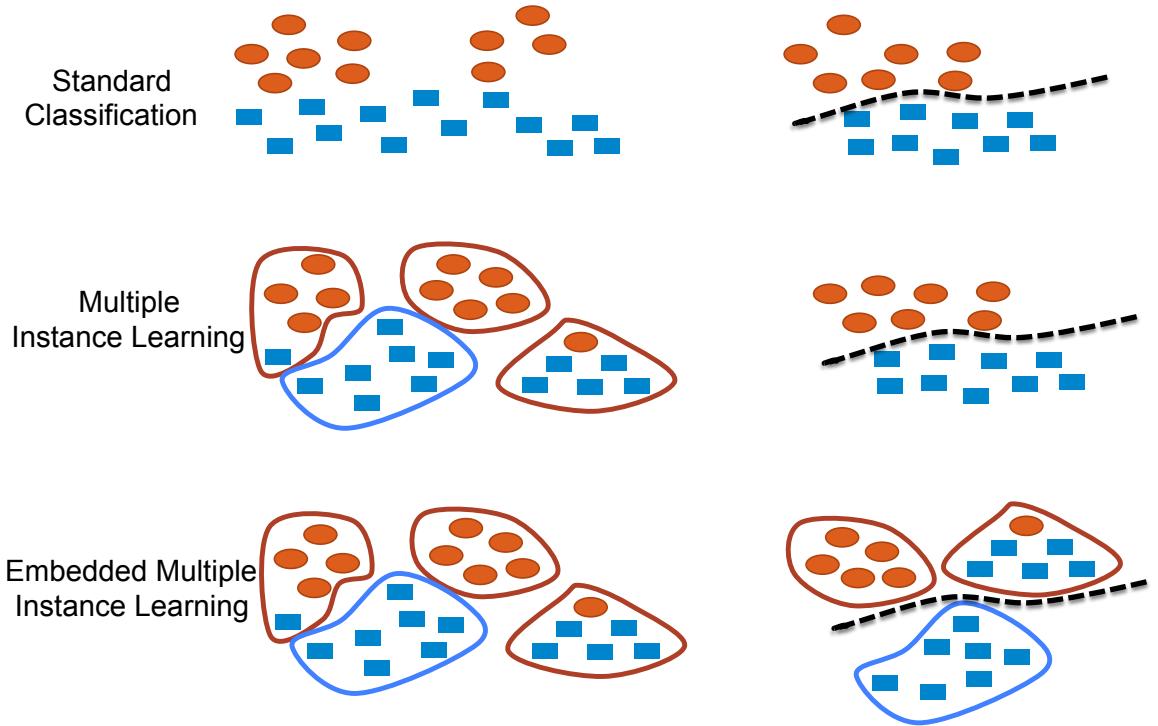


Figure 2.2: Illustration of standard supervised classification, multiple instance learning classification and embedded multiple instance learning classification. The left column is training stage and the right column is testing stage. Orange color marks positive instances/bags and blue color marks negative instances/bags in the feature space.

The EM-DD technique [32] combines the Expectation-Maximization (EM) method [33] and the Diverse Density (DD) approach [13, 26, 34]. For two-class classification problems, the maximum Diverse Density is defined as [13, 34]:

$$\arg \max_x \prod_i Pr(x = t | B_i^+) \prod_i Pr(x = t | B_i^-), \quad (2.1)$$

where B_i^+ is the i^{th} positive bag and B_i^- is the i^{th} negative bag, B_{ij} is the individual feature values of the j^{th} instance of the i^{th} positive bag, t is the true concept, and x represents all the points in feature space. Maron et al. [34] proposed a noisy-or model, where for all the instances in each bag,

$$\prod_i Pr(x = t | B_i^+) = 1 - \prod_j (1 - Pr(x = t | B_{ij}^+)), \quad (2.2)$$

and

$$\prod_i Pr(x = t | B_i^-) = \prod_j (1 - Pr(x = t | B_{ij}^-)). \quad (2.3)$$

The causal probability of an individual instance on a potential target was computed based on the distance between them using

$$Pr(x = t | B_{ij}) = \exp(-\|B_{ij} - x\|^2), \quad (2.4)$$

where B_{ij} is the individual feature values of the j^{th} instance of the i^{th} bag.

To extend the DD model, the EM-DD views the relationship between all the instances in the bag and the label of the bag (which instance corresponds to the label of the bag?) as a latent variable that can be estimated using the EM approach [32]. In the E-step, one instance is picked from each bag as the most influential instance for its bag-level label. In the M-step, the DD is maximized by a gradient ascent search and the process is iterated until a stopping criteria is met. Note that EM-DD will terminate after finite number of iterations since there are only finite number of instance-combinations that the algorithm

may pick, but the convergence rate is undetermined [32]. EM-DD scales up well as the bag size increases but the performance will depend on initialization. The EM-DD algorithm can be used for both MI classification and regression [32].

Citation-kNN [35] algorithm uses the Hausdorff distance to compute distance between two bags and assigns bag-level label based on the K Nearest Neighbor rules [36]. The Hausdorff distance for two Bags A and B is defined as:

$$H(A, B) = \max\{h(A, B), h(B, A)\}, \quad (2.5)$$

and

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|, \quad (2.6)$$

where a and b are instances in bags A and B , respectively. A modified version of Hausdorff distance can be computed as follows instead of (2.7):

$$h(A, B) = k_{a \in A}^{\text{th}} \min_{b \in B} \|a - b\|. \quad (2.7)$$

Citation-kNN algorithm then assigns label for bag B_i considering its K nearest neighbors (“references”) and also C bags that counts B_i as a neighbor (“citors”). The Citation-kNN algorithm extends the traditional K Nearest Neighbor to suit multiple instance learning purposes. Extensions of citation-kNN include Bayesian citation-kNN [37] and fuzzy citation-kNN [38, 39].

Andrews et al. [40] proposed two MIL formulations, mi-SVM and MI-SVM, based on support vector machine (SVM) learning approaches. For the mi-SVM two-class classification problem, the objective function is defined as [40]:

$$\min_{\{y_i\}} \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i, \quad (2.8)$$

such that

$$\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \forall i : y_i (< \mathbf{w}, \mathbf{x}_i > + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (2.9)$$

where \mathbf{w} is the weights, b is the bias, y_i is the instance-level label, ξ_i are the slack variables (similar to that of a standard SVM). The labels Y satisfy the MIL assumption. The mi-SVM algorithm learns a linear discriminate function and maximizes the margin to separate the positive from the negative classes based on instance labels. MI-SVM extends based on mi-SVM and defines the objective function as [40]:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_I \xi_I, \quad (2.10)$$

such that

$$\forall I : Y_I \max_{i \in I} (< \mathbf{w}, \mathbf{x}_i > + b) \geq 1 - \xi_I, \xi_I \geq 0, \quad (2.11)$$

where Y_I is the label for bag I . The MI-SVM objective function extends the concept of margin maximization to the bag-level.

Multiple-Instance Learning via Embedded Instance Selection (MILES) is a representative embedded multiple instance learning method [41, 42]. MILES maps both training and testing bag into high-dimensional feature vectors and then performs classification in the mapping space using a one-norm SVM [43]. The distance between an instance \mathbf{x}^k and a bag \mathbf{B}_i is computed as the smallest distance between \mathbf{x}^k and all the instances in bag \mathbf{B}_i [41, 43]:

$$s(\mathbf{x}^k | \mathbf{B}_i) = \max_j \exp \left(-\frac{\|\mathbf{x}_{ij} - \mathbf{x}^k\|^2}{\sigma^2} \right), \quad (2.12)$$

where \mathbf{x}^k is the feature vector for k^{th} instance, \mathbf{B}_i is the i^{th} bag, \mathbf{x}_{ij} is the j^{th} pixel in the i^{th} bag, and σ is a fixed parameter. The feature vector for bag \mathbf{B}_i consists of this similarity measure computed and concatenated over all n instances in the data set.

In this way, a high-dimensional feature vector is constructed based on the distance values. The dimensionality of the feature vector is equal to the number of instances in a data set. The feature vector reflects the similarity between each instance in the data set and each bag. The complete feature mapping \mathbf{m} given all l^+ positive and l^- negative bags is written as [41, 43]:

$$\begin{aligned}
& [\mathbf{m}_1^+, \dots, \mathbf{m}_{l^+}^+, \mathbf{m}_1^-, \dots, \mathbf{m}_{l^-}^-]^T \\
&= [\mathbf{m}(\mathbf{B}_1^+), \dots, \mathbf{m}(\mathbf{B}_{l^+}^+), \mathbf{m}(\mathbf{B}_1^-), \dots, \mathbf{m}(\mathbf{B}_{l^-}^-)]^T \\
&= \begin{bmatrix} s(\mathbf{x}^1, \mathbf{B}_1^+) & \dots & s(\mathbf{x}^1, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^1, \mathbf{B}_1^-) & \dots & s(\mathbf{x}^1, \mathbf{B}_{l^-}^-) \\ s(\mathbf{x}^2, \mathbf{B}_1^+) & \dots & s(\mathbf{x}^2, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^2, \mathbf{B}_1^-) & \dots & s(\mathbf{x}^2, \mathbf{B}_{l^-}^-) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s(\mathbf{x}^n, \mathbf{B}_1^+) & \dots & s(\mathbf{x}^n, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^n, \mathbf{B}_1^-) & \dots & s(\mathbf{x}^n, \mathbf{B}_{l^-}^-) \end{bmatrix}^T, \tag{2.13}
\end{aligned}$$

where the complete mapping for all the bags to all the training instances has dimensionality $NumBags \times NumInstances$.

After feature mapping, a one-norm Support Vector Machine (SVM) is used to perform classification and simultaneously select the most discriminating training instances. The one-norm SVM classifier can be expressed as follows:

$$y = sign \left(\sum_{k=1}^n w_k \mathbf{s}(\mathbf{x}^k, \mathbf{B}_i) + b \right), \tag{2.14}$$

where weights $\mathbf{w} = [w_1, w_2, \dots, w_k, \dots, w_n]^T$ ($k = 1, \dots, n$ for n training instances) and bias b are model parameters. The weights \mathbf{w} and bias b are determined through the following optimization problem:

$$\min_{\mathbf{w}, b, \eta, \xi} \lambda \sum_{k=1}^n |w_k| + c_1 \sum_{i=1}^{l^+} \xi_i + c_2 \sum_{j=1}^{l^-} \eta_j \tag{2.15}$$

with constraints $(\mathbf{w}^T \mathbf{m}_i^+ + b) + \xi_i \geq 1; -(\mathbf{w}^T \mathbf{m}_j^- + b) + \eta_j \geq 1; \xi_i, \eta_j \geq 0$ for $i =$

$1, \dots, l^+$ and $j = 1, \dots, l^-$. In the above formulation, l^+ and l^- are the number of positive bags and negative bags, respectively. \mathbf{m}_i^+ is the mapping for all positive bags and \mathbf{m}_i^- is the mapping for all negative bags. ξ and η are hinge loss parameters that are estimated through the above linear programming problem together with the weights \mathbf{w} and bias b . λ , c_1 and c_2 are scale parameters set by the user. These parameters must satisfy the following constraints: $c_2 = 1 - c_1$ and $0 < c_1 < 1$. These constraints ensure the total training error (i.e., the objective function measure we are trying to minimize through optimization) is associated with a convex combination of training error on both the positive bags and negative bags [41].

After the training, the instances corresponding to non-zero weights, \mathbf{w} , are identified as the “selected” and “discriminative” training samples. These samples will be used during the feature mapping and classification of test data. The usage of the one-norm penalty on the weights is to promote sparsity and drive more elements of \mathbf{w} to zero, thus making the testing process efficient.

Inspired by drug activity prediction problem [26], Multiple Instance Classification has wide applications in natural scene classification [44, 45], human action recognition in videos [46], object detection and tracking [47–49], context identification and context-dependent learning [43, 50], and music information retrieval [51, 52].

2.2 Multiple Instance Regression

Multiple instance regression (MIR) deals with multiple instance problems where the prediction values are real-valued instead of class labels. MIR has been used in the literature for applications such as predicting the ability of antigen peptides to bind to major histo-

compatibility complex class II (MHC-II) molecules [53], predicting aerosol optical depth from remote sensing data [54, 55], and predicting crop yield [55–57].

Ray and Page [25] first proposed an MIR algorithm based on the “primary-instance” assumption, which assumes there is one primary instance in a bag that is responsible for the real-valued bag-level label. A linear regression hypothesis was assumed and the goal is to find a hyperplane $\mathbf{Y} = \mathbf{X}\mathbf{b}$ such that

$$\mathbf{b} = \arg \min_{\mathbf{b}} \sum_{i=1}^n L(y_i, X_{ip}, \mathbf{b}), \quad (2.16)$$

where X_{ip} is the primary instance in bag i , and L is some error function. An Expectation Maximization (EM) algorithm was proposed to solve for the ideal hyperplane. First, a random hyperplane was used for initialization. For each instance j in each bag i , the error L of the instance X_{ij} to the hyperplane $\mathbf{Y} = \mathbf{X}\mathbf{b}$ was computed as $L(y_i, X_{ij}, \mathbf{b}) = (y_i - X_{ij}\mathbf{b})^2$. In the E step, the instance with the lowest error L was selected as the “primary instance.” In the M step, a new hyperplane was constructed by performing a multiple regression over all the instances selected in the E step. The two steps were repeated until the algorithm converges and the best hyperplane solution was returned. Their algorithm was tested using synthetic data sets only but showed benefits of multiple instance regression over ordinary regression, especially when the non-primary instances in the bag were not correlated with the primary instances.

Dooly et al. [58] presented three multiple-instance variants of k -Nearest Neighbor (k -NN) [59], citation- k NN [35] and the diverse density algorithms [34] for real-valued prediction. The minimal Hausdorff distance from [35] was used to measure the distance between two bags. Given two sets of points $A = a_1, \dots, a_m$ and $B = b_1, \dots, b_n$, the Hausdorff

distance is defined as:

$$H(A, B) = \max\{h(A, B), h(B, A)\}, \quad (2.17)$$

where $h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$, $\|a - b\|$ is the Euclidean distance between points a and b . The Hausdorff distance, however, is very sensitive to outliers. If there is one point in B that is in some very large distance from all points in A , the Hausdorff distance will depend entirely on this one outlier point.

In their MI k -NN algorithm, the prediction made for a bag B is the average label of the k closest bags (measured in Hausdorff metric). In their MI citation- k NN algorithm, the prediction made for a bag B is the average label of the R closest bag neighbors of B (measured in Hausdorff metric) and C -nearest citers, where the “citors” include the bags where B is one of their C -nearest neighbors. It is generally recommended that $C = R + 2$. The third variant, a diverse density approach for the real-valued setting, maximizes

$$\prod_{i=1}^b Pr(r|B_i) \quad (2.18)$$

where

$$Pr(t|B_i) = (1 - |l_i - Label(B_i|t)|)/Z, \quad (2.19)$$

b is the total number of bags, t is the target point, l_i is the label for the i^{th} bag, and Z is a normalization constant. Their results showed good performance of all three variants on a benchmark Musk Molecules data set [26, 58] but the performance of both the nearest neighbor and diverse density algorithms are very sensitive to the number of relevant features, as expected based on the sensitivity of the Hausdorff distance.

Goldman and Scott [60] investigated MIR for learning real-valued geometric patterns for landmark matching in robot navigation applications. They associated a real-valued label

with each point and used the Hausdorff metric as well to help classify a bag as positive (if the points in the bag are within some Hausdorff distance from target concept points). Their algorithm differs from the supervised MIR in that the standard supervised MIR learns from a given set of training bags and bag-level training labels, while [60] applies an online agnostic model [61–63] where the learners make predictions as the bag \mathbf{x}_t is presented at iteration t . [64] also used the idea of online MIR, i.e. use the latest arrived bag with its training label to update the current predictive model. This work is also extended in [65].

Cheung and Kwok [66] proposed a regularization framework for MIR by defining a loss function that takes into consideration both training bags and training instances. The first part computes the error (loss) between training bags label and its predictions and the second part considers the loss between the bag label prediction and all the instances in the bag. Cheung and Kwok still adopted the “primary instance” assumption but simplified to assume the primary instance was the instance with the highest prediction output value. Their model provided comparable or better performance on the Musk Molecules data set [58] as citation- k NN [35] and Multiple Instance kernel-based SVM [66, 67].

Most MIR works methods discussed thus far only provided theoretical discussions or results on synthetic data set(s). Wagstaff et al. in [56, 57] started to further investigate more applications of MIR to predict crop yield from a remote sensing data set collected over California and Kansas. [56] provided a novel method for inferring the “salience” of each instance with regard to the real-valued bag label. The salience of each instance, i.e. its “relevance” with respect to all other instances in the bag to predict the bag label, is the weight associated with each instance. The salience values were defined to be non-negative and sum to one for all instances in each bag. Like Ray and Page [25], Wagstaff et al. followed the “primary-instance” assumption but their primary instance, or “exemplar” of a

bag, is the weighted average of all the points in the bag instead of one single instance from the bag. Given training bags and instances, a set of salience values are solved based on a fixed linear regression model and given the estimated salience, the regressor is updated and the algorithm reiterates until convergence. This work did not intend to provide predictions over new data but instead focused on understanding the contents (the salience) of each training instance.

Wagstaff et al. then made use of the salience learned to provide predictions for new, unlabeled bags by proposing an MI-ClusterRegress algorithm (or in some references, Cluster-MIR algorithm) that mapped instances onto (hidden) cluster labels [57]. The main assumption of MI-ClusterRegress is that the instances from a bag are drawn (with noise) from a set of underlying clusters and one of the cluster is “relevant” to the bag-level labels. After obtaining k clusters for each bag by EM (Expectation Maximization)-based Gaussian mixture models (or any other clustering method), a local regression model is constructed for each cluster. MI-RegressCluster then selects the best-fit model and use it to predict labels for test bags. A support vector regression learner [68] is used for regression prediction. Results on simulated and predicting crop yield data sets show that modeling the bag structure when the structure (cluster) is present is effective for regression prediction, especially when the cluster number k is equal to or larger than what is actually present in the bags.

More recently, Trabelsi and Frigui [69] proposed the Robust Fuzzy Clustering for MIR (RFC-MIR) algorithm that, similar to Cluster-MIR, clusters the training instances and learns multiple regression models for each cluster. RFC-MIR uses fuzzy clustering methods such as the fuzzy c-means (FCM) [70] or probabilistic c-means (PCM) [70], and uses features as well as labels in clustering. The current regression model was assumed to be linear. The probabilistic memberships were used to identify the primary instances in each

bag. Results on synthetic and predicting crop yield data sets show improved accuracy. Both Cluster-MIR and RFC-MIR combine all instances from all training bags for clustering.

MIR has since then been used in more real-world applications. EL-Manzalawy et al. [53] adapted the 1-norm SVM classifier in the MILES algorithm [41] to a support vector regression (SVR) model and applied their MHCMIR method to predict MHC-II binding affinity in molecules to help develop new vaccine. Pappas and Popescu-Belis [71] extends the works of Wagstaff et al. to deal with high dimensions and learns the instance relevance together with the target labels with application in aspect-based sentiment rating and analysis using texts from datasets such as TED talks or audiobooks. Hsu et al. [72] proposed the Augmented MIR (AMIR) algorithm to estimate object contours from images. The bounding box outside of an object in the image is regarded as “bags” and the object contour pixels are the instances. The gradient descent optimization method was used to solve for the weights of the regressor. The AMIR algorithm was evaluated using the Pascal VOC 2007 Segmentation Challenge data set and obtained good results in semantic segmentation. Notably, Wang et al. [54, 55] proposed a novel, probabilistic and generalized mixture model for MIR based on the primary-instance assumption. It is assumed that the bag label is a noisy function of the primary instance, and the conditional probability $p(y_i|Bag_i)$ for predicting label y_i for the i^{th} bag is dependent entirely on the primary instance. A binary random variable z_{ij} is defined such that $z_{ij} = 1$ if the j^{th} instance in the i^{th} bag is the primary instance and $z_{ij} = 0$ if otherwise. The mixture model for each bag i is written as:

$$p(y_i|bag_i) = \sum_{j=1}^{b_i} p(z_{ij} = 1|bag_i)p(y_i|\mathbf{x}_{ij}) \quad (2.20)$$

$$= \sum_{j=1}^{b_i} \pi_{ij}p(y_i|\mathbf{x}_{ij}), \quad (2.21)$$

where π_{ij} is the (prior) probability that the j^{th} instance in the i^{th} bag is the primary instance,

$p(y_i|\mathbf{x}_{ij})$ is the label probability given the primary instance \mathbf{x}_{ij} and b_i is the total number of instances in the i^{th} bag. Therefore, the learning problem is transformed to learning the mixture weights π_{ij} and $p(y_i|\mathbf{x}_{ij})$ from training data and an EM algorithm is used to optimize the parameters. This work discussed several methods to set the prior π_{ij} , including using deterministic function, or as a (Gaussian) function of prediction deviation, or as a parametric function (in this case a feedforward neural network). It was discussed in [55] that several previous algorithms, including Prime-MIR [25] and Pruning-MIR [54], are in fact the special case of the mixture model they proposed. Their results showed better performance on simulated data as well as for predicting aerosol optical depth (AOD) from remote sensing data and predicting crop yield applications, compared with the Cluster-MIR [57] and Prime-MIR [25] algorithms described above.

Overall, the study of Multiple Instance Regression has been in the literature for about one and a half decades and most studies base their algorithms on the primary-instance assumption proposed by Ray and Page in 2001. Linear regression was used in most cases if a regressor was used and experiments have been conducted on synthetic data sets as well as real data sets such as crop yield and AOD prediction.

2.3 Fuzzy Measure and Choquet Integral

The Choquet integral (CI) has a long history of providing an effective framework for non-linear information fusion [8, 9, 73–75]. The CI is an aggregation operator based on the fuzzy measures [11]. Depending on the values of each element in the fuzzy measure, the CI can represent a variety of relationships and combinations amongst the information sources. Therefore, a crucial aspect of using the CI for information/sensor fusion is learning the

fuzzy measures for the CI [76, 77]. This section provides description and definition of the fuzzy measure and the Choquet integral and reviews previous methods in learning the fuzzy measures, specifically within the CI.

2.3.1 Fuzzy Measure

Consider the case that there are m sources, $C = \{c_1, c_2, \dots, c_m\}$, for fusion. The set C contains $2^m - 1$ non-empty subsets. The power set of all (crisp) subsets of C is denoted 2^C .

Definition 2.3.1. A monotonic and normalized **fuzzy measure**, g , is a real valued function that maps $2^C \rightarrow [0, 1]$. It satisfies the following properties [14, 78–80]:

1. $g(\emptyset) = 0$;
2. $g(C) = 1$; *normalized*
3. $g(A) \leq g(B)$ if $A \subseteq B$ and $A, B \subseteq C$. *monotonic* 

One special and useful case of the fuzzy measure is the Sugeno λ -measure [9, 11, 81, 82]. The Sugeno λ -measure is defined as follows and will be further referred to in Section 2.3.3.

Definition 2.3.2. A **Sugeno λ -measure**, g , is a real valued function that maps $2^C \rightarrow [0, 1]$.

Let $\lambda \in (-1, +\infty)$, it satisfies the following properties [9, 11, 81, 82]:

1. $g(C) = 1$; *normalized*
2. if $A, B \subseteq C$ and $A \cap B = \emptyset$, then $g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B)$. 

Fuzzy measures, in general, model the relationship amongst the sources. Each measure element value represents the power/“worth” of a certain combination of the sources. In this dissertation, the measure elements within a fuzzy measure are denoted with a subscript

matching its corresponding subset. For example, element g_1 corresponds to subset $\{c_1\}$, element g_{12} corresponds to subset $\{c_1, c_2\}$, etc. Note that g has a total of $2^m - 1$ elements and $g_{123\dots m}$ is always equal to 1 (property 2). All other measure elements hold real values between $[0, 1]$ and satisfy monotonicity (property 3). Non-monotonic fuzzy measures have been studied in the literature [82–84] but this dissertation only deals with monotonic and normalized fuzzy measures. The term “fuzzy measures” in this dissertation always refers to monotonic and normalized fuzzy measures.

Figure 2.3 shows an illustration of the monotonicity property in between fuzzy measure elements given four sources. Take measure element g_{23} for example, g_2 and g_3 are its subsets and g_{123} and g_{234} are its supersets. Therefore, the element value of g_{23} satisfies $g_{23} \geq g_2$, $g_{23} \geq g_3$, $g_{23} \leq g_{123}$ and $g_{23} \leq g_{234}$. The red arrows show one path to “climb up the lattice” of the fuzzy measure elements. Notice that the measure element that corresponds to the full set, g_{1234} in this case, is always equal to 1 (property 2). All measure elements take values between zero and one.

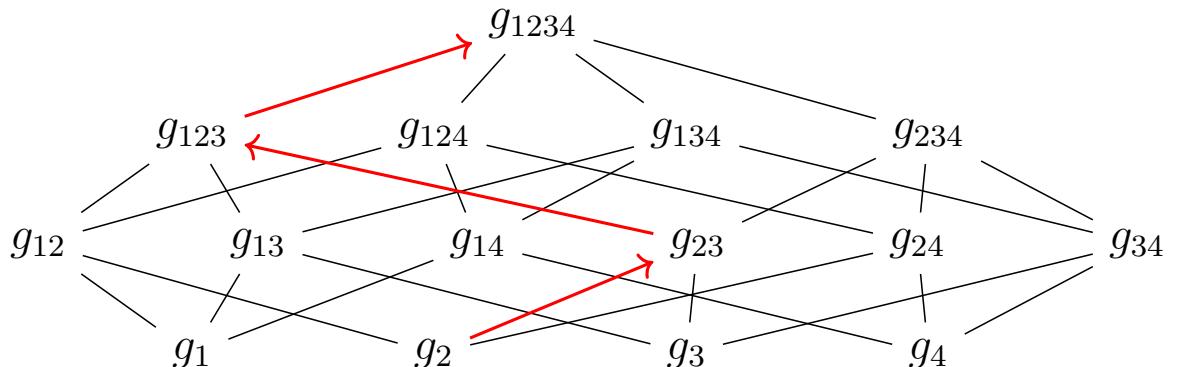


Figure 2.3: An illustration for the subset and superset relationships between fuzzy measure elements given four sources. The red arrows describe one path for “climbing up the lattice”.

Fuzzy measures have applications in multicriteria decision making [73, 85, 86], model-

ing human evaluation and decision processes [87, 88], image segmentation [89–91], image enhancement [92], gene ontology [93, 94], medical diagnostic reasoning [95], and evaluating the similarity between sets of linguistic summaries [96].

2.3.2 Choquet Integral

Fuzzy measures are used to define fuzzy integrals such as the Sugeno fuzzy integral (hereinafter referred to as “the Sugeno integral”) [78] and the Choquet fuzzy integral (hereinafter referred to as “the Choquet integral”) [14, 97] in the literature. There has been studies on the statistical properties of the Choquet and Sugeno Integrals [98–100]. The Choquet integral (CI) is a natural extension of the Lebesgue integral [11, 88] and has long been used as an effective aggregation operator [80].

Definition 2.3.3. The **Choquet integral**, denoted as $(C) \int$, of a measurable function h with respect to fuzzy measure g is defined as [11, 97]: $(C) \int h dg = \int_0^\infty g(\{h > \alpha\}) d\alpha$.

Note that if the function $h \rightarrow [0, 1]$, the ∞ sign will be replaced by 1 (integrating from 0 to 1).

In this dissertation, the discrete Choquet integral is used to fuse information provided by different sources. Suppose the sources are the outputs from the set of m classifiers or regressors, $C = \{c_1, c_2, \dots, c_m\}$, as mentioned above. Denote the classifier/regressor output of k^{th} classifier/regressor, c_k , on n^{th} data point/instance, \mathbf{x}_n , as $h(c_k; \mathbf{x}_n)$.

Definition 2.3.4. The **discrete Choquet integral** on instance \mathbf{x}_n given C is then computed as [10, 11, 80]:

$$C_g(\mathbf{x}_n) = \sum_{k=1}^m [h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)] g(A_k), \quad (2.22)$$

where C is sorted so that $h(c_1; \mathbf{x}_n) \geq h(c_2; \mathbf{x}_n) \geq \dots \geq h(c_m; \mathbf{x}_n)$. Since there are only m sources, $h(c_{m+1}; \mathbf{x}_n)$ is defined to be zero. The fuzzy measure element value corresponding to the subset $A_k = \{c_1, \dots, c_k\}$ is $g(A_k)$. 

Gader et al. [3, 101] applied the CI to multi-algorithm and multi-sensor fusion in landmine detection applications. In their work, detectors were applied to Advanced Himawari Imager (AHI) hyperspectral and synthetic aperture radar (SAR) imagery or ground penetrating radar (GPR) and electro-magnetic induction (EMI) sensors and then, the CI was used to fuse the detector results. A linguistic version of the Choquet fuzzy integral (LCFI) was proposed in [102] and was used to perform fusion on GPR, EMI and infrared (IR) sensors also for landmine detection [103]. Liu et al. [104] used the CI to perform classifier fusion on Doppler radar sensor signals for fall detection in application towards elder care. Wang et al. [105] used the CI to integrate classifier results for Landsat image classification. The CI is and has provided effective fusion performance in fusing homogeneous possibility and probability distributions [106].

The Choquet integral has wide applications in pattern recognition [74] and classification [80, 107]. The Choquet integral has been applied with Multiple Kernel Learning (MKL) for pattern recognition and feature-level and decision-level fusion [108, 109]. The CI was used in real applications such as landmine detection [9, 110], handwritten word recognition [111, 112], text classification [113], gesture recognition [114], temperature prediction [115], chromosome abnormalities detection [116] and immunoinformatics [117]. The Choquet integral has also been used as a morphological image filter [80, 118–120]. The discrete Choquet integral has also been proven to define a distance metric “with the versatility of nonlinearity and the ability to use order statistics information” [121]. In addition, the Choquet integral has been used as an effective aggregator for crowdsourced information such

as aggregating aspect rating of restaurants from customer ratings [122].

2.3.3 Learning The Fuzzy Measure

In a classifier or regressor fusion problem with training data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, the $h(c_1; \mathbf{x}_n), h(c_2; \mathbf{x}_n), \dots, h(c_m; \mathbf{x}_n)$ values for all n are known. The desired bag-level labels for sets of $C_{\mathbf{g}}(\mathbf{x}_n)$ values are known. A crucial aspect of using the CI to perform fusion is to learn all the element values of the fuzzy measure \mathbf{g} from training data of this form [10]. This dissertation adopts and adapts the categorization in [80] and [123] and describes below three main types of methods to learn the fuzzy measure, especially (but not necessarily limited) within the Choquet integral: least-square based approaches, gradient descent algorithm and evolutionary algorithms.

Least-square based approaches

As introduced in Section 2.3.1, for m input sources, there are $2^m - 1$ non-empty subsets and hence $2^m - 1$ fuzzy measure elements. Excluding $g_{123\dots m} \equiv 1$, there are $2^m - 2$ unknown fuzzy measure element values to be estimated. A quadratic programming (QP) approach to solve for the fuzzy measures based on the least-square criteria was discussed in [11, 124]. Given the discrete CI formula in equation (6.10) and assuming the desired labels for the n^{th} data point/instance \mathbf{x}_n is d_n , the goal of the the least-square criteria is to find the fuzzy measure \mathbf{g} so that the squared error is minimized between the Choquet integral outputs of all training data points given \mathbf{g} and their desired labels [11, 124]:

$$\min_{\mathbf{g}} E^2 = \sum_{n=1}^N (C_{\mathbf{g}}(\mathbf{x}_n) - d_n)^2. \quad (2.23)$$

For convenience, write the measure elements in a vector form according to the lattice

given in Figure 2.3 in a bottom-up order:

$$\mathbf{g} = [g_1, g_2, \dots, g_m, g_{12}, g_{13}, \dots, g_{123\dots m-1}, \dots, g_{23\dots m}]^T. \quad (2.24)$$

Denote Γ matrix as the difference matrix between the sources. Specifically, for each data point/instance n , the vector Γ_n is defined to depict the differences between the sources [11]:

$$\Gamma_n = [0, \dots, 0, h(c_1; \mathbf{x}_n) - h(c_2; \mathbf{x}_n), \dots, h(c_{m-1}; \mathbf{x}_n) - h(c_m; \mathbf{x}_n), 0, \dots, 0]^T. \quad (2.25)$$

The Γ_n has only $m - 1$ non-zero elements corresponding to the indices that occur in the CI definition in Eq. (6.10) [11].

Therefore, Eq. (2.23) can be rewritten as [11, 80]:

$$\min_{\mathbf{g}} E^2 = \sum_{n=1}^N (C_{\mathbf{g}}(\mathbf{x}_n) - d_n)^2 \quad (2.26)$$

$$= \sum_{n=1}^N (\Gamma_n^T \cdot \mathbf{g} + [h(c_m; \mathbf{x}_n) - d_n])^T (\Gamma_n^T \cdot \mathbf{g} + [h(c_m; \mathbf{x}_n) - d_n]) \quad (2.27)$$

$$= \sum_{n=1}^N (\mathbf{g}^T \Gamma_n \Gamma_n^T \mathbf{g} + 2 [h(c_m; \mathbf{x}_n) - d_n] \Gamma_n^T \mathbf{g} + [h(c_m; \mathbf{x}_n) - d_n]^2) \quad (2.28)$$

$$= \mathbf{g}^T \left(\sum_{n=1}^N \Gamma_n \Gamma_n^T \right) \mathbf{g} + 2 \left(\sum_{n=1}^N [h(c_m; \mathbf{x}_n) - d_n] \Gamma_n^T \mathbf{g} \right) \mathbf{g} \quad (2.29)$$

$$+ \sum_{n=1}^N [h(c_m; \mathbf{x}_n) - d_n]^2,$$

in which $(\cdot)^T$ denotes the transpose of a matrix.

We can then define,

$$D \equiv \left(\sum_{n=1}^N \boldsymbol{\Gamma}_n \boldsymbol{\Gamma}_n^T \right), \quad (2.30)$$

$$\boldsymbol{\Gamma} \equiv \left(\sum_{n=1}^N [h(c_m; \mathbf{x}_n) - d_n] \boldsymbol{\Gamma}_n \right), \quad (2.31)$$

$$d^2 \equiv \sum_{n=1}^N [h(c_m; \mathbf{x}_n) - d_n]^2, \quad (2.32)$$

Hence, Eq. (2.29) can be written as

$$\min_{\mathbf{g}} E^2 = \mathbf{g}^T D \mathbf{g} + \boldsymbol{\Gamma}^T \mathbf{g} + d^2. \quad (2.33)$$

Considering the monotonicity property of the fuzzy measure \mathbf{g} (Section 2.3.1), we require:

$$g_1 - g_{12} \leq 0,$$

$$g_1 - g_{13} \leq 0,$$

\vdots

$$g_1 - g_{123\dots m} \leq 0,$$

\vdots

$$g_{123\dots m-1} - g_{123\dots m} \leq 0,$$

\vdots

$$g_{23\dots m} - g_{123\dots m} \leq 0,$$

i.e., a total of $m(2^{m-1} - 1)$ inequality constraints on elements of \mathbf{g} . Notice that $g_{123\dots m} \equiv 1$ according to the normalization property of the fuzzy measure \mathbf{g} (Section 2.3.1).

To write those inequality constraints in $\mathbf{A} \cdot \mathbf{g} + \mathbf{b} \leq \mathbf{0}$ format,

$$\begin{bmatrix}
1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
1 & 0 & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1
\end{bmatrix}_{(m(2^{m-1}-1) \times 2^m-2)} \cdot \begin{bmatrix} g_1 \\ \vdots \\ g_{12} \\ g_{13} \\ \vdots \\ g_{123\dots m-2} \\ g_{123\dots m-1} \\ g_{134\dots m} \\ \vdots \\ g_{234\dots m} \end{bmatrix}_{((2^m-2) \times 1)} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}_{(m(2^{m-1}-1) \times 1)} \leq \mathbf{0} \quad (2.35)$$

Note that the “-1” elements of the \mathbf{b} vector comes from the $g_{123\dots m} \equiv 1$ property.

Here is an example for three input sources ($m = 3$). We know there are a total of 9 constraints, i.e.

$$\begin{aligned}
g_1 &\leq g_{12}, & g_1 &\leq g_{13}, \\
g_2 &\leq g_{12}, & g_2 &\leq g_{23}, \\
g_3 &\leq g_{13}, & g_3 &\leq g_{23}, \\
g_{12} &\leq g_{123} = 1, & g_{13} &\leq g_{123} = 1, & g_{23} &\leq g_{123} = 1.
\end{aligned} \tag{2.36}$$

The constraints can be rewritten in matrix form as:

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(9 \times 6)} \cdot \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_{12} \\ g_{13} \\ g_{23} \\ \vdots \\ (6 \times 1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \end{bmatrix}_{(9 \times 1)} \leq \mathbf{0} \quad (2.37)$$

Therefore, the problem of finding the fuzzy measure \mathbf{g} according to the least square criteria formulated in Equation (2.33) turns into a quadratic programming (QP) problem [11, 124]:

$$\min_{\mathbf{g}} E^2 = \mathbf{g}^T D \mathbf{g} + \mathbf{\Gamma}^T \mathbf{g} + d^2, \quad (2.38)$$

subject to $\mathbf{A} \cdot \mathbf{g} + \mathbf{b} \leq \mathbf{0}$ and $\mathbf{0} \leq \mathbf{g} \leq \mathbf{1}$.

The quadratic programming approach can then be solved by a QP solver such as the MATLAB built-in *quadprog()* function. This method is called “CI-QP” method and was used as a baseline comparison method later in the dissertation.

There are other approaches including a maximum split approach [125], a minimum variance method [126, 127] and a less constrained approach [128] as discussed in [123]. The maximum split approach models the pairs and interactions of the fuzzy measure elements subject to the monotonicity and normalization constraints and solves the maximization problem using linear programming. The maximum split approach is a simple model [123] but introduces many variables and threshold parameters that must be chosen care-

fully [125]. The minimum variance method is based on a maximum entropy principle for fuzzy measures [127]. The principle argues that the maximum entropy capacities (fuzzy measures) “is the one that will exploit the most on average its arguments”, i.e. the optimal solution is the one that maximizes the entropy and hence minimizes the variance of the fuzzy measures (the link between the entropy and the variance can be seen in [127]). The less constrained approach [128] also writes the objective function based on the least squares criteria and optimizes the objective function subject to the properties of the fuzzy measure, using a convex quadratic program. However, it relaxes some of the constraints by converting the constraints on the fuzzy measures to constraints on the (unknown) overall evaluation values. This methods can be regarded as a generalization of the standard least-squares QP method [123].

Gradient descent algorithm

A gradient-descent algorithm has been used to learn a fuzzy measure (a Sugeno λ -measure in this case) for the Choquet integral in [9]. The Sugeno λ -measure is defined in Section 2.3.1. Denote the measure element corresponding to a singleton set $\{c_j\}$, also known as “densities”, as g_j , $j = 1, \dots, m$. Take the derivative of the Choquet integral in Equation (6.10), each partial derivative of $C_{\mathbf{g}}(\mathbf{x}_n)$ with respect to the measure element g_j is given by:

$$\frac{\partial C_{\mathbf{g}}(\mathbf{x}_n)}{\partial g_j} = \sum_{k=1}^m \frac{\partial g(A_k)}{\partial g_j} [h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)]. \quad (2.39)$$

Recall the notation that fuzzy measure element $g(A_k)$ corresponds to the subset $A_k = \{c_1, \dots, c_k\}$, as discussed in Section 2.3.2. According to the definition of the Sugeno λ -

measure in Section 2.3.1, $g(A_k)$ can be written as

$$\begin{aligned} g(A_k) &= g(\{c_k\} \cup A_{k-1}) \\ &= g_k + g(A_{k-1}) + \lambda g_k g(A_{k-1}). \end{aligned} \tag{2.40}$$

Then, the $\frac{\partial g(A_k)}{\partial g_j}$ term in Equation (2.39) can be written as

$$\begin{aligned} \frac{\partial g(A_k)}{\partial g_j} &= \frac{\partial g_k}{\partial g_j} + \frac{\partial g(A_{k-1})}{\partial g_j} \\ &\quad + \frac{\partial \lambda}{\partial g_j} g_k g(A_{k-1}) + \dots \\ &\quad + \lambda \frac{\partial g_k}{\partial g_j} g(A_{k-1}) + \lambda g_k \frac{\partial g(A_{k-1})}{\partial g_j}. \end{aligned} \tag{2.41}$$

It is known in [11] that the value λ for any Sugeno λ -measure satisfies

$$(1 + \lambda) = \prod_{k=1}^m (1 + \lambda g_k), \tag{2.42}$$

given the normalization property of the Sugeno λ -measure. Therefore, the $\frac{\partial \lambda}{\partial g_j}$ term in Equation (2.41), as proven in [9], can be rewritten as

$$\frac{\partial \lambda}{\partial g_j} = \frac{\lambda^2 + \lambda}{(1 + g_j \lambda) \left[1 - (\lambda + 1) \sum_{k=1}^m \left(\frac{g_k}{1+g_k\lambda} \right) \right]}, \lambda \neq 0. \tag{2.43}$$

Then, a gradient descent algorithm can be used for any objective function (such as the aforementioned least-squares criterion) to solve for the fuzzy measure to be used with the Choquet integral [9, 104].

Keller and Osborn proposed a neuron model and a “reward-punishment” training algorithm to train fuzzy measures for multi-class decision making [129]. A neuron is used to represent each data class and the fuzzy measure element values were initialized. Then, the Choquet integral output is calculated and compared with the true class label. If the CI output is incorrect, the density values are decreased (“punished”) and if correct, increased (“reward”) [129]. The process is continued until all data are correctly classified. The neural

networks approach to determine a (Sugeno) fuzzy measure can also be seen in [130, 131].

Evolutionary Algorithms

The Genetic Algorithm (GA) is a special instance and a popular algorithm in Evolutionary algorithms [132]. The GA has been used in the literature to determine fuzzy measure values [10, 76, 133–141]. The GA algorithm considers the fuzzy measure as a chromosome and generates a population of potential measures. A fitness function is pre-defined to model and evaluate the measure. An example of the fitness function is a function that calculates misclassification rate or the least-squared error between the CI output and the actual class label. Measures are selected from the measure population based on their fitness values. Measures from the last iteration are called the “parents.” The measure element values are then updated by mutation based on their fitness values and the “child” measure population is generated. The process continues until a stopping criterion is met (for example, maximum number of iterations or the minimum misclassification rate). The measure that gives the best fitness value is returned as the optimal fuzzy measure solution.

Additional methods, such as the Particle Swarm Optimization (PSO) [142, 143], have also been used to solve nonlinear multi-regression based on generalized Choquet integrals [144]. Results in [144] suggest that the PSO speeds up the convergence and reduces the computation time as compared with the GA algorithm, specifically for determining the unknown regression parameters for nonlinear multiregression model.

2.4 Sensor Fusion

Sensor fusion is the process of combining data or information from multiple (sensory) inputs (“sources”). Sensor data, features extracted from the data, and classifier or prediction outputs estimated from the data can all be used as sources during fusion [145, 146]. Sensor fusion can be performed at three levels: data level, feature level, and decision level [4, 17]. Data-level fusion takes in sensor data directly and perform transformation on the data itself to project, map or co-register data from multiple sensors. Feature-level fusion performs fusion of features, such as characteristics extracted from the data like shapes or objects [4]. Decision-level fusion takes multiple classifier or real-valued prediction outputs estimated from the data as sources of information. It then derives one fused output with the aim of providing a finer, more accurate and/or descriptive classification or regression result [10, 146, 147]. Sensor fusion has wide applications including remote sensing and hyperspectral image classification [148–150], object detection [151], landmine detection [103, 147, 152–154], handwriting recognition [155], and medical and fault diagnosis [156–158].

There are a number of challenges in sensor fusion, such as data alignment, varying modality/data types, and fusion of imperfect data [159]. In the case where the fusion sources are images or maps, it is necessary, if not “an absolute prerequisite” [160], to make sure the pixels across all images correspond to the same region and objects [18, 160]. The process of (geometrically) align two or more of the images to be fused is called **image co-registration** [160]. Notice that image co-registration is a pre-processing step for fusion that only deals with images or maps on a grid of pixels as input sources. This section will review image co-registration methods as a pre-processing step for image-type-only sensor fusion.

On the other hand, sensors often produce data of mixed data types other than images

or maps (for example, point clouds). Sensor imagery can also be of multi- spatial, spectral and temporal resolution and with mixed uncertainties [108]. This section reviews fusion methods in the literature that deals with multi-resolution sensor fusion, in particular fusion of mixed spatial, spectral and temporal resolution. Furthermore, this section reviews fusion methods that can handle data of mixed types, such as fusing multispectral (MS) or hyperspectral (HS) imagery with LiDAR point cloud data.

2.4.1 Co-registration

Image co-registration is a fundamental step in image processing that has applications in computer vision and pattern recognition, medical image analysis and remote sensing data processing and fusion [161–163]. Image co-registration deals with a variety of problems, such as (1) registration of images over the same scene from different sensors (multimodal registration), (2) registration of images over the same scene from different viewpoints (multiviewpoint registration), (3) registration of images over the same scene at different times or under different conditions (temporal registration), and (4) find pattern or object in the image that matches a desired pattern (template registration) [161]. For multi-sensor fusion purposes, the co-registration step usually refers to the first three scenarios.

Most multi-sensor image co-registration methods are based on geocoding, parametric methods such as similarity measures, or non-parametric methods such as optical flow [18].

Geocoding

Geocoding co-registration refers to the process of transforming the pixels in the image to a geographic location on the surface of the Earth, represented in spatial coordinates such as a latitude and longitude [4, 18, 164]. Approaches of geocoding includes rigorous range

doppler approach, interpolation and nearest-neighbor resampling, among others [18, 164, 165].

The geocoding co-registration process plays an essential role for fusion, but the accuracy of geocoding highly depends on the availability and accuracy of auxilary geographic data such as a dital elevation model (DEM) [18, 166]. However, many Global Positioning System (GPS) device are often only accurate to the level of several meters [24, 167]. Ground control points can be selected to enhance the accuracy of geocoding [166] but the process still requires geometrically corrected reference systems and it requires extra expense for picking control points.

Similarity measures

Similarity measures methods, described here, are feature-based co-registration methods [168]. In the literature, features such as edges, edge orientations, points, regions and line features have been used to perform image co-registration [169, 170]. The scale-invariant feature transform (SIFT), for example, has been used as a popular feature descriptor [171]. Image intensity was also used in classical area-based methods such as cross-correlation [172] or mean square difference of image intensity values [173]. Feature-based methods are useful if the images contain enough detectable and distinctive details and/or objects in the scene [168].

Once the features were selected, a similarity measure or function is used to evaluate and optimize the similarity between images. The choice of similarity measures plays a significant role in image co-registration [18]. Mutual Information (MI) [174] is widely used as a similarity measure to co-register images, especially in applications such as medical imaging and remote sensing [175–182]. The MI between two random variables A and B is

defined as [174, 183]:

$$I(A, B) = H(A) + H(B) - H(A, B), \quad (2.44)$$

where $H(A)$ and $H(B)$ are the Shannon entropies [184] of A and B , respectively and $H(A, B)$ is the joint entropy of A and B .

Suppose A and B are two sensor images to be co-registered and assume the intensity values in image A ranges from $[0, M - 1]$ and the intensity values in image B ranges from $[0, N - 1]$. Define the joint histogram matrix, h , as:

$$h = \begin{bmatrix} h(0, 0) & h(0, 1) & \dots & h(0, N - 1) \\ h(1, 0) & h(1, 1) & \dots & h(1, N - 1) \\ \vdots & \vdots & \dots & \vdots \\ h(M - 1, 0) & h(M - 1, 1) & \dots & h(M - 1, N - 1) \end{bmatrix}, \quad (2.45)$$

where $h(a, b)$ is the number of pairs having intensity value a in image A and intensity value b in image B . The joint histogram matrix h , thus, describes the relationship between the image pairs A and B . The joint probability mass function $p_{A,B}(a, b)$ is then defined as:

$$p_{A,B}(a, b) = \frac{h(a, b)}{\sum_{a,b} h(a, b)}. \quad (2.46)$$

The marginal probability mass functions $p_A(a)$ and $p_B(b)$ are computed as:

$$p_A(a) = \sum_b p_{A,B}(a, b), \quad (2.47)$$

and

$$p_B(b) = \sum_a p_{A,B}(a, b). \quad (2.48)$$

The entropies $H(A)$ and $H(B)$ and joint entropy $H(A, B)$ can be computed as:

$$H(A) = \sum_a -p_A(a)\log p_A(a), \quad (2.49)$$

$$H(B) = \sum_b -p_B(b)\log p_B(b), \quad (2.50)$$

$$H(A, B) = \sum_{a,b} -p_{A,B}(a, b)\log p_{A,B}(a, b). \quad (2.51)$$

Here the intensity values are used as suggested in [174, 181] but other features can be used as well. Other interpolation algorithms, such as nearest neighbor [185], can also be used to estimate the joint histogram matrix.

Similarity measures methods such as the MI-based co-registration face challenges such as high computation time due to large data volume and wide differences of sensor geometry and radiometry due to increased spatial resolution of sensor data [181]. Traditional features tend not to work well with multiangle remote sensing images as the resolution changes in images with large view angles may affect the accuracy of key points selection [186]. Other properties, such as the low-rank constraint [186], has been explored for image co-registration. The RANdomSAmple Consensus (RANSAC) algorithm [187–191] was also widely used in feature-based image registration.

Transformation, Interpolation and Resampling

Transformation, interpolation and resampling methods are studied in the literature for image co-registration [169, 192]. To co-register two images, one image is fixed to be the “master image” and the other image is projected and resampled to the same reference system as the master image [18, 193]. A variety of transformations can be used to co-register images, such as similarity transform, affine transform, perspective projection, and elastic

transform [169]. Popular interpolation methods include the nearest neighbor [194, 195], cubic convolution [196], bilinear/tri-linear interpolation [197–199], quadratic interpolation [200], polynomial interpolation [192], and spline interpolation [201, 202]. A survey of interpolation methods in medical imaging applications can be found in [203]. Interpolation methods can reduce the computational expense of conventional co-registration methods [192]. The efficiency and performance, however, will depend on the selection of the optimal interpolation kernel [192].

Additionally, the spectral diversity technique is used to perform co-registration, more specifically for SAR images, based on the spectral properties of the complex SAR signal [204, 205]. The spectral diversity co-registration does not need any interpolation or cross-correlation and can be “at least as accurate as” conventional co-registration methods [204].

Conflation

The term “conflation” has been used, in some cases, interchangably in the literature as image (co-)registration [206–211]. Conflation is in particular associated with applications relating to map compilation and geospatial data, such as overlaying a vector road map to a geospatial imagery [212–214]. Conflation integrates and combines (in particular) geographic information from multiple sources to “retain accurate data, minimize redundancy, and reconcile data conflicts.” [211, 215] Just like image co-registration, features and similarity measures can be extracted to match images or to overlay different geospatial information sources [211].

2.4.2 Multi-resolution Fusion

Existing optical sensors produce data at varying spatial, temporal and spectral resolutions [17]. In order to obtain an image that has higher spatial, temporal or spectral resolution and integrating complementary information provided by multiple sensors, multi-resolution fusion methods are needed and studied in the literature. This sections describes previous fusion methods that deal with multi-spatio-temporal-spectral resolutions, concentrating on remote sensing applications.

Spectral-spatial fusion is an important branch in multi-resolution fusion [17]. Spectral-spatial fusion combines images of varying spectral and spatial resolutions to obtain an image with high spectral/spatial resolution. **Pan-sharpening** is one example of spectral-spatial fusion that has been heavily studied in the literature [216–218]. Pan-sharpening refers specifically to fusing a high-resoluion panchromatic (PAN) image with a lower-resolution multispectral (MS) image [219]. It is necessary to consider the tradeoff between computational complexity and detection/classification accuracy when considering pan-sharpening algorithms [219]. Model-based optimization pan-sharpening approaches [220] model the relationships between PAN and MS images and optimizes an objective function (e.g. minimizes error between observed and desired images) to solve for the deisred fused output. Model-based optimization generally has higher precision than component substitution approaches [221–223] or multiresolution analysis-based approaches [224] but can be complex and time-consuming when the image size is large [17]. A sparse coding method proposed in [225] takes into account the signal correlation between the spectral bands and turns pan-sharpening into a dictionary learning problem. It relies on “dictionary atoms”, i.e. columns of the dictionary, trained from observed PAN images, learns the (sparse) coefficients for each dictionary atom and reconstruct the desired images

as a linear combination of those dictionary atoms. Pan-sharpening methods also include methods based on compressed sensing [226–228], wavelet-based algorithms [229–231], deep neural networks [232] and particle swarm optimization [233, 234].

In addition to PAN/MS fusion, PAN/hyperspectral (HS) fusion [235, 236] and MS/HS fusion [237, 238] are also studied within spectral-spatial fusion. In general, panchromatic images offer higher spatial resolution but MS or HS images provide spectral information over multiple (tens to hundreds of) bands and higher spectral resolution. PAN/HS and MS/HS are more challenging due to the increased number of spectral bands [17].

Sensors such as satellite systems often produce multi-view, multi-angle and multi-frame images. Temporal information can also be used in fusion. Multi-view remote sensing and aerial imagery data integration was studied in [239–241]. [17] provides a summary of multiview spatial fusion and spatio-temporal fusion.

There have also been studies on dealing with spatial, temporal and spectral resolutions with a unifying framework. Shen et al. [17] and Meng et al. [242] proposed an integrated framework that uses a maximum a posteriori probability (MAP) estimate for spatio-temporal-spectral fusion. It is based on a model-based optimization scheme and models the relationships between the desired (fused) image and the multi-source multi-resolution observed images using matrices. The objective function is written as minimizing the difference between the observed images and the desired image (the fused image) after applying the matrix transformation. A conjugate gradient algorithm [243] is used to solve for the desired fused image. Chen et al. [244] proposed methods for six individual one-to-one fusion modes (spatial-spectral, spatial-temporal, spatial-angular, etc.) and proposed a “general mode” of spatial-temporal-spectral fusion by going through each one-to-one fusion mode step by step.

Applications of multi-resolution remote sensing image fusion include and not limited to extraction of urban road networks [239], building detection [245], precision agriculture [246], and anomaly detection in archaeology [247].

2.4.3 Fusion of Mixed Data Types

Sensor sources can not only produce multi-resolution image data but also data of mixed geospatial types and mixed uncertainties [108]. One important and widely-studied example is the fusion of remote sensing imagery data (such as hyperspectral or multispectral imagery) and LiDAR (Light Detection and Ranging) data. Take the fusion of hyperspectral imagery (HSI) and LiDAR data [248–250] for example. Hyperspectral imagery provide spectral information of materials in the scene over a wide range of spectral bands while LiDAR data provides height information. HSI can help distinguish materials and objects but only if the materials and objects have distinct spectral signature, for example vegetation versus asphalt. HSI alone cannot tell an asphalt road from building roof made of asphalt. On the other hand, LiDAR data distinguishes materials based on height information, so LiDAR data can easily tell asphalt road apart from a rooftop, but will have a hard time distinguishing a tree and a building, if they are of similar height. In real applications, LiDAR data is highly effective in detecting buildings [251–254], modeling and identifying trees or plants [255–259], and modeling water bathymetry [260–262]. Therefore, the fusion of HSI and LiDAR can integrate complementary information and effectively extract features and distinguish materials in the scene [249].

Hyperspectral cameras produce images on a pixel grid. Each pixel covers a certain area in the scene, based on the image resolution and ground sample distance. The values of each pixel is a vector carrying reflectance values across wavelengths/spectral bands.

LiDAR sensors, on the other hand, obtain data by densely sampling the surface of the earth. The raw LiDAR data are usually in the form of point clouds instead of maps or images on a pixel-by-pixel grid. LiDAR point cloud is a vector-based structure. Each point in the point cloud is associated with an X-Y coordinate that matches a location on earth (UTM coordinates or Latitude/Longitude, for example). Z-coordinate value at each data point provides height information. There may be multiple LiDAR points corresponding to each pixel area on earth. The size of both data would not even match. It is, thus, difficult, to directly fuse the HSI and LiDAR sources.

One solution is to pre-process the LiDAR data by **rasterization**. LiDAR data can be rasterized by projecting data points into a pixel coordinate plane similar to that of an HSI image [21]. Or, in some cases, a digital surface model (DSM) or digital elevation model (DEM) derived from the LiDAR is available instead of the raw data [263]. A DSM represents the earth's surface and all objects (such as buildings and trees) on the surface, while a digital terrain model (DTM) represents the bare ground surface without any objects. A DEM is often used as a generic term for DSMs and DTMs. When a DSM or a DEM is available, the LiDAR data can be converted to or represented as an image based on pixel grid that can be further processed and fused with HSI or any other sensor images [21]. Features can then be extracted from both LiDAR and HSI images based on methods such as morphological attribute profiles [263–267]. Those features can be stacked together as a large feature matrix and classifiers such as the support vector machine (SVM) [268] or logistic regression [269] can be applied to obtain the final fusion results [248, 263]. Such pre-processing allows the usage of classical image processing and fusion methods based on image inputs. However, the raw dense LiDAR point cloud data can offer higher geographic accuracy [22] because the raw data does not depend on grid size. Besides, most

image registration methods rely heavily on both the accuracy of input images and registration parameters [18, 23]. Rasterization and co-registration can also bring in another layer of uncertainty regarding geometric misalignment and mismatch [15, 21].

Sparse representation has also been used for HSI and LiDAR fusion [270]. The fusion is modeled as a dictionary learning problem. For each data sample $\mathbf{x} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m]$ for m sources, given dictionary $\mathbf{A}^i = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_C]$ as the dictionary representation of C classes for the i^{th} source, the sparse coefficient can be estimated by [270]

$$\hat{\mathbf{S}} = \arg \min_{\mathbf{S}} \frac{1}{2} \sum_{i=1}^m \|\mathbf{x}^i - \mathbf{A}^i \mathbf{s}^i\|_2^2 + \lambda \|\mathbf{S}\|_{1,q}, \quad (2.52)$$

where $\|\mathbf{S}\|_{1,q}$ is the l_1/l_q norm of the coefficient matrix \mathbf{S} and λ is a regularization parameter. Notice that this method also requires an one-to-one correspondence for all the data samples from m sources so that \mathbf{x} may be stacked into one data matrix .

2.5 Summary and Discussion of Literature Review

The standard Choquet integral fusion method and applications mentioned in Section 2.3 only work if the following two assumptions are met.

First, the data to be fused are homogeneous. It assumes that the input sources for CI fusion must be of the same type, of the same resolution, on the same grid, or be possible to match and link together if the data types are arbitrary and/or heterogeneous [15]. This assumption is implied from the standard discrete CI formulation in Equation (6.10). In order to compute the CI output, the $[h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)]$ term on the right-hand side of the equation must be valid and calculable. Therefore, the function output of source k and source $k + 1$ on data point \mathbf{x}_n must be comparable and of the same size for all sources

$k = 1, \dots, m$. However, as discussed in Sections 2.4.2 and 2.4.3, this assumption does not generally hold true in real applications for sensor fusion.

Sensor data may be available in mixed geospatial types. It is not necessarily feasible to always convert all data to images on a pixel grid. Suppose a hyperspectral imaging (HSI) camera scans the scene and provides an image with 1-meter ground sample distance (GSD) [20]. That means each pixel in the hyperspectral image covers $1 \times 1m^2$ area. LiDAR technology, on the other hand, produces point clouds by densely sample the surface of the earth in the scene. There can be several LiDAR data points inside an $1 \times 1m^2$ area. Traditional fusion methods fail to consider the scenario where the data point/pixel number does not match among sensor sources, if the raw heterogeneous data types are used for fusion. Techniques may not always allow transformation the raw data into images of the same resolution or map to the same pixel grid.

Even if the sensor data are all imagery, existing optical sensors operate on varying spatial and spectral and temporal resolutions [17]. The first step in multisensor image data fusion is usually data alignment, i.e., the observations from the individual sensors must be processed to a form that is suitable for subsequent processing [271]. Techniques such as rasterization and image co-registration, as discussed in Section 2.4.1, align images from multiple sensors for further processing. However, co-registration only deals with image inputs and may introduce mis-alignment error. Rasterization projects vector data or point cloud data to a pixel-grid and that process may lose some of the original information as well.

Second, to learn a fuzzy measure to be used with the CI, standard methods require training labels available for each data point. This assumption states that the desired classification or prediction values must be known for training data. If the least-squares criterion

is used with the CI fusion, for example, the d_n term in Equation (2.23) must be known for each and every training data point \mathbf{x}_n [16]. This is difficult to achieve in real sensor fusion applications as well. Data-point specific labels are often unavailable or difficult and expensive to obtain [10]. For target detection in real remote sensing applications, for example, a Global Positioning System (GPS) device may only be accurate to the level of several meters [24]. Depending on the GSD of the imagery, the target ground truth locations in the scene measured by a GPS can only be accurate to the level of several pixels. It is, thus, difficult to pinpoint accurate pixel-level target locations and provide accurate training labels.

The multisensor fusion algorithms presented in this research are capable of fusing data of mixed types and multi-resolution, while accounting for uncertainty in training labels. The new algorithms are able to be applied to classifier fusion with binary classification labels as well as regression problems with real-valued class labels.

Chapter 3

Multiple Instance Choquet Integral

This chapter provides a detailed description on the proposed Multiple Instance Choquet Integral (MICI) framework for multi-sensor classifier fusion uncertainties in training labels. Three variations of the objective function based on three models, i.e. noisy-or, min-max and generalized-mean models, were derived for the MICI classifier fusion framework. A monotonic normalized fuzzy measure is learned to be used with the Choquet integral to perform two-class classifier fusion given bag-level training labels. An optimization scheme based on an evolutionary algorithm is used to optimize the models proposed.

The Multiple instance Choquet integral classification model learns a fuzzy measure to be used within a Choquet integral for two-class classifier fusion. The learning approach relies on the MIL framework [26, 34] given bag-level training labels. The computed Choquet integral with the learned measure aims at matching as much as possible to the desired labels. In the two-class classifier fusion problems in this dissertation, the positive class (target) is marked label “+1” and the negative class (non-target) is marked label “0”.

3.1 Noisy-or Model

In standard MIL, a bag is labeled negative if all the instances in the bag are negative and a bag is labeled positive if there is at least one positive instance in the bag. The noisy-or model can be used to express these assumptions [34]:

$$\begin{aligned} P(\mathbf{X}|\boldsymbol{\theta}) &= \prod_{n=1}^N p(\mathbf{x}_n|\boldsymbol{\theta}) \\ &= \left(\prod_{a=1}^{B^-} \prod_{i=1}^{N_a^-} 1 - p(\mathbf{x}_{ai}^-|\boldsymbol{\theta}) \right) \left(\prod_{b=1}^{B^+} \left(1 - \prod_{j=1}^{N_b^+} 1 - p(\mathbf{x}_{bj}^+|\boldsymbol{\theta}) \right) \right) \end{aligned} \quad (3.1)$$

where N is the total number of data points, \mathbf{x}_n is the n^{th} data point/instance, B^+ is the total number of positive bags, B^- is the total number of negative bags, N_b^+ is the total number of instances in positive bag b , and N_a^- is the total number of instances in negative bag a . Each \mathbf{x}_n is either positive or negative, this is indicated by the following notation: \mathbf{x}_{ai}^- is the i^{th} instance in the a^{th} negative bag and \mathbf{x}_{bj}^+ is the j^{th} instance in the b^{th} positive bag. $\boldsymbol{\theta}$ represents the model parameters.

For all $a = 1, \dots, B^-$, we want:

$$C_g(\mathbf{x}_{ai}^-) = 0, \forall \mathbf{x}_{ai}^- \in \mathcal{B}_a^- . \quad (3.2)$$

For all $b = 1, \dots, B^+$, we want:

$$C_g(\mathbf{x}_{bj}^+) = 1, \exists \mathbf{x}_{bj}^+ \in \mathcal{B}_b^+ . \quad (3.3)$$

where C_g is the Choquet integral given measure g , \mathcal{B}_a^- is the a^{th} negative bag, and \mathcal{B}_b^+ is the b^{th} positive bag.

In this algorithm, we define

$$\begin{aligned} p(\mathbf{x}_n|\boldsymbol{\theta}) &= \mathcal{N}(C_{\mathbf{g}}(\mathbf{x}_n)|\mu, \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(C_{\mathbf{g}}(\mathbf{x}_n) - \mu)^2\right\}. \end{aligned} \quad (3.4)$$

We set $\mu = 1$ to encourage the Choquet integral values of positive instances to be 1 and the Choquet integral values of negative instances to be far from 1. Here, the model parameter vector $\boldsymbol{\theta}$ consists of the variance of the Gaussian σ^2 and the fuzzy measure \mathbf{g} values used to compute the Choquet integral.

Then, after taking the logarithm, the objective function in (3.1) becomes:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{a=1}^{B^-} \sum_{i=1}^{N_b^-} \ln \left(1 - \mathcal{N}(C_{\mathbf{g}}(\mathbf{x}_{ai}^-)|1, \sigma^2) \right) + \sum_{b=1}^{B^+} \ln \left(1 - \prod_{j=1}^{N_b^+} 1 - \mathcal{N}(C_{\mathbf{g}}(\mathbf{x}_{bj}^+)|1, \sigma^2) \right). \quad (3.5)$$

By maximizing the objective function in (3.5), we encourage the Choquet integral of all the points in the negative bag to zero (first term) and encourage the Choquet integral of at least one of the points in the positive bag to one (second term). The variance σ^2 is a user-defined parameter. The variance parameter controls how sharply the Choquet integral values are pushed to 0 and 1, and thus controls the weighting of the two terms separately. A larger variance parameter allows for more noise in the data by allowing points in negative bags to have higher CI values and positive points to have lower CI values.

3.2 Min-Max Model

The min-max model substitutes the noisy-or with simple “min” and “max” operators following the same MIL assumption. The MIL framework assumes that for negative bags, all

the instances in the bag are negative (label “0”):

$$J^- = \sum_{a=1}^{B^-} \max_{\forall \mathbf{x}_{ai}^- \in \mathcal{B}_a^-} (C_g(\mathbf{x}_{ai}^-) - 0)^2; \quad (3.6)$$

For positive bags, at least one instances in the bag should be positive (label “1”):

$$J^+ = \sum_{b=1}^{B^+} \min_{\forall \mathbf{x}_{bj}^+ \in \mathcal{B}_b^+} (C_g(\mathbf{x}_{bj}^+) - 1)^2, \quad (3.7)$$

where B^+ is the total number of positive bags, B^- is the total number of negative bags, \mathbf{x}_{ai}^- is the i^{th} instance in the a^{th} negative bag and \mathbf{x}_{bj}^+ is the j^{th} instance in the b^{th} positive bag. C_g is the Choquet integral output given measure g computed using (6.10), \mathcal{B}_a^- is the a^{th} negative bag, and \mathcal{B}_b^+ is the b^{th} positive bag.

Thus, the objective function for the MICI Min-Max model classifier fusion approach is written as follows:

$$J = J^- + J^+. \quad (3.8)$$

By minimizing the objective function in (3.8), we encourage the Choquet integral of all the instances in the the negative bag to zero (“ J^- ” term) and encourage the Choquet integral of at least one of the points in the positive bag to one (“ J^+ ” term), which fits the MIL assumption.

3.3 Generalized Mean Model

Instead of hard min and max operators as proposed in Section 3.2, a generalized mean model is used to replace the hard min and max operators.

Definition 3.3.1. If p is a non-zero real number, and x_1, \dots, x_n are positive real numbers,

then the **generalized mean**, or power mean, with the exponent p of x_1, \dots, x_n is [272]:

$$M_p(x_1, x_2, \dots, x_n) = \left(\frac{1}{n} \sum_{i=1}^n x_i^p \right)^{\frac{1}{p}}. \quad (3.9)$$

The generalized mean has the following two properties:

$$M_{-\infty}(x_1, x_2, \dots, x_n) = \lim_{p \rightarrow -\infty} M_p(x_1, x_2, \dots, x_n) = \min(x_1, x_2, \dots, x_n). \quad (3.10)$$

$$M_\infty(x_1, x_2, \dots, x_n) = \lim_{p \rightarrow \infty} M_p(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n). \quad (3.11)$$

Therefore, by adjusting the p value, the term can act as varying aggregating operators.

For negative bags, all the instances in the bag are negative (label “0”).

$$J^- = \sum_{a=1}^{B^-} \left[\frac{1}{N_a^-} \sum_{i=1}^{N_a^-} (C_g(\mathbf{x}_{ai}^-) - 0)^{2p_1} \right]^{\frac{1}{p_1}}, \quad (3.12)$$

Similarly, for positive bags, at least one instances in the bag should be positive (label “1”).

$$J^+ = \sum_{b=1}^{B^+} \left[\frac{1}{N_b^+} \sum_{j=1}^{N_b^+} (C_g(\mathbf{x}_{bj}^+) - 1)^{2p_2} \right]^{\frac{1}{p_2}}, \quad (3.13)$$

where B^+ is the total number of positive bags, B^- is the total number of negative bags, N_b^+ is the total number of instances in positive bag b , and N_a^- is the total number of instances in negative bag a , \mathbf{x}_{ai}^- is the i^{th} instance in the a^{th} negative bag and \mathbf{x}_{bj}^+ is the j^{th} instance in the b^{th} positive bag. C_g is the Choquet integral given measure g , \mathcal{B}_a^- is the a^{th} negative bag, and \mathcal{B}_b^+ is the b^{th} positive bag. p_1 and p_2 are the two exponent parameters of the generalized mean and $p_1 \rightarrow \infty$ and $p_2 \rightarrow -\infty$, according to properties (3.10) and (3.11),

Thus, we can define the objective function for the proposed MICI softmax classifier

fusion as follows:

$$J = J^- + J^+. \quad (3.14)$$

By minimizing the objective function in (3.14), we encourage the Choquet integral of all the points in the negative bag to zero (“ J^- ” term) and encourage the Choquet integral of at least one of the points in the positive bag to one (“ J^+ ” term).

3.4 Optimization

An optimization scheme based on the evolutionary algorithm is used to optimize the models proposed. Pseudocode for the MICI algorithm for both training and testing stages can be seen in Algorithm 1. The notation used in Algorithm 1 please refer to the LIST OF SYMBOLS AND NOTATIONS provided in Page x in the beginning of this dissertation. The right arrows in the pseudocode mark the corresponding subsections where each step will be explained in detail.

3.4.1 Measure Initialization

In the algorithm, a population (size P) of the Choquet integral measures is generated and each measure in the population is initialized randomly to a set of values between $[0, 1]$ that satisfies monotonicity.

Two types of initialization approaches, “top-down” and “bottom-up” approaches, were implemented. In the “top-down” initialization, the values of the measure elements were sampled from the top of the lattice towards the bottom. Again using the example of the measure lattice of four sources as shown in Figure 2.3, the $(m - 1)$ -tuple measure elements

Algorithm 1 MICI Algorithm [10]**TRAINING****Require:** Training Data, Training Labels, Parameters

```
1: Initialize a population of measures                                ▷ 3.4.1
2:  $F^* = \max(\mathbf{F}_P^0)$ ,  $\mathbf{g}^* = \arg \max_{\mathcal{G}} \mathbf{F}_P^0$ 
3: for  $t := 1 \rightarrow I$  do
4:   for  $p := 1 \rightarrow P$  do
5:     Evaluate valid intervals of  $\mathcal{G}\{p\}$                                 ▷ 3.4.2
6:     Randomly sample  $z \in [0, 1]$ 
7:     if  $z < \eta$  then                                                 ▷ 3.4.3
8:       Update  $\mathcal{G}\{p\}$  by small-scale mutation
9:     else
10:      Update  $\mathcal{G}\{p\}$  by large-scale mutation
11:    end if
12:  end for
13:  Evaluate fitness of updated measures using (3.5)
14:  Select measures                                              ▷ 3.4.4
15:  if  $\max(\mathbf{F}_P^t) > F^*$  then
16:     $F^* = \max(\mathbf{F}_P^t)$ ,  $\mathbf{g}^* = \arg \max_{\mathcal{G}} \mathbf{F}_P^t$ 
17:  end if
18: end for
return  $\mathbf{g}^*$ 
```

TESTING**Require:** Testing Data, \mathbf{g}^*

```
19:  $TestLabels \leftarrow$  Choquet integral output computed based on Equation (6.10) using the
   learned  $\mathbf{g}^*$  above
return  $TestLabels$ 
```

$(g_{123}, g_{124}, g_{134}$, and g_{234}) were first sampled randomly between 0 and 1. Then, the $(m - 2)$ -tuple measure elements $(g_{12}, g_{13}, g_{14}, g_{23}, g_{24}$, and g_{34}) were sampled between 0 and its corresponding superset. For example, g_{12} measure element value is sampled randomly between 0 and $\min(g_{123}, g_{124})$, due to monotonicity property. The process goes on until the singletons (g_1, g_2, g_3, g_4) were each sampled between 0 and their corresponding superset values.

Similarly, the “bottom-up” approaches samples measure element values from the bottom of the lattice up. First the singletons were sampled between 0 and 1 randomly. Then, the duples were sampled between its corresponding subsets and 1. For example in the four-source case, the initial value of g_{12} was sampled randomly between $\max(g_1, g_2)$ and 1. The process goes on until the $(m - 1)$ -tuple measure elements were sampled, thus initializing all the element values in the entire measure. Note that the measure element corresponding to the full set is always equal to 1 (e.g. $g_{1234} \equiv 1$ for four sources), according to the normalization property of the fuzzy measure.

In our experiments, the two initialization approaches yield different sets of measure element values but seem to have little impact on the final measure learned. In the following experiments, the measure is initialized by randomly flipping a coin and pick either “top-down” or “bottom-up” initialization approach. Detailed comparison of the top-down versus bottom-up initialization experimental results can be seen in Section 6.4.1.

3.4.2 Evaluation of Valid Intervals

A measure element, when updated, must still satisfy monotonicity. The term “valid interval” is used in this paper to define how large the element value can change without sacrificing monotonicity. The valid interval width for each measure element is set as the difference

between the lower and upper bound for each element. The lower and upper bounds of a measure element are computed as the largest value of its subsets and the smallest value of its supersets, respectively. For example, when number of sources $m = 3$, g_1 and g_2 corresponds to the subsets of measure element g_{12} , while g_{123} corresponds to the superset of g_{12} . Therefore, the lower bound of g_{12} is equal to $\max(g_1, g_2)$ and the upper bound of g_{12} is $g_{123} = 1$. The valid interval width for element g_{12} is thus $1 - \max(g_1, g_2)$. In this step, the valid interval width for each measure element is computed.

3.4.3 Mutation

An evolutionary algorithm is used to learn the fuzzy measure in the MICI algorithm. Mutations of two different scales were designed in search of the optimal solution. The valid intervals computed above are used in determining which measure element to update during both types of mutations.

In the large-scale mutation, all the measure elements are sorted based on their valid interval widths in descending order. Then, all the measure elements are sampled according to their sort order. That is to say, the measure element with the largest valid interval width (and has the largest room to change) will be sampled first, the measure elements with the second largest valid interval width will be sampled second, and so on. The new measure values are sampled from a truncated Gaussian (TG) distribution [273]. Appendix A describes the method to sample from a truncated Gaussian distribution in mathematical detail. The Beta distribution and the full Gaussian distribution have been investigated but the TG provides best results so far. Other distributions, such as Cauchy, can be explored in the future. The lower and upper bounds of the truncated Gaussian are equal to the lower and upper bounds of the valid interval of the corresponding measure element.

In the small-scale mutation, only one measure element is sampled. The element to be sampled is chosen by randomly sampling from a multinomial distribution based on the valid intervals of all the measure elements. The probability of sampling a particular measure element g_l is set to

$$P(g_l) = \frac{w_l}{\sum_{o=1}^{2^C-1} w_o}, \quad (3.15)$$

where w_l is the valid interval width for measure element g_l . The measure element with the largest valid interval width will, therefore, have the largest probability to be sampled. As in the large-scale mutation, the new measure values are obtained by sampling from a truncated Gaussian (TG) distribution.

The rate of small-scale mutation $\eta \in [0, 1]$ is defined by users. The rate of large-scale mutation is $1 - \eta$.

3.4.4 Selection

The measures retained for the next generation are selected based on their fitness function values computed using Equation (3.5). In each iteration, all measures in the population are sampled, yielding a child measure population of size P . The measure population before sampling is regarded as the parent measure population (size P). Both the parent and child measure populations are pooled together (size $2P$) and their fitness values are computed using Equation (3.5). Then, $P/2$ measures with the top 25% fitness values are kept and carried over to the next iteration (elitism), and the remaining $P/2$ measures to be carried over are sampled according to a multinomial distribution based on their fitness values from the remaining 75% of the parent and child population pool, following a similar approach to Equation (3.15). Among the new measure population, the measure with the highest fitness

value is kept as the current best measure g^* . The process continues until a stopping criterion is reached, such as when the maximum number of iterations is reached or the change in the objective function value from one iteration to the next is smaller than a fixed threshold.

At the end of training process, the best measure g^* with the highest fitness value so far is returned as the learned measure and used for testing.

Chapter 4

Multiple Instance Choquet Integral Regression

This chapter describes the proposed Multiple Instance Choquet Integral Regression (MICIR) algorithm. The Multiple Instance Choquet Integral Regression model is proposed to solve regression problems where the desired prediction values are real numbers. The proposed MICIR algorithm adopts the “primary-instance” assumption that there is one primary instance responsible for the label for each bag [25]. The MICIR algorithm fuses multiple sources with real-valued label as well as taking into account the uncertainties in the label.

Given a set of training data and bag-level training labels, MICI Regression learns a fuzzy measure to be used with the Choquet integral and the CI value is used to perform classification and regression on the testing data. The following objective function is proposed to optimize the fitness given a fuzzy measure \mathbf{g} :

$$\min \sum_{b=1}^{N_b} \min_{\forall i, x_{bi} \in X_b} (C_{\mathbf{g}}(\mathbf{x}_{bi}) - d_b)^2, \quad (4.1)$$

where N_b is the total number of training bags, d_b is the desired training bag-level label for

bag b , X_b is the set of all the instances in bag b , $C_g(\mathbf{x}_{bi})$ is the choquet integral output for the i^{th} instance in bag b given fuzzy measure g . The Choquet integral output is computed based on Equation (6.10). The objective function encourages the Choquet integral value of one instance in the bag b to the desired real-valued label d_b .

The proposed MICIR is applicable to two-class classification problems with uncertain labels as well. For two-class classification problems, the desired training bag label $d_b = 1$ for positive bags and $d_b = 0$ for negative bags. In this formulation, a positive bag contains at least one positive instance and each negative point forms its own negative bag, as the MIL assumes that a bag is labeled negative if all of the instances in the bag is negative. Notice that when using MICIR for classification, the CI values should correspond to the class labels. That is to say, the desired CI values should be high (close to 1) for positive class (class label “+1”) and low (close to 0) for negative class (class label “0”). This is due to the objective function encouraging all the negative points to label “0” and encouraging one of the instances in the positive bags to be “+1”, following the MIL framework. This assumption fits applications such as target detection well, where the CI values represent the confidence values. The higher the confidence value (the CI value) of a pixel/data point, the more likely it is a target. For multi-class classification, the class labels must be carefully selected, as the fuzzy measure used in this dissertation is limited to between $[0, 1]$ (normalization property). Future work can include investigating MICIR for multi-class classification problems.

Optimization scheme of MICI Regression follows the MICI framework described in Chapter 3. The same evolutionary algorithm is used to optimize the MICIR algorithm using training data and the optimal fuzzy measure obtained from the training is used to compute the CI value for testing.

Chapter 5

Multi-Resolution Multiple Instance Choquet Integral

This chapter describes the proposed Multi-Resolution Multiple Instance Choquet Integral (MR-MICI) framework that takes in heterogeneous data (for example, images at different resolutions or data of different geospatial types) from multiple sensors with uncertainty in training labels and perform fusion, considering uncertainties in training labels. The proposed MR-MICI algorithm works under the assumption that at least one point in each bag from each data source is accurate but the remainder data points in the bags can have uncertain labels. The proposed MR-MICI algorithm also learns a monotonic normalized fuzzy measure to be used with the Choquet integral to perform fusion on heterogeneous data sources.

The proposed Multi-Resolution Multiple Instance Choquet Integral algorithm is able to fuse heterogeneous data sources (for example, image data at different resolutions or data of different geo-spatial data types) with consideration of uncertain training labels. As

described in the Introduction chapter, traditional fusion methods only work with homogeneous data types where data points are assumed to be corresponding across sources exactly. However, previous methods to accommodate this restriction such as image co-registration and rasterization may cause a loss of information or add noise with problems such as misalignment. Besides, the GPS devices are often only accurate to the level of several pixels, which makes it difficult to pinpoint accurate pixel-level locations, let alone mapping exact pixel correspondence between sources. The proposed Multi-Resolution Multiple Instance Choquet Integral uses the MIL framework to take into account the variability of corresponding data points from each source and is able to fuse sources at different resolutions.

The objective function for Multi-Resolution Multiple Instance Choquet Integral is proposed as follows:

$$\sum_{a=1}^{B^-} \max_{\forall \mathbf{S}_{ai}^- \in \mathcal{B}_a^-} \left(\min_{\forall \mathbf{x}_k^- \in \mathbf{S}_{ai}^-} C_g(\mathbf{x}_k^-) - 0 \right)^2 + \sum_{b=1}^{B^+} \min_{\forall \mathbf{S}_{bj}^+ \in \mathcal{B}_b^+} \left(\max_{\forall \mathbf{x}_l^+ \in \mathbf{S}_{bj}^+} C_g(\mathbf{x}_l^+) - 1 \right)^2, \quad (5.1)$$

where B^+ is the total number of positive bags, B^- is the total number of negative bags, \mathbf{S}_{ai}^- is the collection of i^{th} instance set in the a^{th} negative bag and similar for \mathbf{S}_{bj}^+ . C_g is the Choquet integral given measure g , \mathcal{B}_a^- is the a^{th} negative bag, and \mathcal{B}_b^+ is the b^{th} positive bag.

The outside min and max operators over negative and positive bags are comparable to the MICI min-max model. We encourage the Choquet integral of all the points in the negative bag to zero (first term) and encourage the Choquet integral of at least one of the points in the positive bag to one (second term), which satisfies the MIL assumption.

The inner max and min operators, on the other hand, are used to handle multi-resolution data sources. Suppose we would like to fuse hyperspectral imagery with LiDAR point cloud data. Suppose the hyperspectral imagery has the ground sample distance (GSD) of

one meter, which means each pixel in the HSI imagery covers $1m^2$ area. There could be more than one LiDAR data points inside the $1m^2$ area, as described in Introduction. Figure 5.1 shows an illustration for multiple LiDAR points (on the right) corresponding to one pixel in HSI imagery on the left.

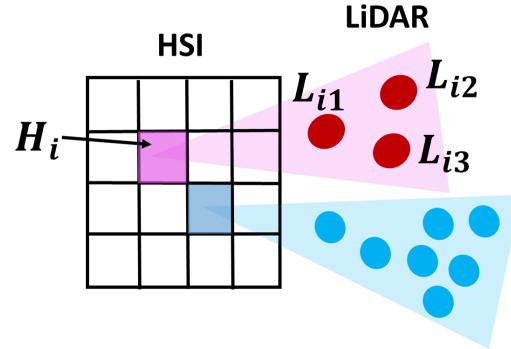


Figure 5.1: Illustration for HSI and LiDAR fusion. All LiDAR data points in the pink shade fell in the same area covered by the pink pixel in the HSI image, and all LiDAR data points in the blue shade corresponds to the blue pixel in the HSI image. Note there could be more than one LiDAR points in the area covered by one pixel in the HSI imagery.

Denote the value of the i^{th} pixel in the HSI imagery as H_i . Suppose three LiDAR points fell in the same one-pixel area, as illustrated in pink in Figure 5.1. Denote the values of the three LiDAR points as L_{i1}, L_{i2}, L_{i3} . The collection of the sensor outputs for pixel i can be written as:

$$\mathbf{S}_i = \begin{bmatrix} H_i & L_{i1} \\ H_i & L_{i2} \\ H_i & L_{i3} \end{bmatrix} \quad (5.2)$$

For each collection of the sensor sources, we assume at least one of the combinations gives the desired label. In the example above, we assume that at least one of the LiDAR points provides accurate information corresponding to the HSI pixel. Therefore, in the ob-

jective function, the inner max and min operators are picking one of all combinations that provides accurate information, and thus taking multi-resolution data sources into consideration.

Optimization scheme of Multi-Resolution MICI follows the MICI framework described in Chapter 3. An optimal fuzzy measure is obtained using an evolutionary algorithm in training. In testing, the fuzzy measure is used to compute the CI values for all combinations of the sensor sources and the inner min and max operators are used to pick one of the combinations as the test label.

Chapter 6

Experimental Results

This chapter presents experimental results of the proposed algorithms on both synthetic data and real applications such as target detection and scene understanding applications in remote sensing imagery. Results indicate that the proposed MICI framework can successfully learn a set of fuzzy measure to be used with the Choquet integral to effectively perform classifier fusion and regression while dealing with uncertainties in training labels. Results also indicate that the proposed MR-MICI framework can successfully perform classifier fusion and yield better classification accuracy when compared with traditional approaches on heterogeneous data (multi-resolution data and data of different geospatial types such as HSI and LiDAR) while taking into account uncertain labels.

6.1 Classification

This section presents results on classifier fusion using the proposed Multiple Instance Choquet Integral (MICI) framework and the noisy-or, min-max, and generalized-mean models

within the MICI framework. Experiments were conducted on both synthetic classification data set as well as a real target detection application on the MUUFL Gulfport hyperspectral data set.

6.1.1 Synthetic 3-Source Classification Data Set

A synthetic classification data set was constructed with 1000 data points. Three individual classifiers were applied to the data set and produced classifier output of either “1” or “0” for each data point. Figure 6.1a to 6.1c show the scatter plot of the three classifier outputs for all the data points. The data points are plotted on the 2-D plane only for visualization, with the color indicating the classifier output source value.

These data points are randomly grouped into 50 bags with 20 points per bag. Each positive bag contains 25% positive points and 75% negative points. MICI is used to perform classifier fusion of the three classifiers (three sources). Figure 6.1d shows the desired true labels. As can be seen in Figure 6.1, source 1 and source 2 are useful in producing the true labels whereas source 3 does not provide useful information.

One example of a measure learned by the MICI noisy-or model is presented in Table 6.1. MICI successfully learns the interaction between the sources, i.e., putting high measure value on the intersection of source 1 and source 2 (g_{12}) and lower values otherwise. The proposed algorithm achieves 100% classification accuracy after fusing the three sources, consistently over multiple runs. The MICI min-max model and the MICI generalized mean model yield similar results.

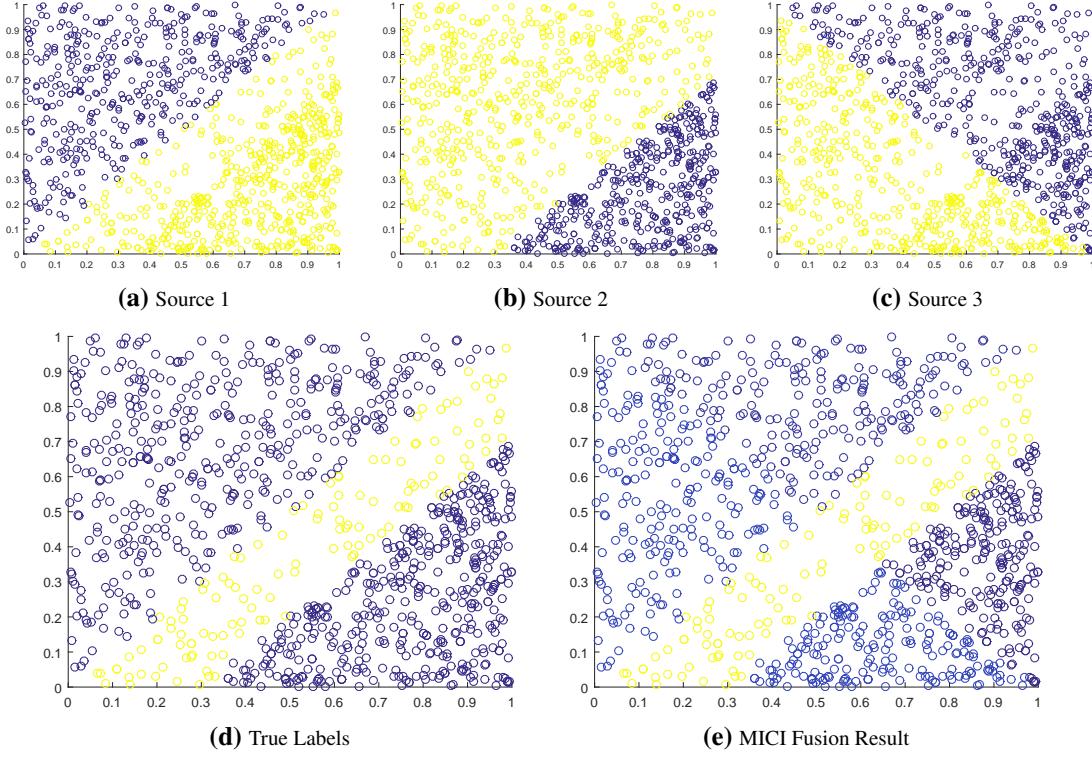


Figure 6.1: Synthetic 3-source dataset and results for MICI classifier fusion model. (a) classifier 1 output; (b) classifier 2 output; (c) classifier 3 output; (d) true labels; (e) MICI fusion result. The 1000 data points are scatter plotted in the 2-D space for visualization. The colors indicate the classification label for each data point according to the colorbar shown in Figure 6.2, bright yellow means “1” and deep blue means “0”.

6.1.2 Synthetic Lane-Based Target Detection Data Set

A synthetic 5-source classification data set was constructed based on the following scenario inspired by real target detection applications. Suppose five detector results were obtained on a target detection data set and MICI was used to perform fusion on the five detector outputs. Each detector result is one source for MICI. There are a total of ten true targets in the $120m \times 40m$ scene, where 1 pixel has a resolution of $1m$ in both directions. Five targets are metal and five targets are plastic. Among the five detectors, Detectors No. 1 and No. 2 can detect metal well and Detectors No. 3 and No. 4 can detect plastic well. Detector

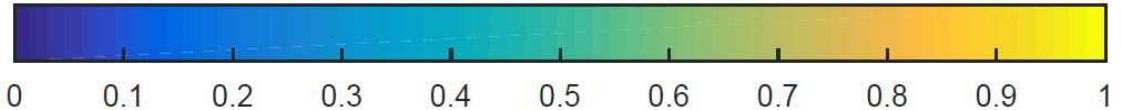


Figure 6.2: Colorbar from $[0, 1]$.

Table 6.1: Mean and standard deviation of estimated and true measure element values learned by MICI noisy-or model for synthetic 3-source MICI classification data set over three runs.

	g_1	g_2	g_3
True	0	0	0
Learned	0.065(0.038)	0.046(0.046)	0.096(0.004)
	g_{12}	g_{13}	g_{23}
True	1	0	0
Learned	0.998(0.001)	0.117(0.011)	0.110(0.008)

No. 5, however, performs very poorly on the targets and triggers high confidence values on non-target locations. There are 5 clutter (non-target) objects detected by each detector. The targets and clutter objects are generated with Gaussian filters with mean intensity value of 0.9. Figure 6.3 shows one data set example. Each $120m \times 40m$ lane in Figure 6.3 represents one detector result. The red boxes mark the true target locations. The above generation process was repeated five times with different true target and clutter locations each time. Now, five $120 \times 40 \times 5$ data sets were generated for the experiment.

The Reed-Xiaoli (RX) detector [274] was applied to the data set and connected component analysis was conducted on each RX detection result. RX is an anomaly detector. For each connected component found in each RX detection result, the centroid location was found. To construct training bags for the MICI algorithm, a 6×6 halo was put around each centroid obtained from the connected components. The number of training bags is the number of connected components and all the pixels within the halo size of each centroid are instances in the training bag. If any of the pixels (instances) in a training bag is within

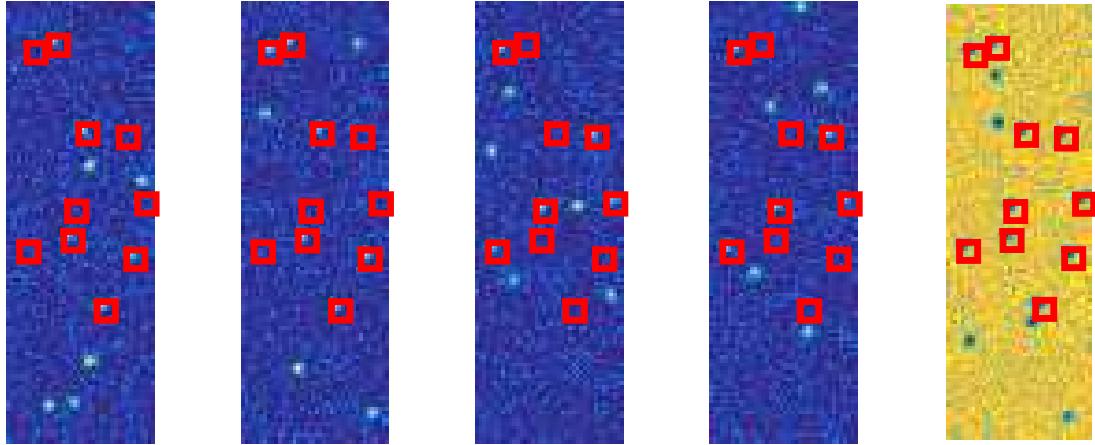


Figure 6.3: One example of the synthetic lane-based target detection data set. Each rectangular image represents one detector output - Detectors No. 1 to No. 5 from left to right. Each rectangular image has 120m (120 pixels) in the vertical direction and 40m (40 pixels) in the horizontal direction. The red boxes mark the true target locations.

the halo of true target locations, that bag is labeled positive; Otherwise, the bag is labeled negative. Figure 6.4 shows one example of RX results. The squares marks the halo and all the points inside the halo belongs to a bag. The red squares form the true positive bags and the white squares are the negative bags.

The colorbar used in all images in this section is shown in Figure 6.2.

Five-fold cross validation was conducted with four data sets as training and one as testing. One example of the measures learned after training on data sets 1, 2, 3, 4 and testing on data set 5 is shown in Table 6.2. The MICI learns near-zero value on source 5 and relatively higher measure values on the combination of source 1 and 2 (g_{12}), for example. This fits the construction of the data set where detectors 1-4 provides information on both plastic and metal targets, while detector 5 does not provide positive reinforcement towards targets.

Figure 6.5 shows the relationship between the fitness values versus the number of iter-

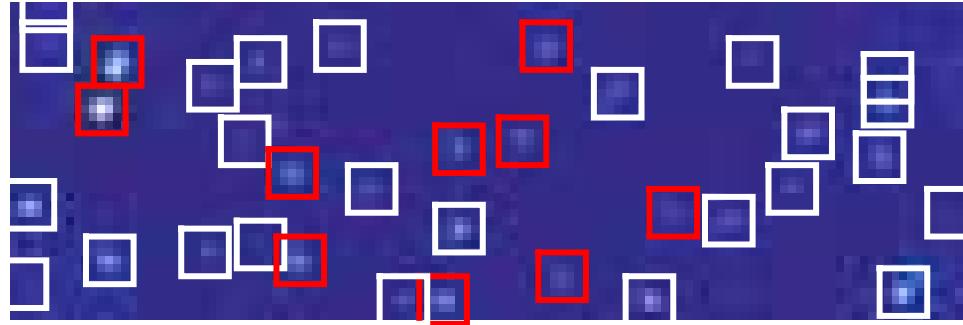


Figure 6.4: RX detection output (plotted horizontally) of the synthetic lane-based target detection data set shown in Figure 6.3. The squares marks the 6×6 halo and all the points inside the halo construct a bag. The red squares marks the true positive bags and the white squares are the negative bags.

ations. The fitness values in the first 1000 iterations over five runs are plotted. As can be seen in Figure 6.5, the fitness values in the first 300 iterations over five runs are increasing rapidly, and the fitness values are increasing more slowly beyond the first few hundred iterations.

Table 6.3 shows the positive detection (PD) vs. false alarm rate (FAR) results after the five-fold cross validation across five runs. MICI algorithm is able to achieve 92.8% positive detection (PD) rate when $FAR = 0.002$. and continues to increase to near 100% positive detection with $FAR = 0.004$.

6.1.3 Synthetic 5-Source Classification Data Set For Varying Parameters

An additional synthetic classification data set was constructed to investigate the relationship between the MICI fusion performance and varying parameters concerning the structure of the training bags. A five-source classification data set was constructed with 100 bags and 10 data points per bag. Half of the bags are positive (with label “1”) and half are negative

Table 6.2: Mean and the standard deviation (in parentheses) of estimated measure element values learned for synthetic 5-source lane-based classification data set over five runs.

g_1	g_2	g_3	g_4	g_5
0.001(0.001)	0.001(0.001)	0.000(0.000)	0.001(0.001)	0.000(0.000)
g_{12}	g_{13}	g_{14}	g_{15}	g_{23}
0.009(0.005)	0.003(0.001)	0.004(0.003)	0.001(0.001)	0.007(0.004)
g_{24}	g_{25}	g_{34}	g_{35}	g_{45}
0.009(0.011)	0.003(0.002)	0.002(0.003)	0.001(0.000)	0.001(0.001)
g_{123}	g_{124}	g_{125}	g_{134}	g_{135}
0.013(0.008)	0.020(0.014)	0.010(0.006)	0.006(0.002)	0.004(0.001)
g_{145}	g_{234}	g_{235}	g_{245}	g_{345}
0.006(0.004)	0.017(0.007)	0.007(0.004)	0.010(0.011)	0.003(0.004)
g_{1234}	g_{1235}	g_{1245}	g_{1345}	g_{2345}
0.050(0.028)	0.014(0.007)	0.020(0.014)	0.008(0.002)	0.018(0.007)

(with label “0”). According to the assumption of MIL, the negative bags must contain all negative points but the positive bag only needs to contain at least one positive points.

The effect of “contamination” was investigated. Assume half of the training bags are positive and contain 100% positive points. Then, a varying percentage of positive points were added to the negative bags so that the negative bags were “contaminated.” Relative error [57] is used to evaluate the performance for the MICI classification models:

$$Error_{reg}(y, \hat{y}) = |y - \hat{y}|, \quad (6.1)$$

where y is the true label (“1” or “0”) and \hat{y} is the estimated label for each data point.

The relative error results with MICI noisy-or model, min-max model, and generalized mean model were recorded and presented in Tables 6.4, 6.5, and 6.6. As can be seen, in all three models, the relative error increases as the percentage of contamination increases. When the percentage of contamination goes towards 100%, the error is nearly 1 (100%

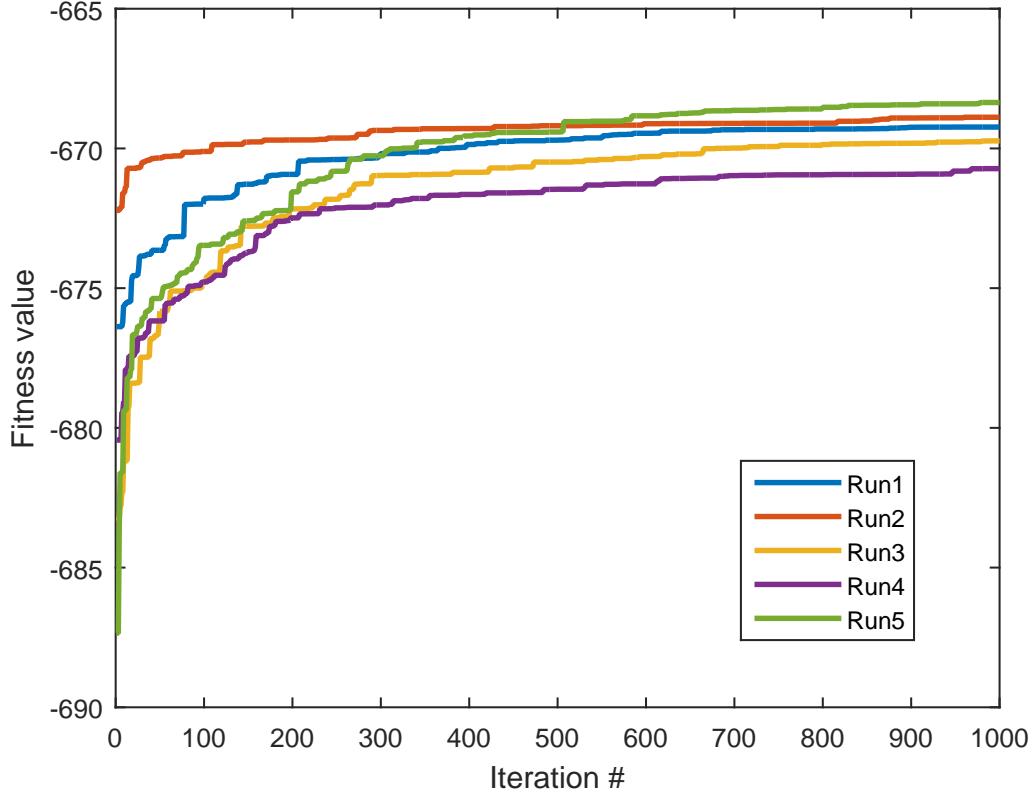


Figure 6.5: Relationship of fitness values vs. number of iterations in the synthetic lane-based target detection experiment. The results in the first 1000 iterations are shown over five runs.

wrong) as well, which makes sense as the negative bags are now filled with positive points and it is not possible to distinguish positive and negative points. When the percentage of contamination increases towards 100%, the relative error also increases as the difficulty grows.

6.1.4 MUUFL Gulfport Target Detection

The MICI classification models were tested on a real target detection experiment using the MUUFL Gulfport hyperspectral data set. The MUUFL Gulfport hyperspectral data set

Table 6.3: The positive detection and false alarm rate of the synthetic lane-based target detection data set after five-fold cross validation across five runs.

		FAR ($/m^2$)						
		5×10^{-5}	1×10^{-4}	4×10^{-4}	1×10^{-3}	2×10^{-3}	4×10^{-3}	4.2×10^{-3}
PD (%)	Run1	18.0	26.0	54.0	70.0	94.0	100.0	100.0
	Run2	18.0	26.0	50.0	70.0	92.0	100.0	100.0
	Run3	20.0	24.0	54.0	70.0	92.0	98.0	100.0
	Run4	0.0	24.0	50.0	68.0	92.0	98.0	100.0
	Run5	18.0	26.0	50.0	68.0	94.0	98.0	100.0
	Total	14.8(8.3)	25.2(1.1)	51.6(2.2)	69.2(1.1)	92.8(1.1)	98.8(1.1)	100.0(0.0)

Table 6.4: Relative error versus contamination for synthetic classification data set for MICI noisy-or model across five runs.

	Percentage of contamination					
	0%	10%	20%	30%	40%	50%
Relative Error	0.371(0.026)	0.492(0.008)	0.534(0.007)	0.588(0.007)	0.630(0.012)	0.668(0.004)
	60%	70%	80%	90%	100%	
Relative Error	0.697(0.014)	0.751(0.012)	0.787(0.019)	0.812(0.012)	0.868(0.014)	

[275] was collected over the University of Southern Mississippi Gulf Park Campus. The data set used in this experiment consists of three hyperspectral data cubes collected on three separate flights at an altitude of 3500' over the campus area. The HSI data cubes have a ground sample distance of 1m.¹ The image from campus 1 is 325×337 pixels in size. The image from campus 3 is 329×345 pixels in size. The image from campus 4 is 333×345 pixels in size. All HSI data cubes contain 72 bands corresponding to wavelengths 367.7nm to 1043.4nm and were collected using the CASI hyperspectral camera [275, 276]. In this experiment, the first four and last four noisy bands were removed.

A total of sixty cloth panel targets were placed in the scene. The targets were cloth panels of five different colors: fifteen brown, fifteen dark green, twelve faux vineyard green (FVG), fifteen pea green, and three vineyard green. Figure 6.6 shows the RGB images of

¹The data set is available at <https://github.com/GatorSense>. The three flights used in this experiment corresponds to “muufl_gulfport_campus_w_lidar_1.mat”, “muufl_gulfport_campus_3.mat”, and “muufl_gulfport_campus_4.mat”.

Table 6.5: Relative error versus contamination for synthetic classification data set for MICI min-max model across five runs.

	Percentage of contamination					
	0%	10%	20%	30%	40%	50%
Relative Error	0.288(0.025)	0.514(0.003)	0.536(0.011)	0.570(0.017)	0.655(0.005)	0.686(0.005)
	60%	70%	80%	90%	100%	
Relative Error	0.255(0.171)	0.217(0.191)	0.359(0.304)	0.579(0.255)	0.345(0.366)	

Table 6.6: Relative error versus contamination for synthetic classification data set for MICI generalized mean model across five runs.

	Percentage of contamination					
	0%	10%	20%	30%	40%	50%
Relative Error	0.247(0.037)	0.517(0.002)	0.551(0.006)	0.595(0.005)	0.660(0.002)	0.694(0.007)
	60%	70%	80%	90%	100%	
Relative Error	0.709(0.020)	0.744(0.022)	0.783(0.027)	0.821(0.017)	0.882(0.011)	

the two flights over the campus and the true target locations. These panels varied from sub-pixel targets (at $0.25m^2$ corresponding to a quarter of a pixel in area) up to super-pixel targets (at $9m^2$) with varying levels of occlusion. For each target, a GPS ground truth location was collected using a Trimble Juno SB hand-held device. The device has accuracy up to 5m. Thus, the groundtruth locations for each target are only accurate within a 5×5 pixel halo. Given the inaccuracy in the groundtruth, a multiple instance learning approach that can address this is required.

The adaptive coherence estimator (ACE) detector [277–279] was applied to the imagery using spectral signatures of four of the target types (as spectral signatures for these targets were available from previous studies²). Their spectral signatures can be seen in Figure 6.7. The background mean and background covariance for the ACE detector was estimated using the global mean and covariance of all the pixels in each image. Note that the signed ACE detector used yields confidence values between $[-1, 1]$. Therefore, all ACE results used in this experiment were normalized to be between zero and one by adding 1 to

²The target spectra used in this experiment come from “tgt_img_spectra .mat” in the data set. The four target types are: brown, dark green, FVG and pea green.

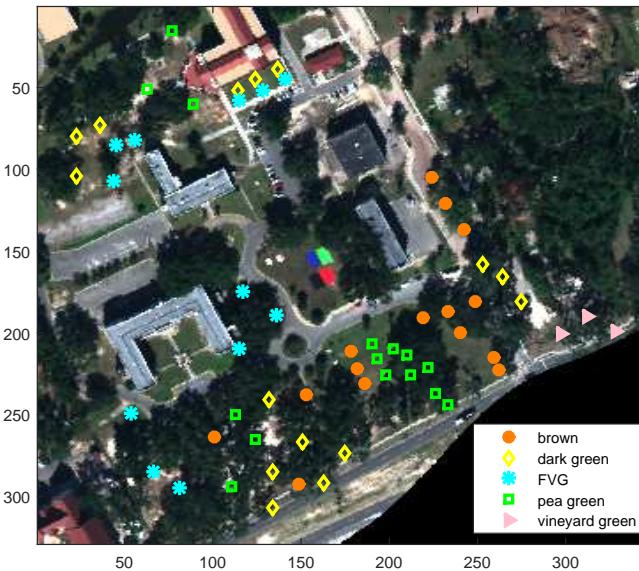


Figure 6.6: The RGB image from MUUFL Gulfport “campus 3” data set. Orange circle marks the true brown target locations, yellow diamond marks the true dark green target locations, cyan asterisk marks the true FVG target locations, green square marks the true pea green target locations, and pink triangle marks the true vineyard green target locations.

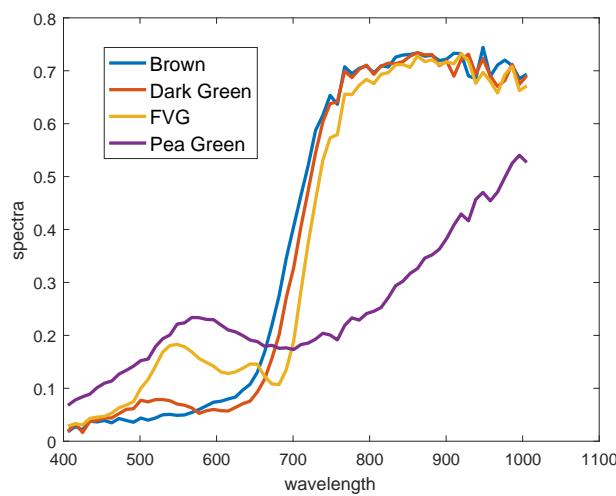


Figure 6.7: The spectral signatures for brown, dark green, FVG, and pea green targets in the MUUFL Gulfport data set. The unit for the wavelength is nanometers.

the original ACE results and divide by 2. All target signature and data pixel values were normalized to have norm 1.

Cross validation is performed on this data set, i.e., training on campus 1 and testing on campus 3 and campus 4, and so on. First, the mean and covariance of the training imagery, μ_{tr} and cov_{tr} , were computed. Then, the ACE detection map for the training imagery, ACE_{tr} , was obtained using μ_{tr} and cov_{tr} . Each pixel in the detection map has four dimensions, each dimension corresponds to ACE confidence values for four target types. Each of the ACE results highlights different locations corresponding to different targets. The four dimensions of the ACE results will be the sources for the MICI to fuse.

To construct training bags for MICI, a 5×5 window was put around each groundtruth target location of the training imagery. Each window forms a positive bag and all the pixels in the windows are instances in the positive bag. The size of the positive bag corresponds to the accuracy of the GPS device used to collect the groundtruth points. One negative bag was constructed by randomly picking 1600 background pixels that do not belong to any of the windows. The positive bags were labeled “1” and the negative bag was labeled “0”. MICI was applied to the training data and a fuzzy measure g^* is learned.

The ACE detection results for the test imagery, ACE_{te} , were computed using training mean and covariance μ_{tr} and cov_{tr} . Then, the Choquet integral given the learned measure g^* and ACE_{te} was computed. The Choquet integral result is the MICI fusion result.

Experiments were conducted on the original, un-normalized HSI data as well as normalized HSI data using three different normalization methods. The “un-normalized” version uses directly the HSI data and target signature values without any normalization. In the first experiment using normalized data, the hyperspectral image cube and the target signature

was normalized by dividing over the norm as shown in Eq.(6.2).

$$I_{norm} = \frac{I}{\|I\|}, \quad (6.2)$$

where I is the data to be normalized, $\|\cdot\|$ is the Euclidean norm (2-norm) of I .

The second normalization method uses a unity-based normalization, where the hyperspectral data were normalized and scaled between 0 and 1 as shown in Eq.(6.3).

$$I_{norm} = \frac{I - \min(I)}{\max(I) - \min(I)}, \quad (6.3)$$

where I is the data to be normalized, $\max(I)$ and $\min(I)$ are the max and min value of the data I , respectively.

The third normalization method uses a Gaussian distribution-based normalization, where the hyperspectral data were normalized by the mean and standard deviation as shown in Eq.(6.4).

$$I_{norm} = \frac{I - \mu}{\sigma}, \quad (6.4)$$

where I is the data to be normalized, μ is the mean and σ is the standard deviation of the data I , respectively.

The proposed MICI noisy-or, MICI min-max, and MICI soft-max models were first compared with the four ACE results using the four target signatures (Brown, Dark Green, FVG and Pea Green) described above. The results are scored over all the targets in the scene with known groundtruth. The fusion results were then compared with a standard Support Vector Machine (SVM) on the four ACE maps, and taking the max, min, or mean over the four ACE results. The MICI models were also compared with CI-QP [73], mi-SVM [40], and EM-DD [32] methods. CI-QP learns a fuzzy measure for Choquet integral by optimizing a least squares error objective using Quadratic Programming. The CI-QP

approach assumes an accurate label for every training data point and, thus, does not inherently support MIL-type learning. In our application of CI-QP to this problem, we gave all points in a positive bag the label of “1” and all points in the negative bag as a label of “0”. The mi-SVM and EM-DD are both multiple instance learning approaches. The mi-SVM method is a Support Vector Machine classification approach with an extension for Multiple Instance Learning. The EM-DD is an Expectation-Maximization (EM) version of the diverse density algorithm [26], which iteratively selects an instance from the training bags and updates the estimated target “concept” based on the selected instances.

All the comparison methods were run on un-normalized and normalized MUUFL Gulfport data. The receiver operating characteristic (ROC) curve is used to evaluate the target detection results. The ROC curve plots the positive detection rate (PD, Y-axis) against the false alarm rate (FAR, X-axis). The performance of the algorithms was evaluated by (1) computing the area under curve (AUC) results with FAR up to $1 \times 10^{-3}/m^2$ (corresponding to a reasonable scale of 1 false alarm in $1000m^2$) and (2) the positive detection rate versus the false alarm rate.

Table 6.7 shows the AUC results at FAR up to $1 \times 10^{-3}/m^2$ for all fusion methods with un-normalized hyperspectral data. Table 6.8, Table 6.9, and Table 6.10 show the AUC results at FAR up to $1 \times 10^{-3}/m^2$ for all fusion methods with normalized hyperspectral data using the three normalization methods as shown in Equations (6.2)-(6.4), respectively. In all the AUC tables in this section, the “best” performance was determined by comparing the mean value of the AUC results, and in the case where the mean is the same between methods, the one result with smaller standard deviation is preferred. The best two results were **bolded** and underlined, respectively.

In all the AUC experiments, the proposed MICI models have the most bolded and/or

underlined results, which outperforms with the best of comparison results in most cases. Taking the max of the ACE results also perform quite well in a few cases, which is understandable as the goal of this experiment is to detect all targets across the four target types and the max operator is capable of merging the four ACE confidence maps as expected. The MICI models learned were able to match the max operator in these cases.

Considering the best target detection AUC results, the normalization by mean and standard deviation method shown in Eq. (6.4) performs better than other normalization methods when trained on campus 1 and 4, and using un-normalized data works well when training on campus 3 data. This could be because the HSI data from campus 1 and 4 fits a normal distribution better. To confirm this observation, quantile-quantile (Q-Q) plots [280, 281] were used to determine the difference between the hyperspectral data and the normal distribution representation. The Q-Q plot compares the similarity of two distributions by plotting their inverse cumulative distribution values against one another at regular intervals [281]. On the other hand, the beta distribution can be a good fit to measured hyperspectral data for some materials [281] especially if the data is limited between zero and one. The Q-Q plot between the original data and a Gaussian distribution is compared with a beta distribution representation. After computing the Q-Q plot, it was found that $\leq 82.1\%$ of points in campus 1 HSI data and $\leq 86.3\%$ of points in campus 4 HSI data fits a normal distribution better, while around 74% of data points in campus 3 fits a normal distribution better. The normal distribution, therefore, deems a better fit for campus 1 and 4 and the normalization method that is based on a Gaussian mean and standard deviation will work better for campus 1 and 4 data.

The ROC curve plots were shown for results of one run from each normalization method on each cross validation. Figure 6.8 to Figure 6.13 shows the ROC curve results for un-

Table 6.7: The AUC results at on un-normalized MUUFL Gulfport data across five runs.

Notes	Methods	Train1Test3	Train1Test4	Train3Test1	Train3Test4	Train4Test1	Train4Test3
<i>Sources:</i>	Brown	0.265	0.264	0.334	0.267	0.307	0.263
<i>individual target types.</i>	Dark Green	0.266	0.261	0.328	0.256	0.293	0.266
	FVG	0.114	0.106	0.122	0.107	0.136	0.109
	Pea Green	0.088	0.000	0.107	0.000	0.100	0.091
<i>Fusion methods in comparison.</i>	SVM	0.185	0.195	0.164	0.175	0.245	0.220
	min	0.000	0.000	0.046	0.073	0.026	0.023
	max	0.345	0.329	0.459	0.339	0.349	0.328
	mean	0.224	0.221	0.269	0.214	0.260	0.246
	CI-QP	0.328	0.325	0.399	0.330	0.260	0.272
<i>MIL methods.</i>	mi-SVM	0.346	0.337	0.350	0.293	0.317	0.317
	EM-DD	0.062(0.014)	0.073(0.013)	0.002(0.003)	0.021(0.004)	0.005(0.008)	0.021(0.021)
<i>Proposed MICI methods.</i>	MICI Noisy-Or	0.346(0.000)	0.331(0.001)	0.461(0.000)	0.340(0.000)	0.349(0.000)	0.329(0.000)
	MICI Min-Max	0.345(0.001)	0.331(0.008)	0.463(0.010)	0.333(0.013)	0.348(0.002)	0.329(0.001)
	MICI Generalized Mean	0.345(0.000)	0.331(0.005)	0.466(0.004)	0.339(0.001)	0.349(0.001)	0.330(0.002)

Table 6.8: The AUC results at on normalized MUUFL Gulfport data across five runs.

Normalized by dividing over the norm of the data, as shown in Eq.(6.2).

	Train1Test3	Train1Test4	Train3Test1	Train3Test4	Train4Test1	Train4Test3
Brown	0.225	0.332	0.383	0.334	0.269	0.238
Dark Green	0.258	<u>0.307</u>	<u>0.379</u>	<u>0.318</u>	0.285	0.265
FVG	0.079	0.000	0.076	0.081	0.094	0.037
Pea Green	0.099	0.050	0.102	0.059	0.083	0.086
SVM	0.039	0.098	0.046	0.110	0.112	0.158
min	0.035	0.060	0.028	0.046	0.078	0.071
max	<u>0.316</u>	0.251	0.347	0.290	0.350	0.312
mean	0.044	0.127	0.061	0.129	0.086	0.122
CI-QP	0.247	0.232	0.233	0.276	0.264	0.252
mi-SVM	0.314	0.252	0.362	0.323	0.279	0.256
EM-DD	0.010(0.012)	0.002(0.005)	0.020(0.019)	0.013(0.017)	0.018(0.017)	0.022(0.018)
MICI Noisy-Or	0.317(0.000)	0.250(0.001)	0.348(0.001)	0.288(0.001)	0.351(0.000)	0.308(0.007)
MICI Min-Max	0.312(0.010)	0.248(0.002)	0.351(0.007)	0.286(0.007)	<u>0.353(0.005)</u>	0.309(0.005)
MICI Generalized Mean	0.316(0.001)	0.249(0.002)	0.352(0.003)	0.290(0.009)	0.356(0.006)	0.298(0.022)

normalized HSI data. Figure 6.14 to Figure 6.19 shows the ROC curve results for the HSI data normalized by dividing the norm of the data to norm 1. Figure 6.20 to Figure 6.25 shows the ROC curve results for the HSI data normalized by the unity-based normalization. Figure 6.26 to Figure 6.31 shows the ROC curve results for the HSI data normalized by the mean and standard deviation. All results are cross validated by flights. The red, magenta and blue solid lines mark the ROC curve of MICI noisy-or, MICI min-max, and MICI generalized mean model. The green solid line marks the CI-QP method and the black solid line marks the mi-SVM method results. The dashed red line marks the EM-DD method

Table 6.9: The AUC results at on normalized MUUFL Gulfport data across five runs. Normalized by unity-based normalization, as shown in Eq.(6.3).

	Train1Test3	Train1Test4	Train3Test1	Train3Test4	Train4Test1	Train4Test3
Brown	0.258	0.262	<u>0.335</u>	0.254	0.314	0.255
Dark Green	0.271	0.243	0.337	0.229	<u>0.310</u>	0.262
FVG	0.149	0.122	0.151	0.148	0.142	0.116
Pea Green	0.082	0.054	0.102	0.048	0.090	0.000
SVM	0.000	0.075	0.254	0.178	0.256	0.284
min	0.000	0.024	0.332	0.025	0.308	0.030
max	0.367	0.313	0.269	0.342	0.265	<u>0.373</u>
mean	0.233	0.127	0.285	0.170	0.264	0.243
CI-QP	0.375	0.304	0.295	0.320	0.221	0.332
mi-SVM	0.345	0.245	0.221	0.288	0.246	0.286
EM-DD	0.035(0.020)	0.008(0.017)	0.071(0.001)	0.026(0.023)	0.067(0.000)	0.003(0.005)
MICI Noisy-Or	0.365(0.000)	0.314(0.000)	0.269(0.000)	0.342(0.000)	0.265(0.001)	<u>0.373(0.001)</u>
MICI Min-Max	0.368(0.003)	0.315(0.001)	0.269(0.011)	0.342(0.002)	0.276(0.004)	0.375(0.001)
MICI Generalized Mean	0.370(0.005)	0.306(0.017)	0.270(0.001)	0.341(0.003)	0.265(0.019)	0.373(0.004)

and dashed magenta line marks the SVM method. The dashed blue, green and black lines mark the min, max and mean results of the ACE detector maps. The dash-dot red, magenta, blue, and green lines mark the original ACE detector results for Brown, Dark Green, FVG, and Pea Green target types. The X-axis of the ROC curves represents the False Alarm Rate (FAR) between $[0, 0.001]$ and the Y-axis represents Positive Detection (PD).

The Choquet integral has $O(2^m)$ parameters for m sources. To compute the objective function of the proposed MICI noisy-or, the computational complexity of I iterations across B bags and N data points is $O(INB2^m)$. Same goes for MICI min-max, and MICI generalized mean models. Table 6.11 lists the running times and the number of iterations until convergence for five runs of the proposed MICI noisy-or, MICI min-max, and MICI generalized mean models. The algorithm was considered reaching convergence if the change of fitness is below 10^{-4} . Note that convergence is not guaranteed with the evolutionary algorithm. As can be seen from the table, the MICI min-max and the MICI generalized mean models converge in much less iterations and much faster as compared to the MICI noisy-or model, given that their detection results are quite comparable. The MICI min-max

Table 6.10: The AUC results at on normalized MUUFL Gulfport data across five runs. Normalized by the mean and standard deviation, as shown in Eq.(6.4).

	Train1Test3	Train1Test4	Train3Test1	Train3Test4	Train4Test1	Train4Test3
Brown	0.258	0.258	0.343	0.236	0.345	0.251
Dark Green	0.268	0.250	0.343	0.225	0.357	0.255
FVG	0.164	0.152	0.186	0.133	0.164	0.169
Pea Green	0.087	0.057	0.099	0.062	0.107	0.077
SVM	0.226	0.164	0.224	0.042	0.315	0.265
min	0.011	0.036	0.000	0.039	0.040	0.015
max	0.405	0.347	0.412	0.267	0.543	0.412
mean	0.234	0.156	0.228	0.104	0.276	0.239
CI-QP	0.361	0.338	0.439	0.286	0.505	0.338
mi-SVM	0.377	0.329	0.421	0.261	0.517	0.358
EM-DD	0.010(0.022)	0.007(0.010)	0.033(0.021)	0.038(0.023)	0.029(0.020)	0.035(0.024)
MICI Noisy-Or	0.405(0.000)	0.347(0.000)	0.412(0.000)	0.269(0.001)	0.543(0.001)	0.412(0.000)
MICI Min-Max	0.404(0.006)	0.340(0.013)	0.425(0.015)	0.273(0.005)	0.543(0.001)	0.398(0.016)
MICI Generalized Mean	0.404(0.003)	0.347(0.003)	0.415(0.003)	0.276(0.007)	0.544(0.000)	0.409(0.001)

model is the fastest among the three models, as the MICI min-max model only takes either a max or a min operation over the positive or negative bags, and therefore limit the time needed to compute the noisy-or or the generalized mean objective functions.

Table 6.11: Running time (seconds) and number of iterations until convergence for MICI models comparison.

	MICI noisy-or		MICI min-max		MICI generalized mean	
	NumIteration	Run Time	NumIteration	Run Time	NumIteration	Run Time
Run 1	5000	2698.913s	95	44.940s	75	59.342s
Run 2	1543	802.690s	123	54.978s	31	24.040s
Run 3	5000	2733.617s	59	34.411s	82	75.877s
Run 4	5000	3064.963s	162	88.156s	148	132.799s
Run 5	3281	1903.074s	71	32.665s	272	204.593s
Summary		2240.651(910.247)s		51.030(22.608)s		99.330(70.743)s

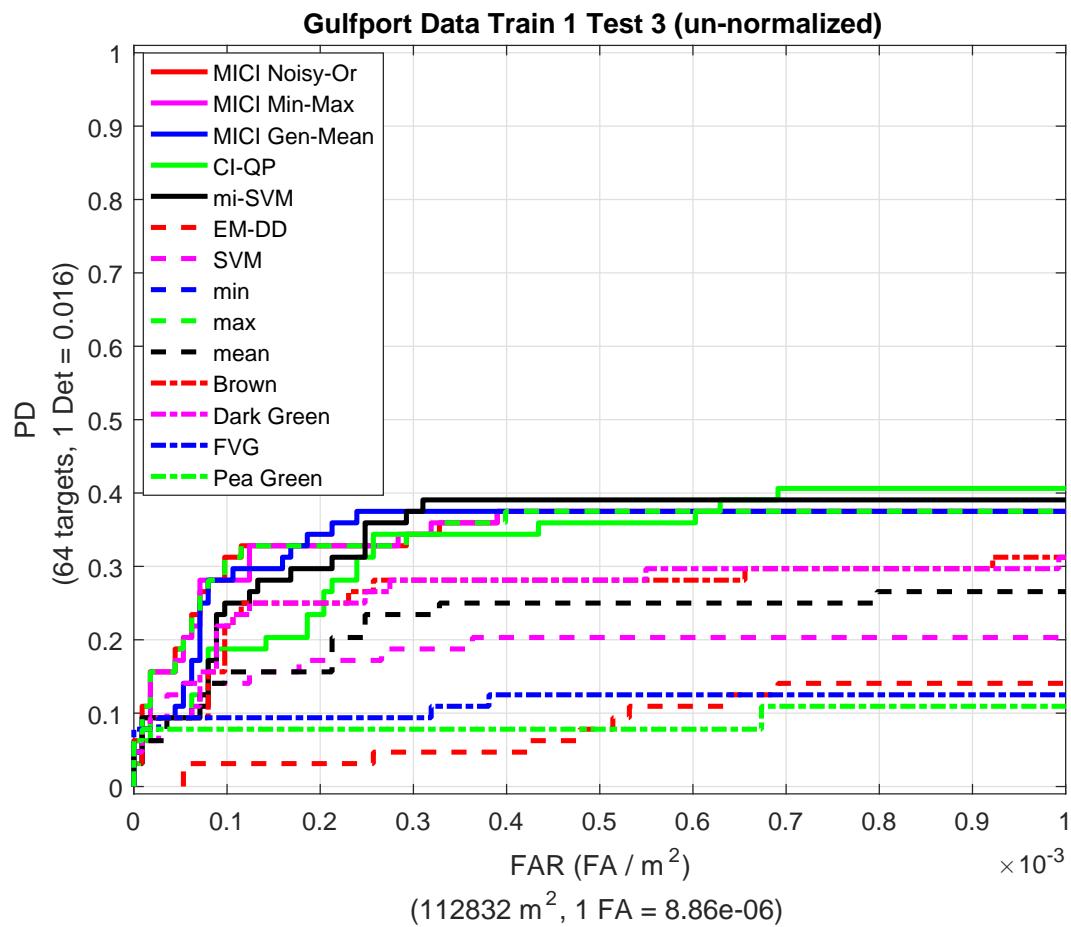


Figure 6.8: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were un-normalized.

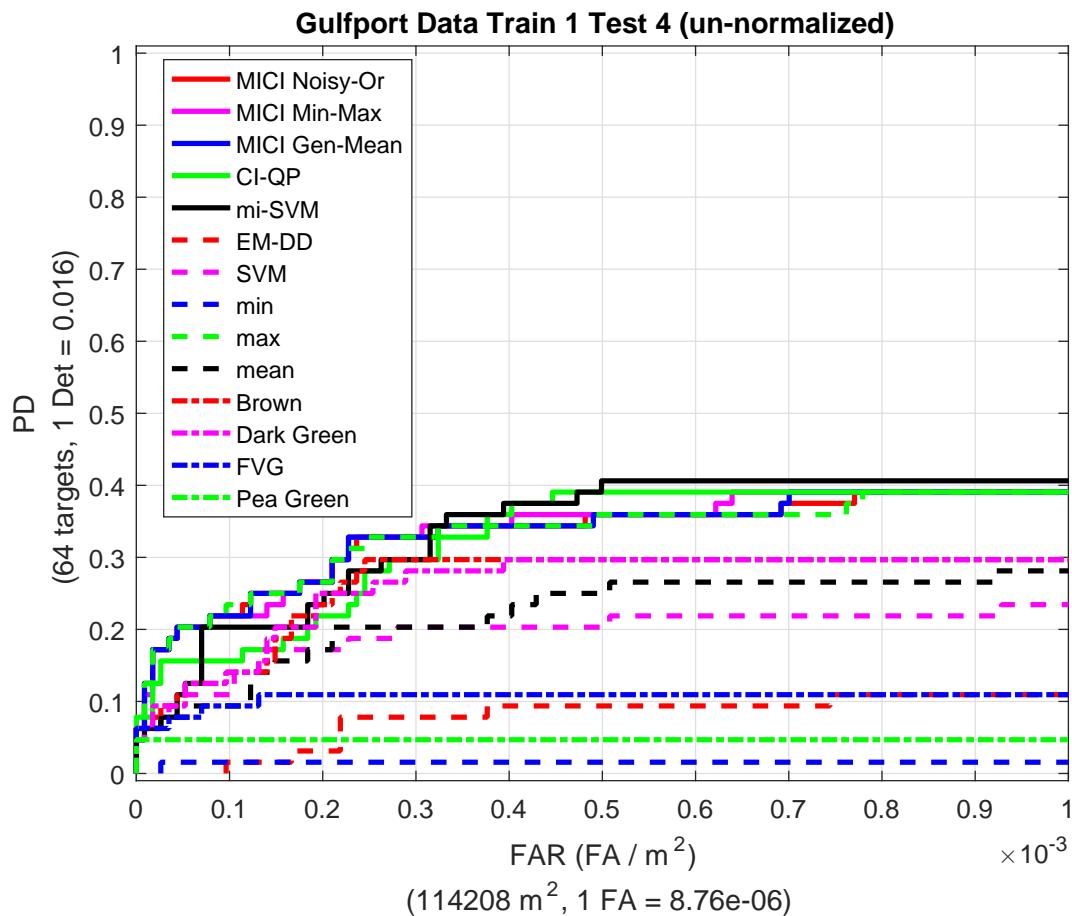


Figure 6.9: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were un-normalized.

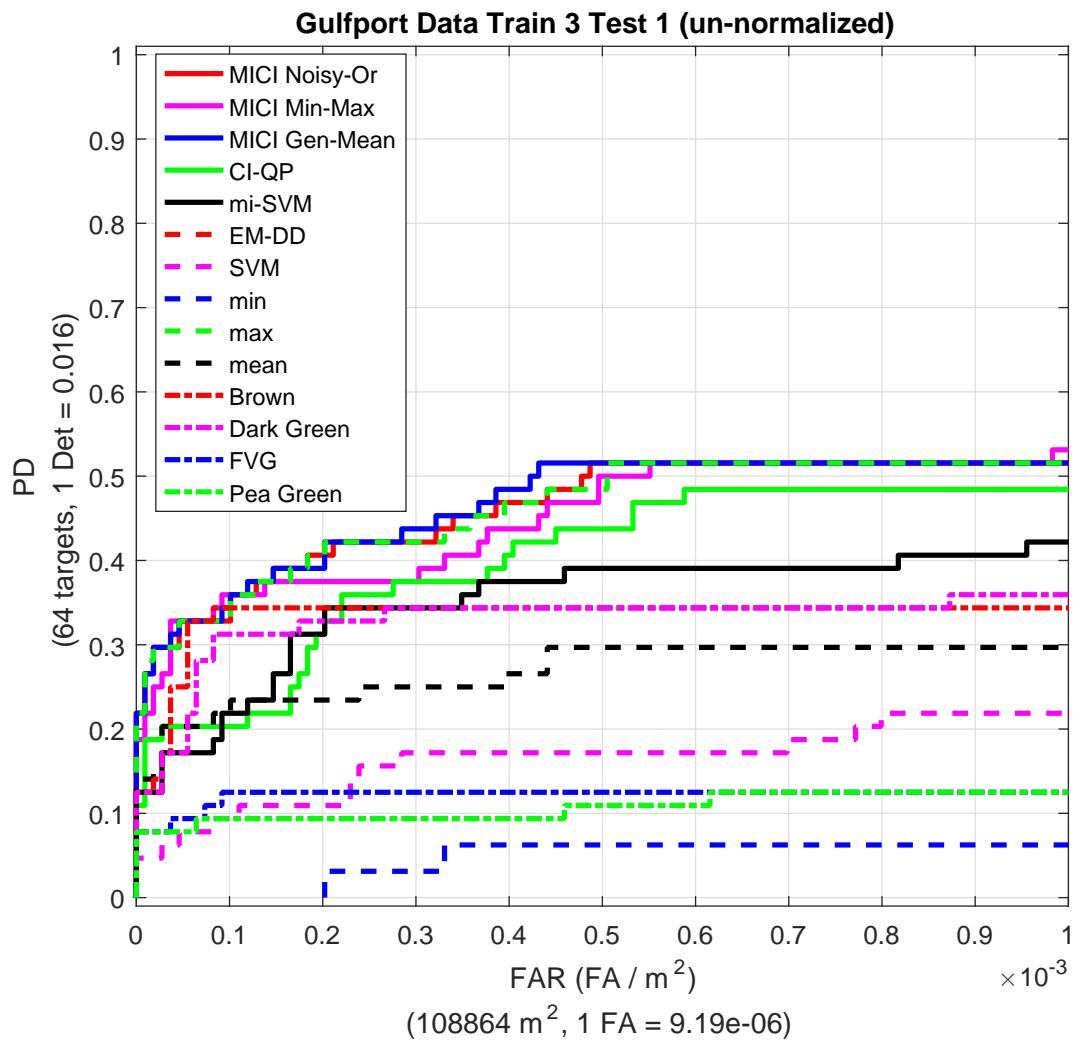


Figure 6.10: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were un-normalized.

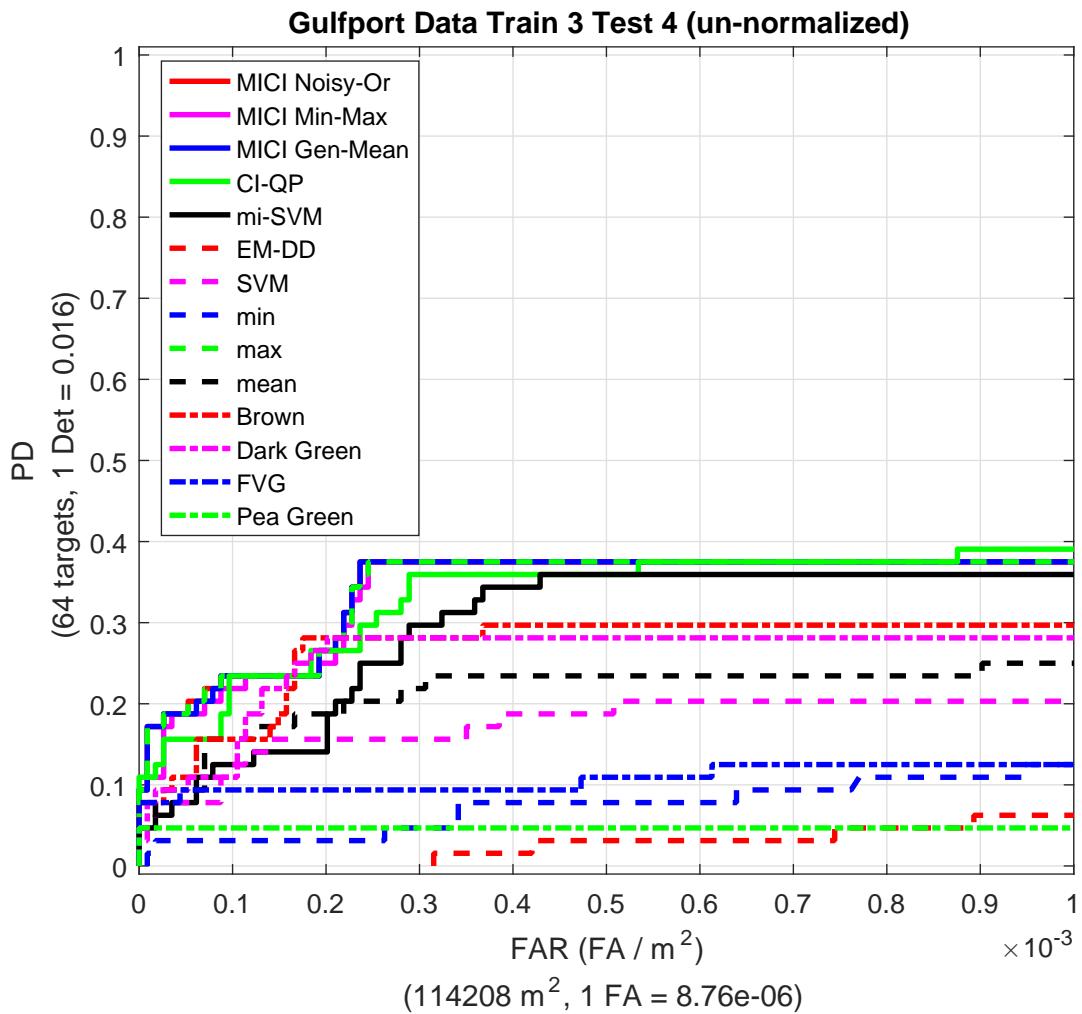


Figure 6.11: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were un-normalized.

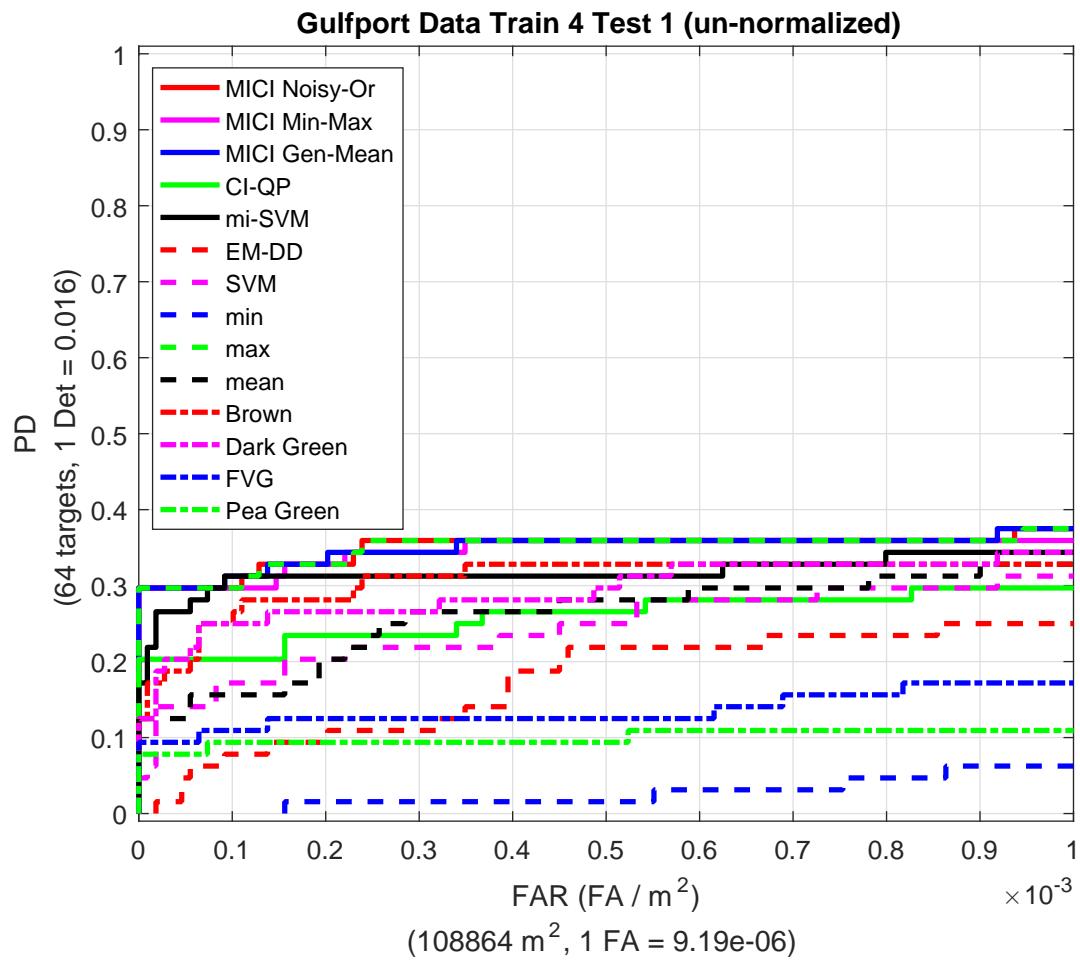


Figure 6.12: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were un-normalized.

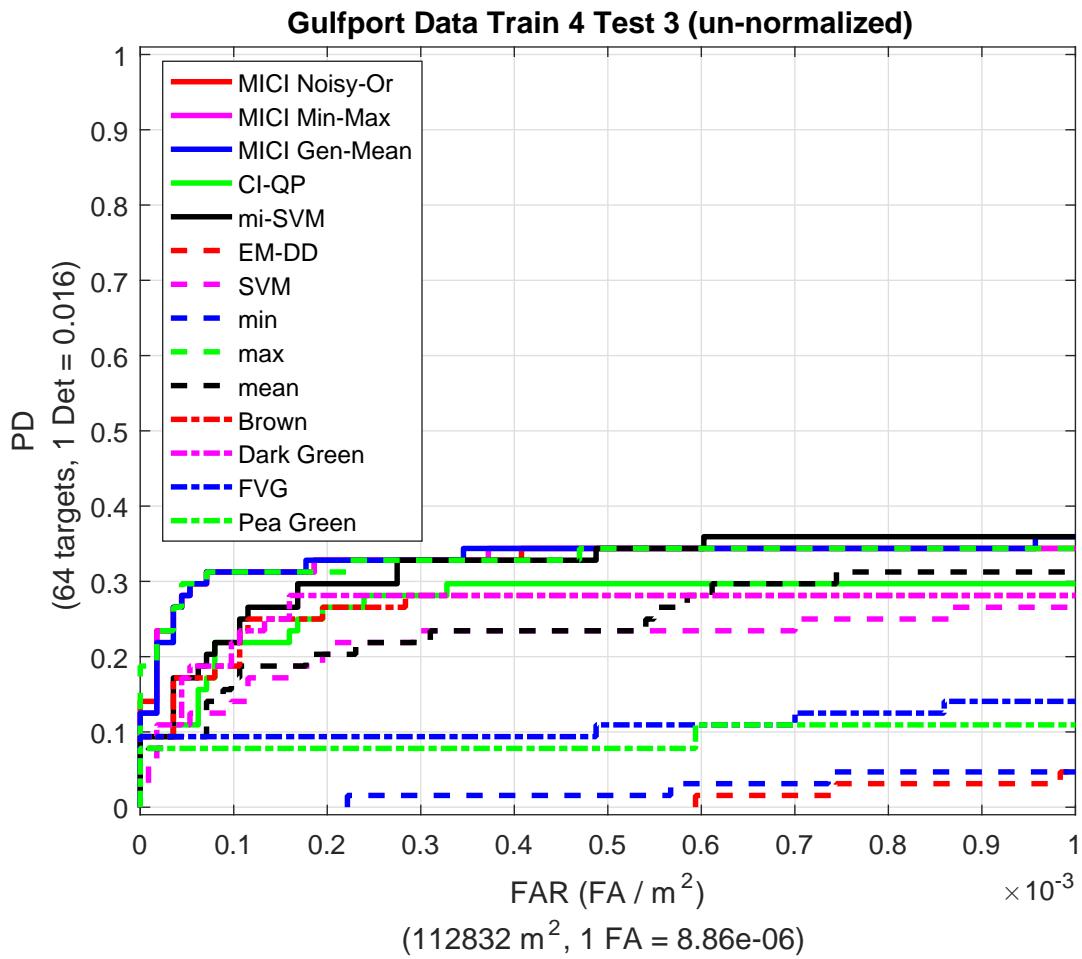


Figure 6.13: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were un-normalized.

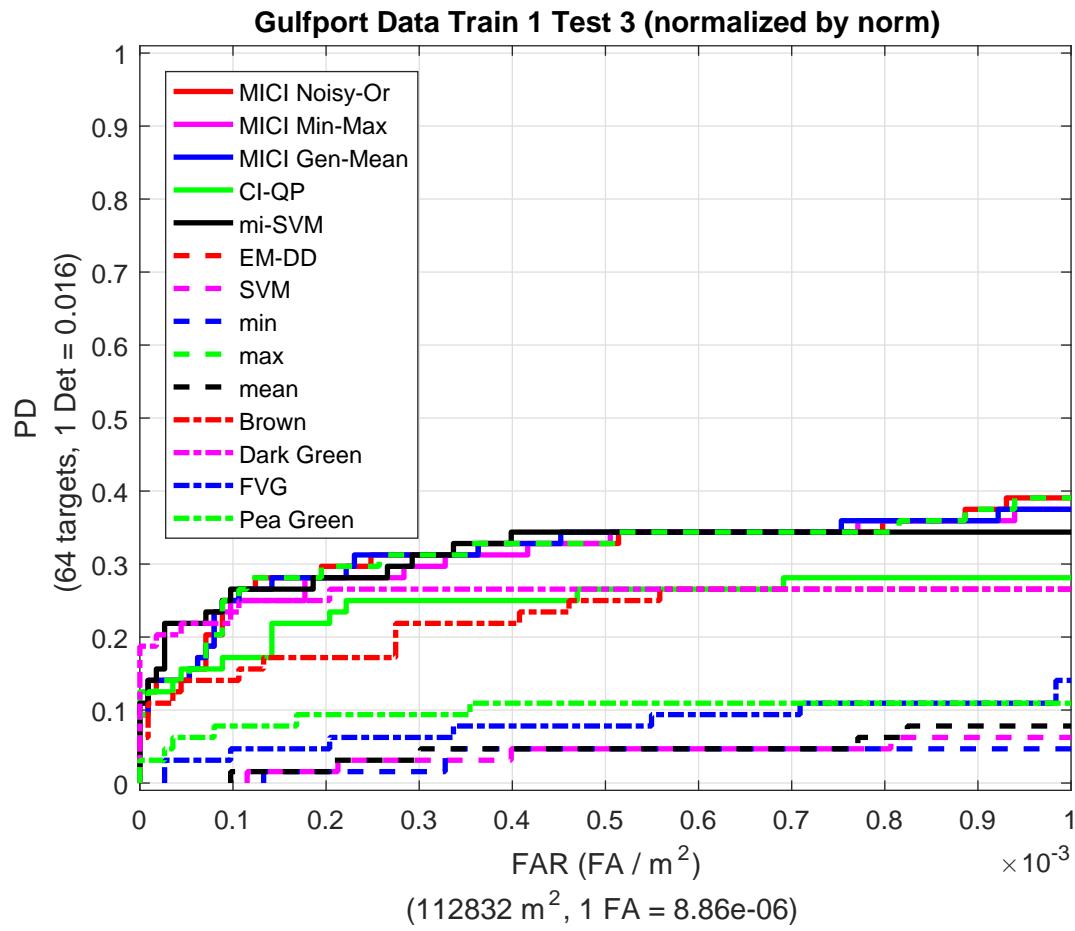


Figure 6.14: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by dividing over the norm of the data.

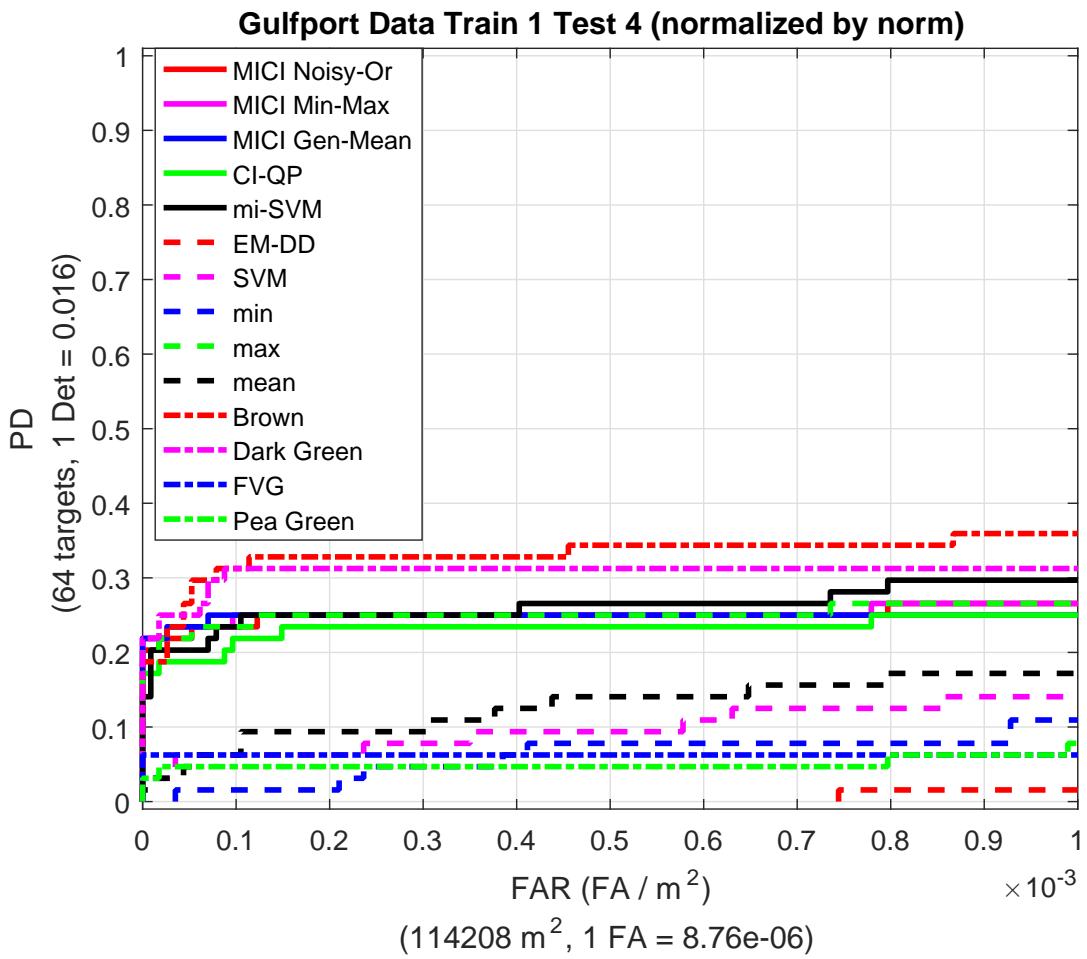


Figure 6.15: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by dividing over the norm of the data.

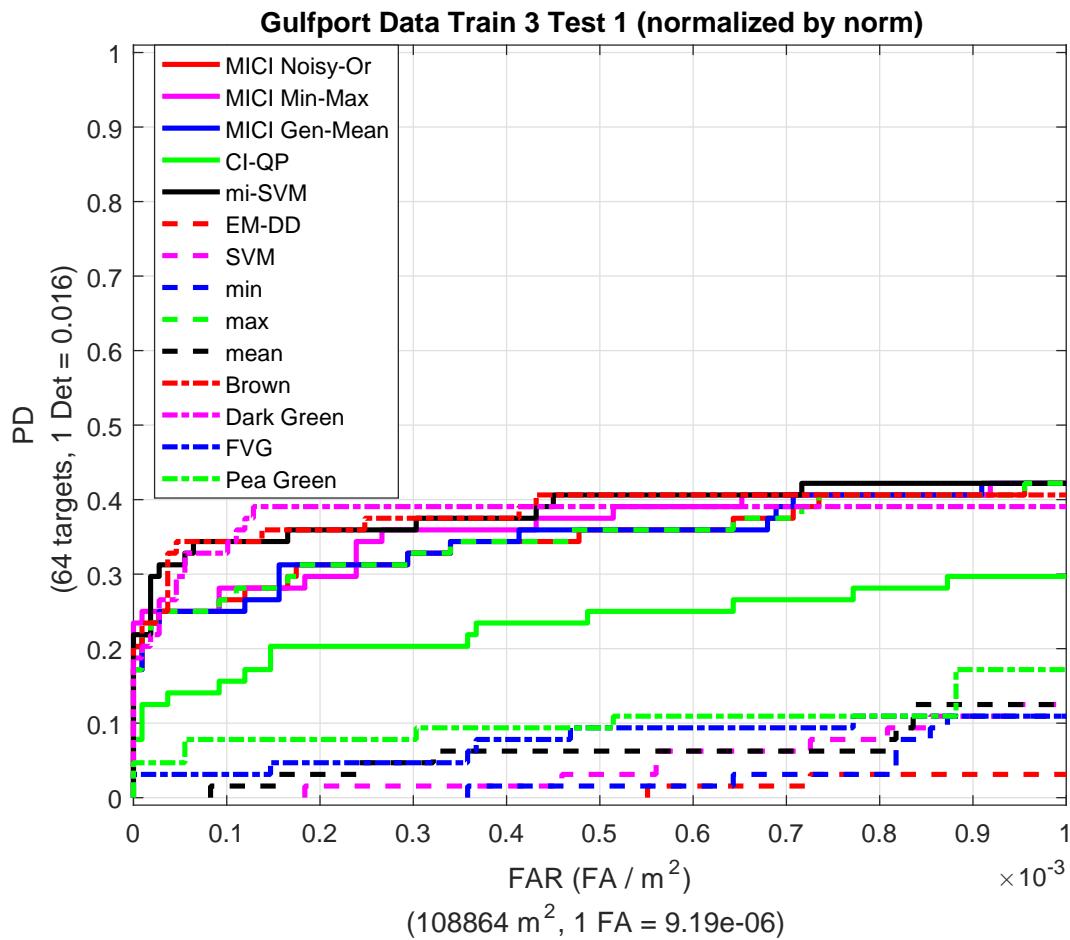


Figure 6.16: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by dividing over the norm of the data.

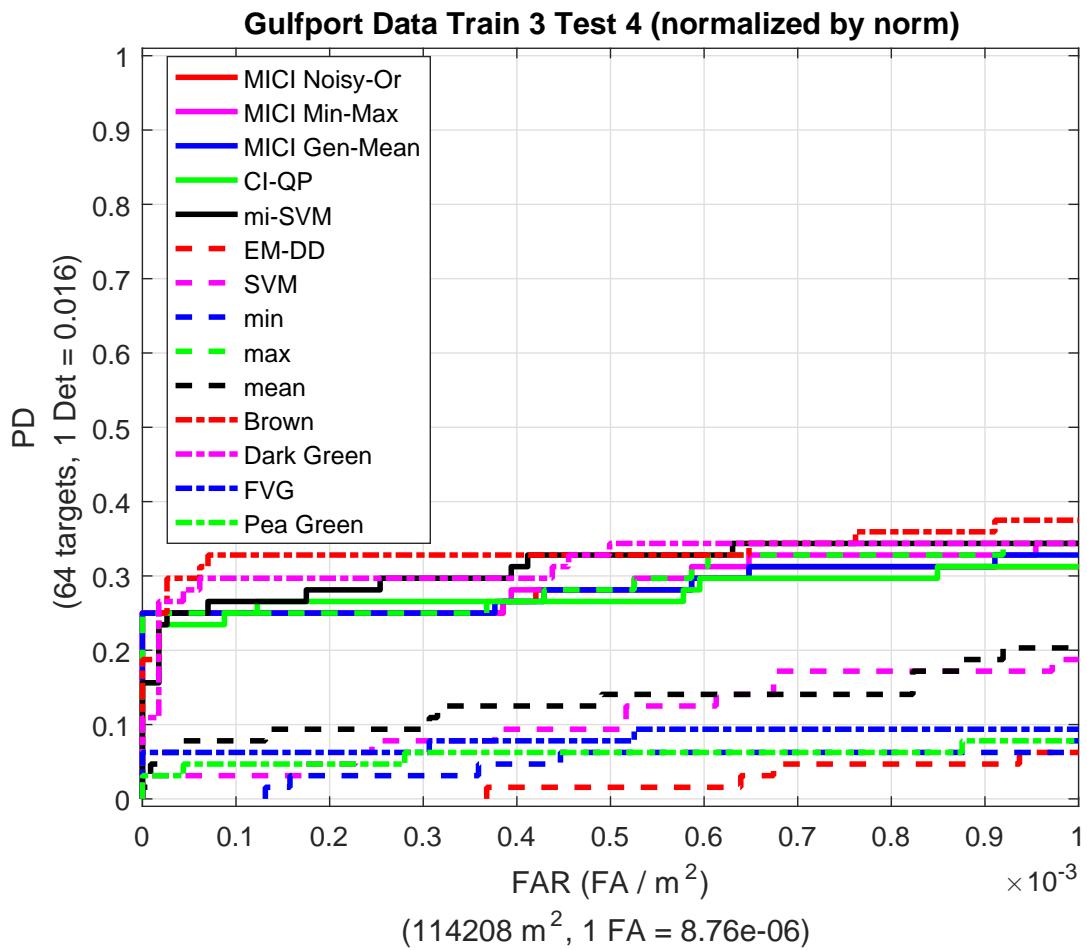


Figure 6.17: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by dividing over the norm of the data.

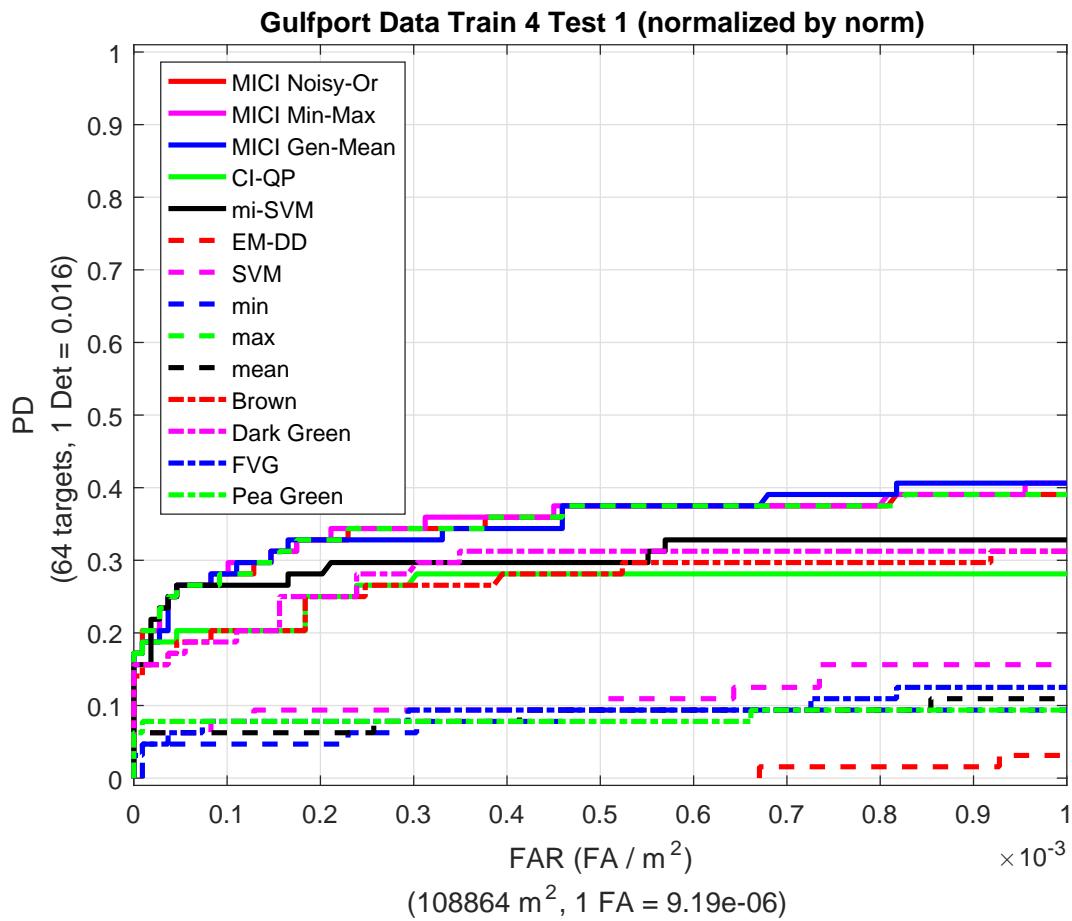


Figure 6.18: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by dividing over the norm of the data.

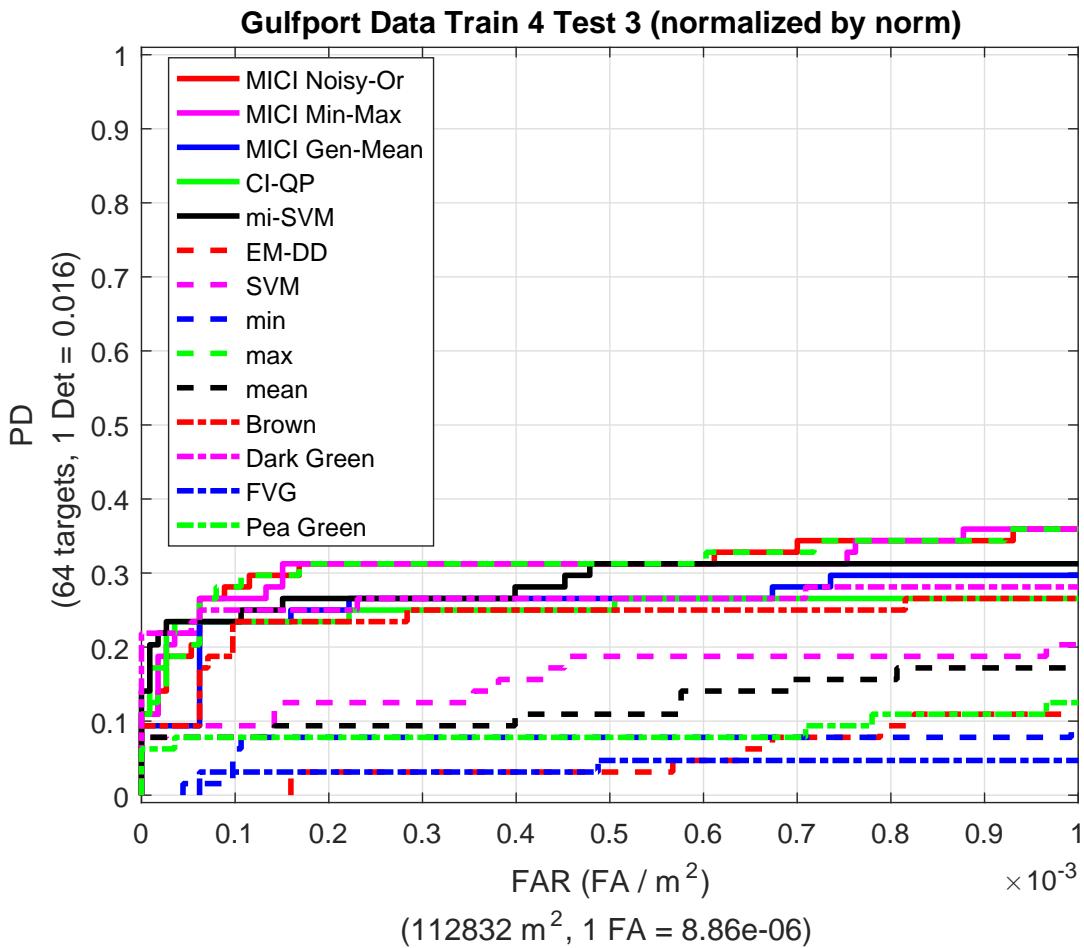


Figure 6.19: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by dividing over the norm of the data.

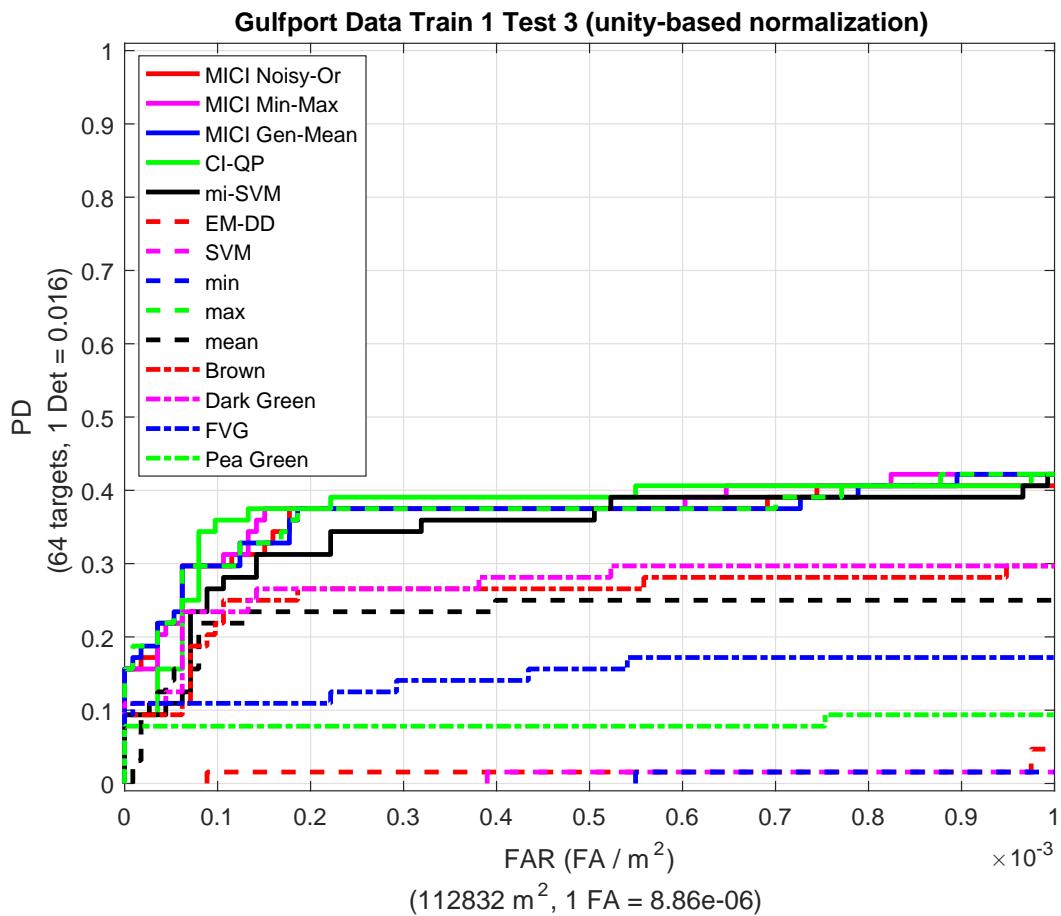


Figure 6.20: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by unity-based normalization.

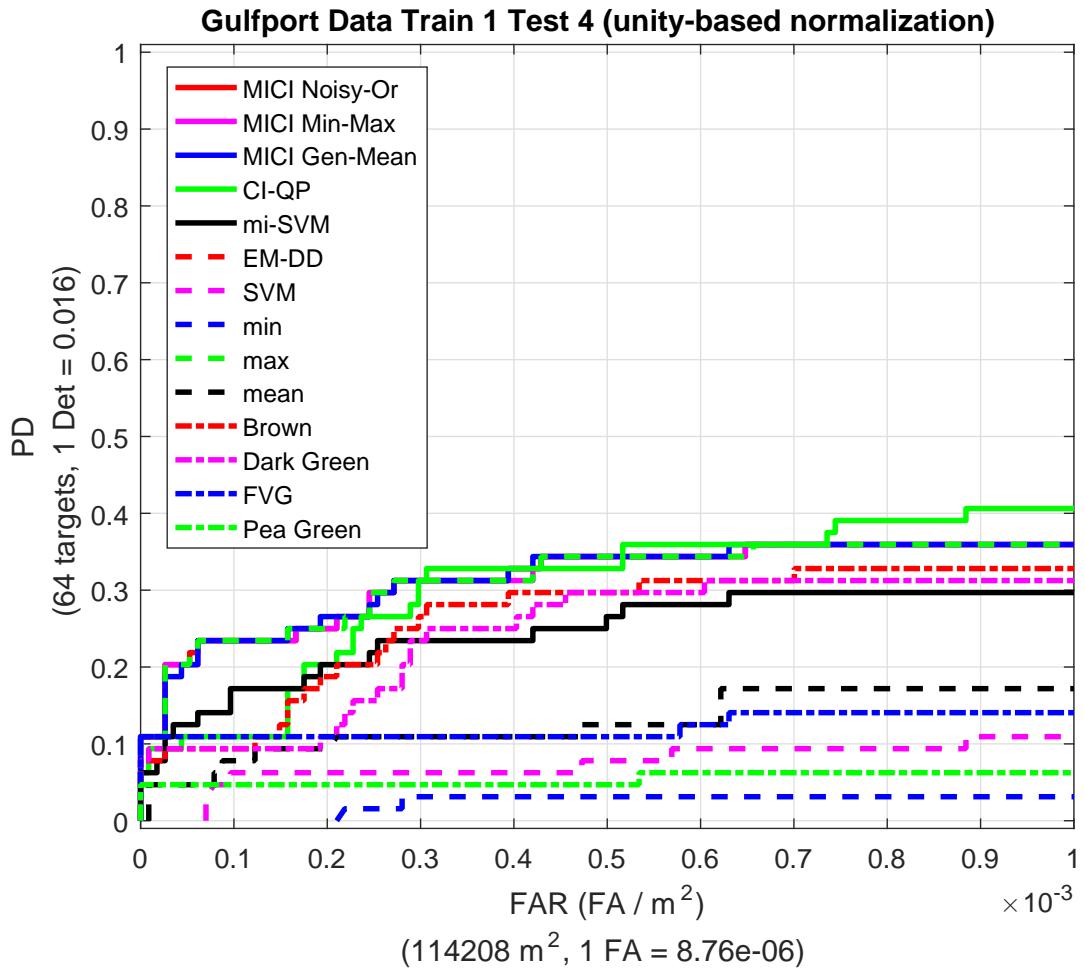


Figure 6.21: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by unity-based normalization.

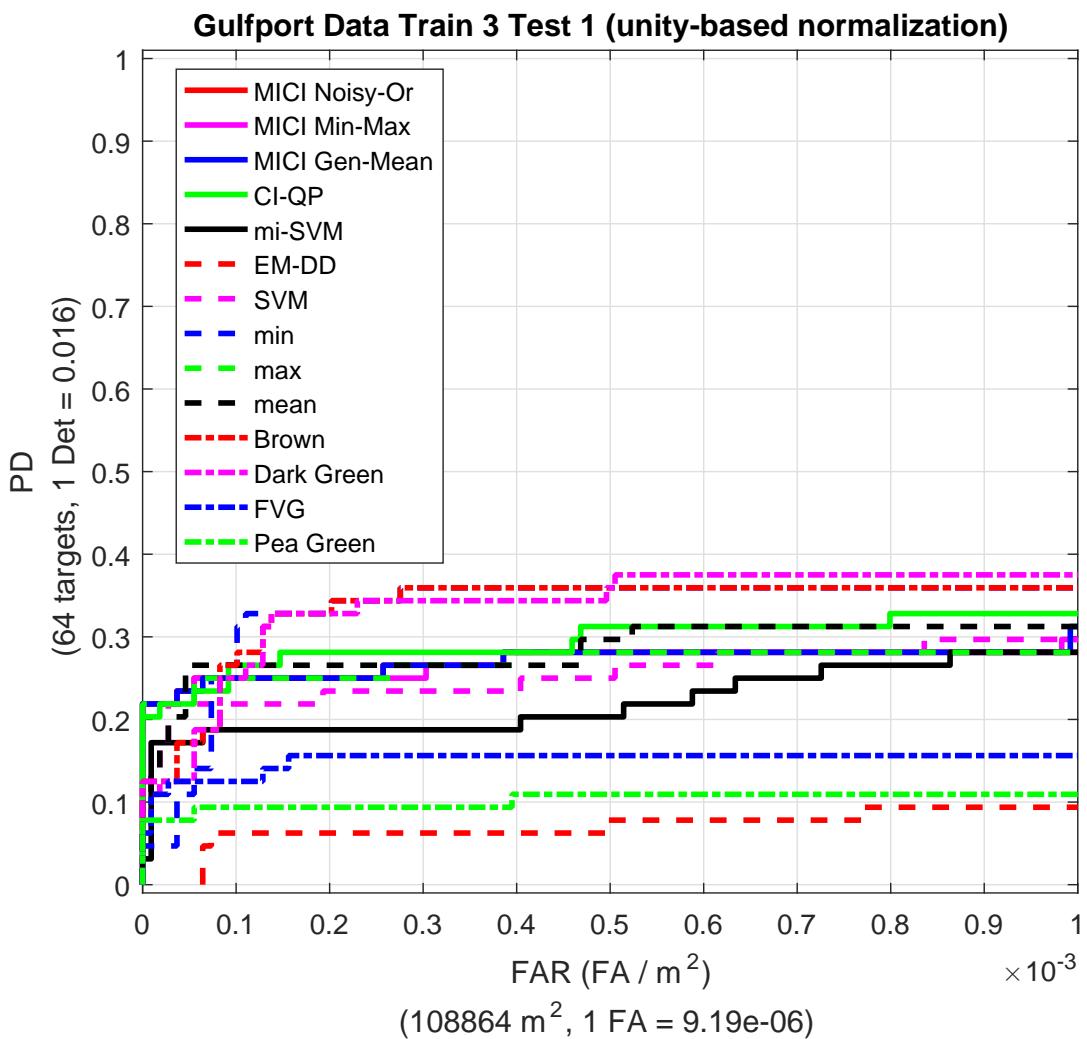


Figure 6.22: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by unity-based normalization.

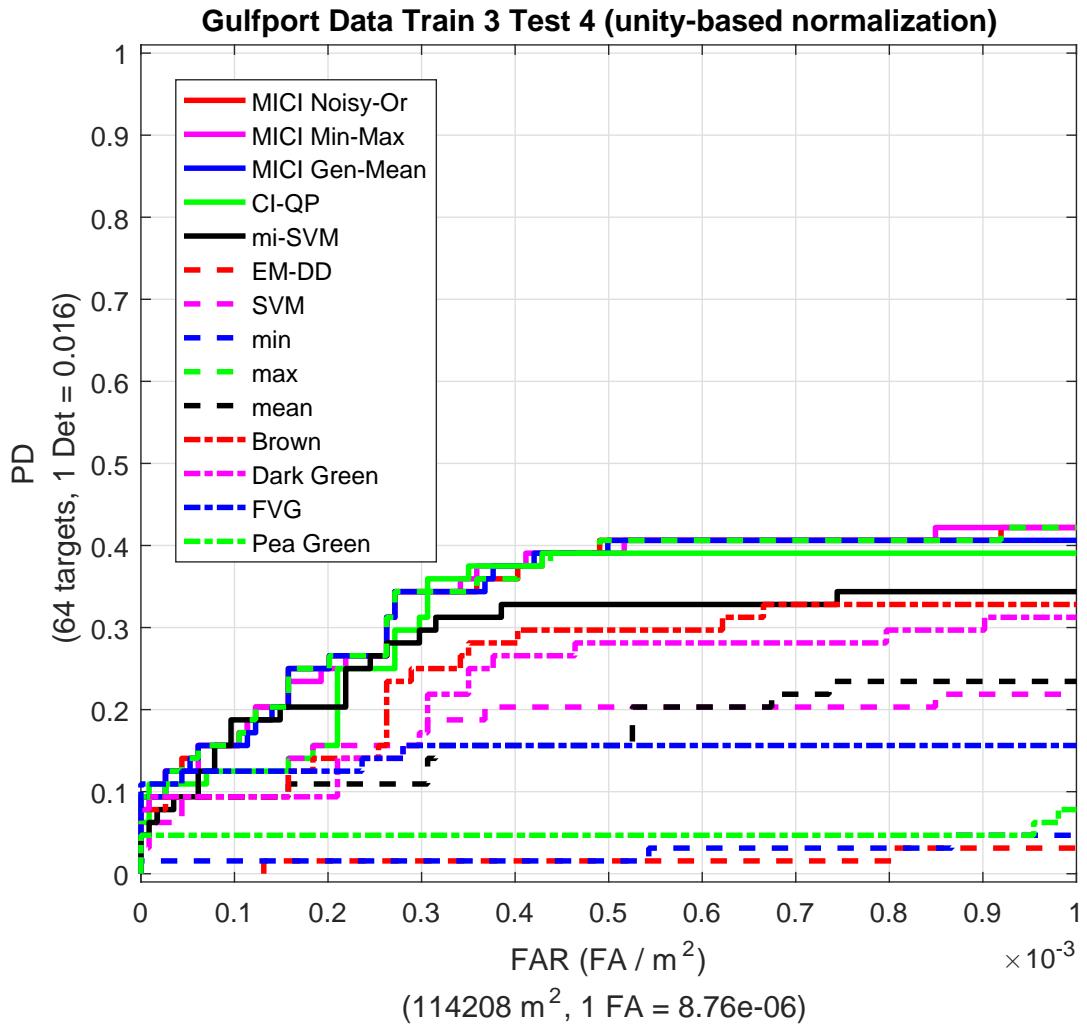


Figure 6.23: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by unity-based normalization.

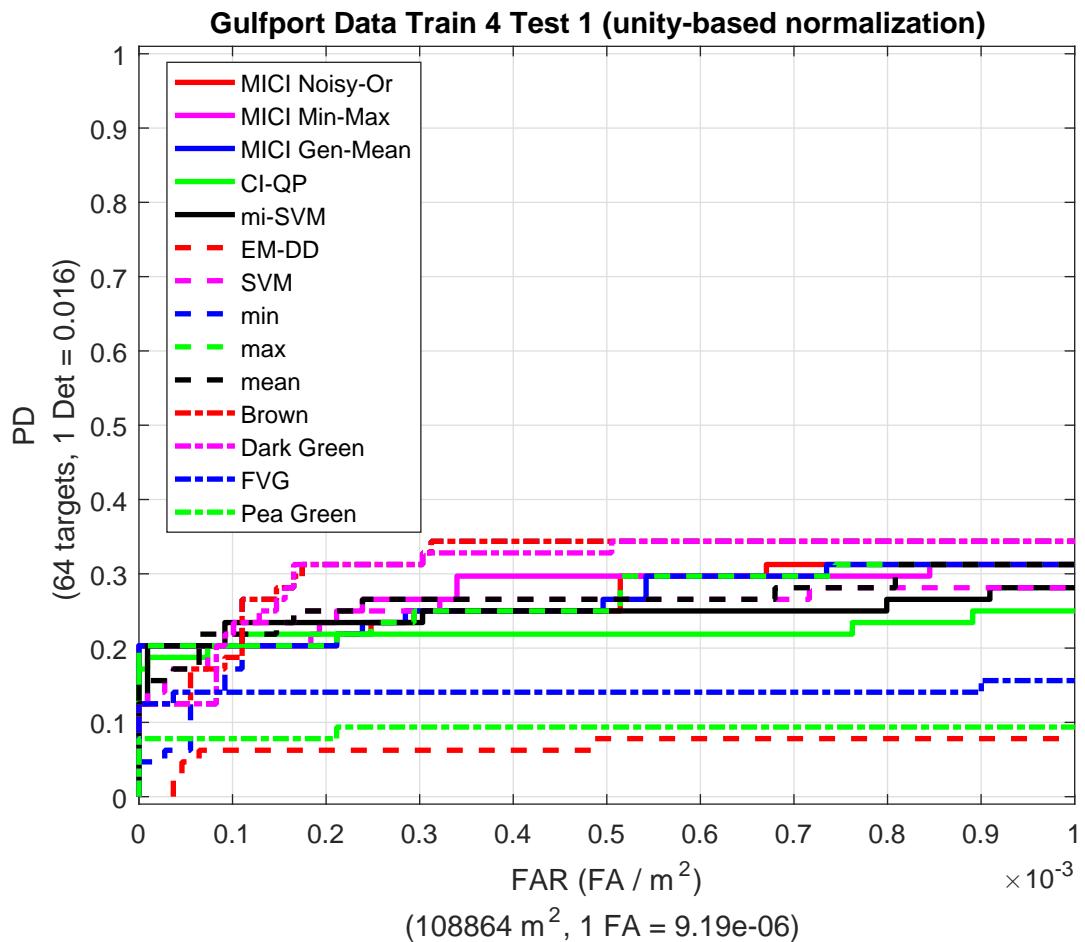


Figure 6.24: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by unity-based normalization.

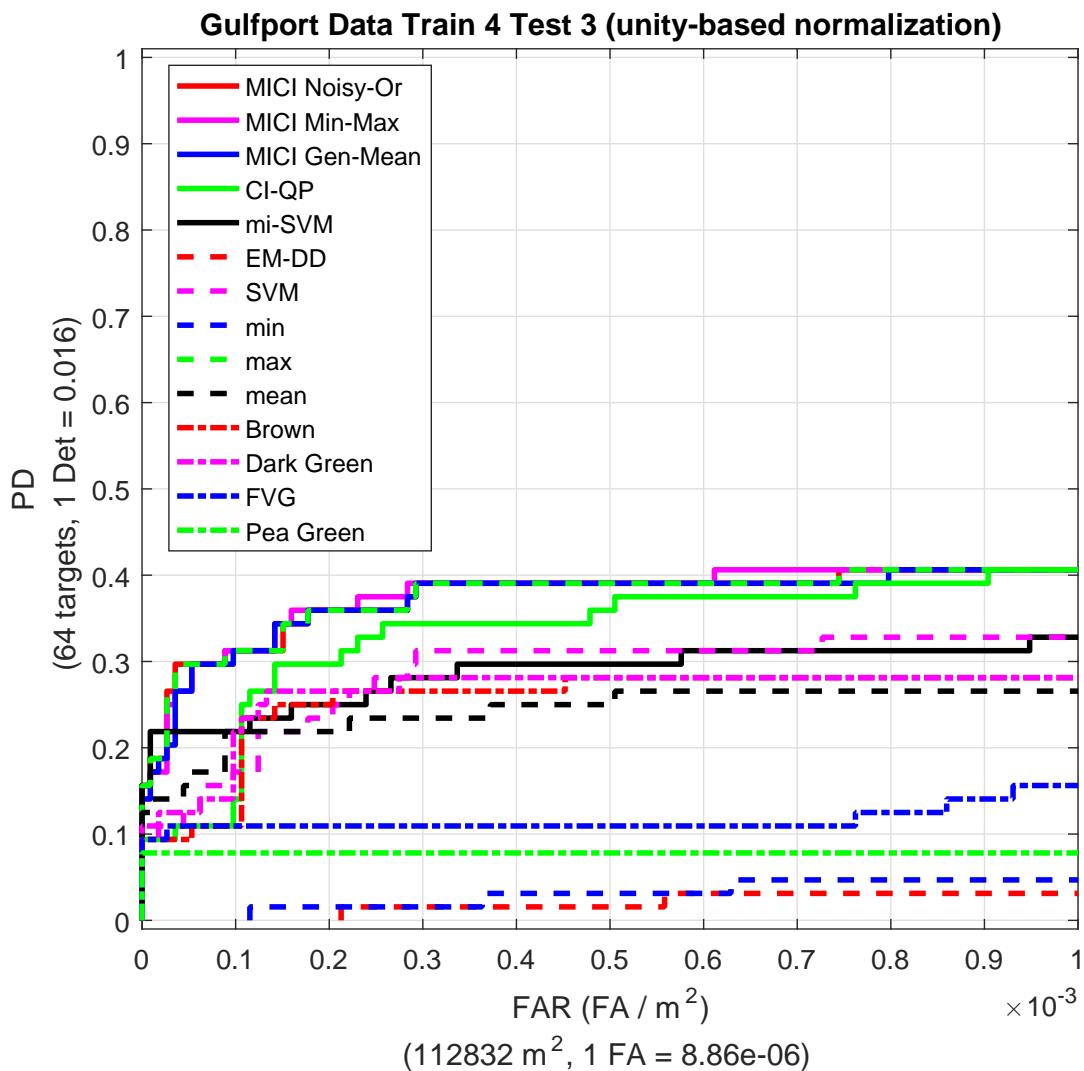


Figure 6.25: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by unity-based normalization.

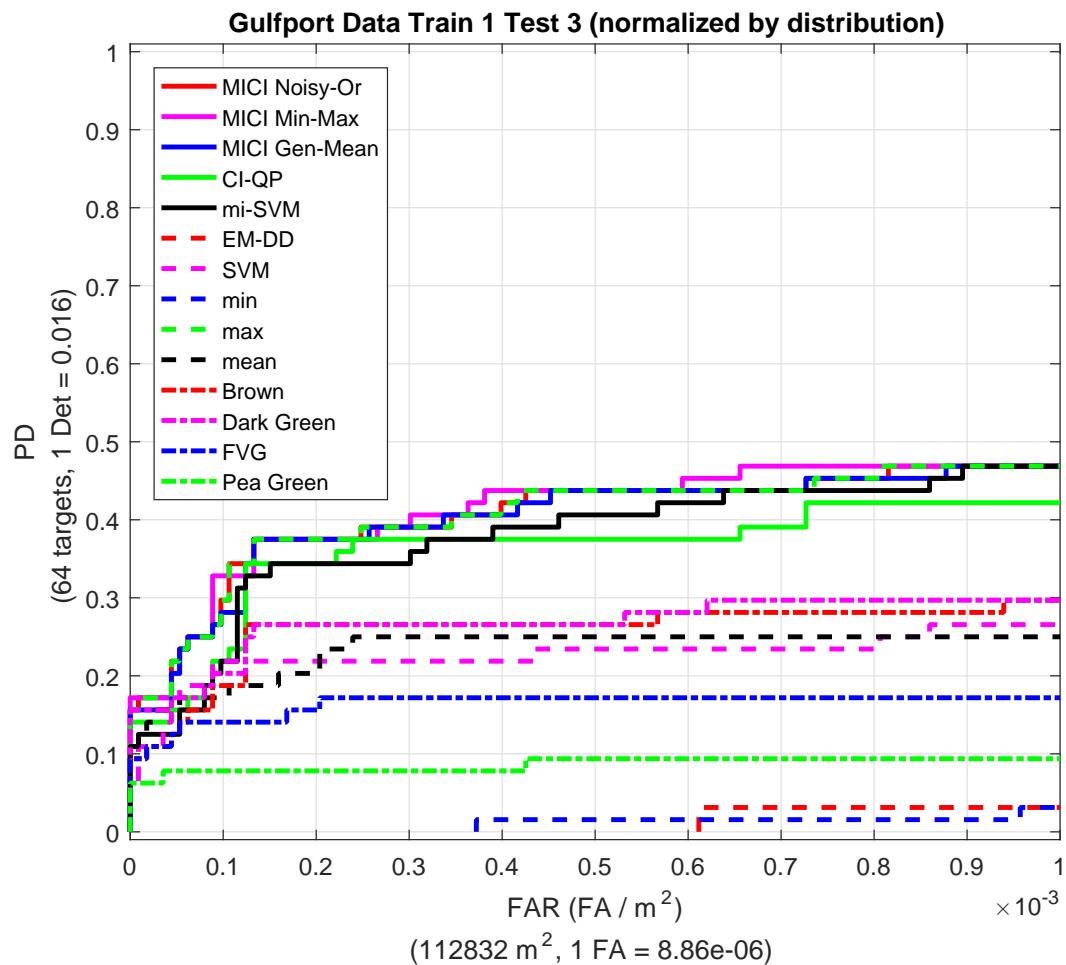


Figure 6.26: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 3. The HSI data were normalized by the mean and standard deviation method.

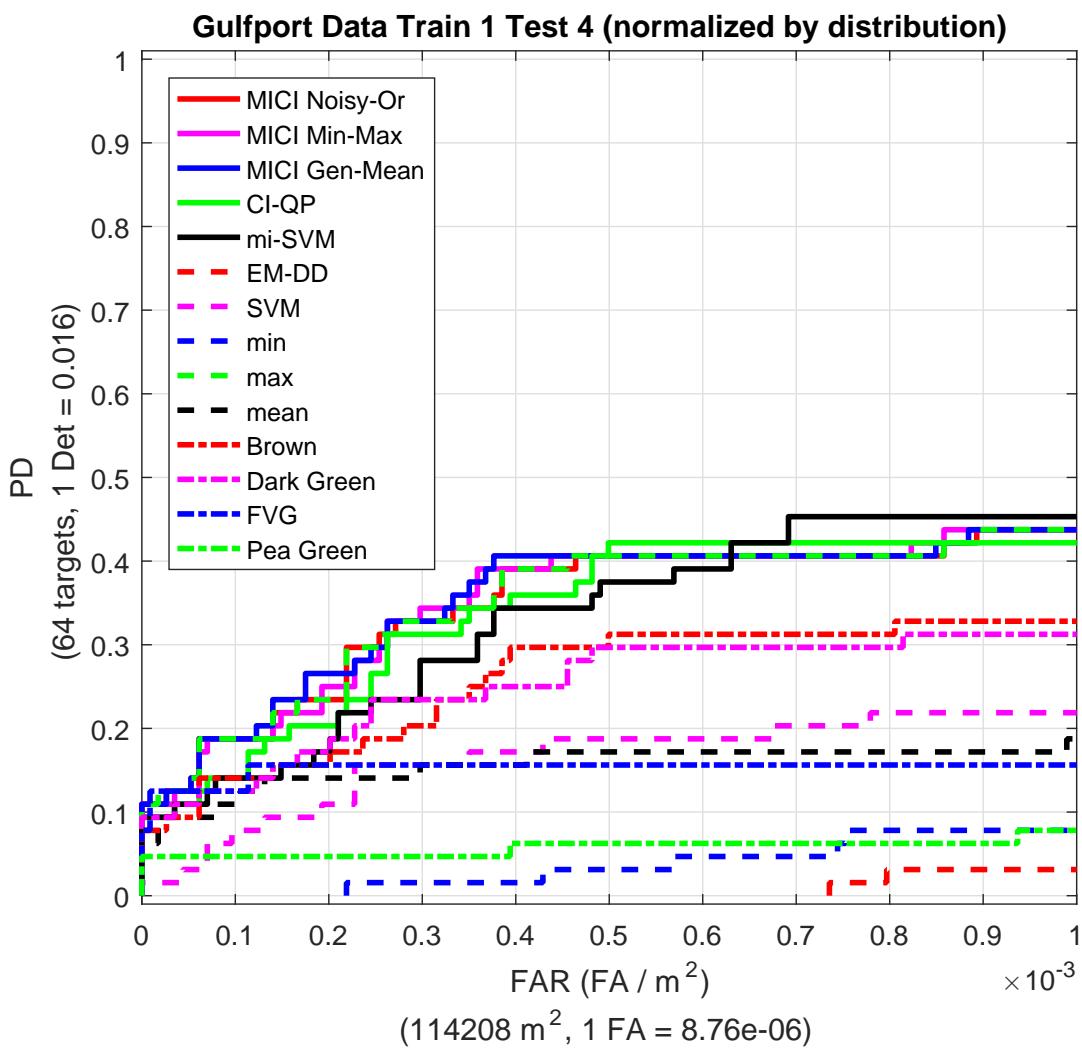


Figure 6.27: ROC curve results for the MUUFL Gulfport data when training on Campus 1 and testing on Campus 4. The HSI data were normalized by the mean and standard deviation method.

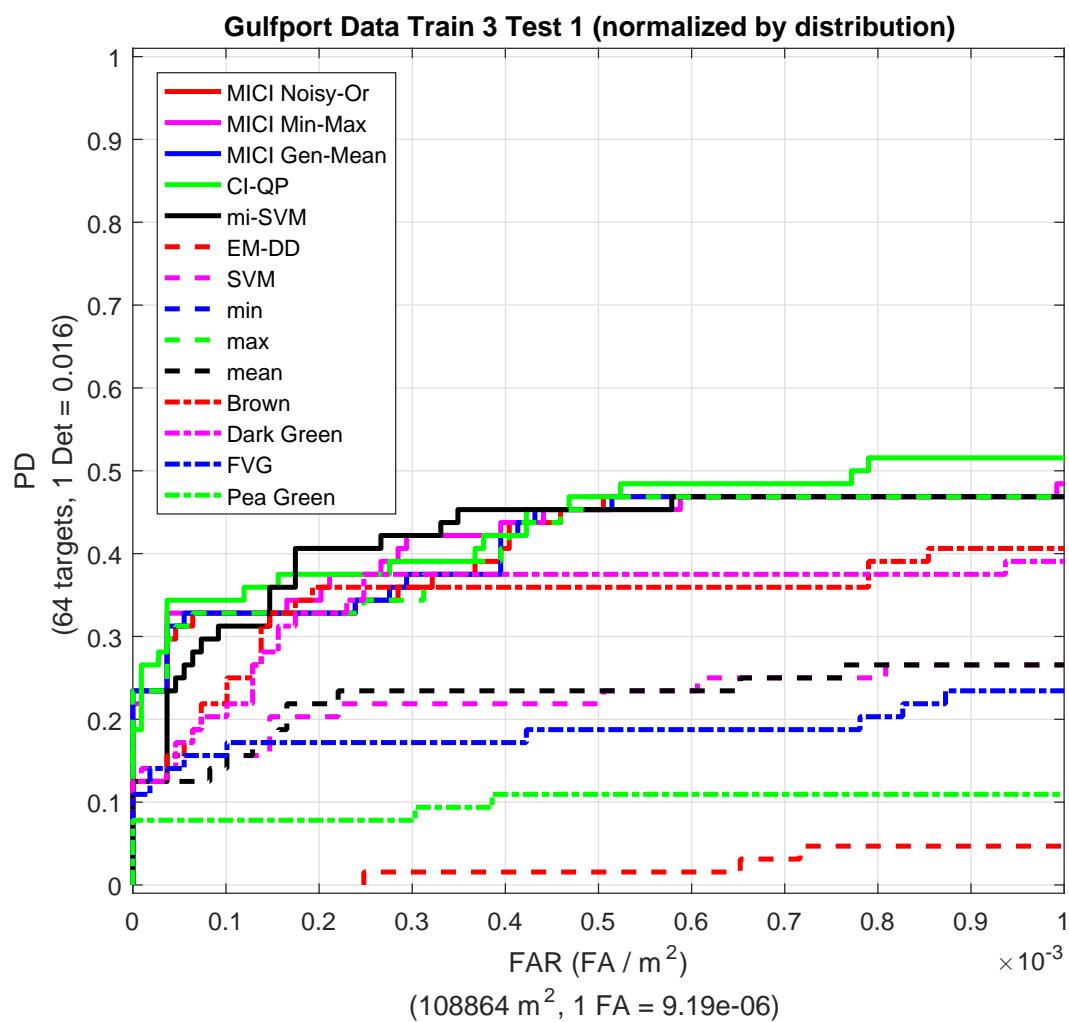


Figure 6.28: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 1. The HSI data were normalized by the mean and standard deviation.

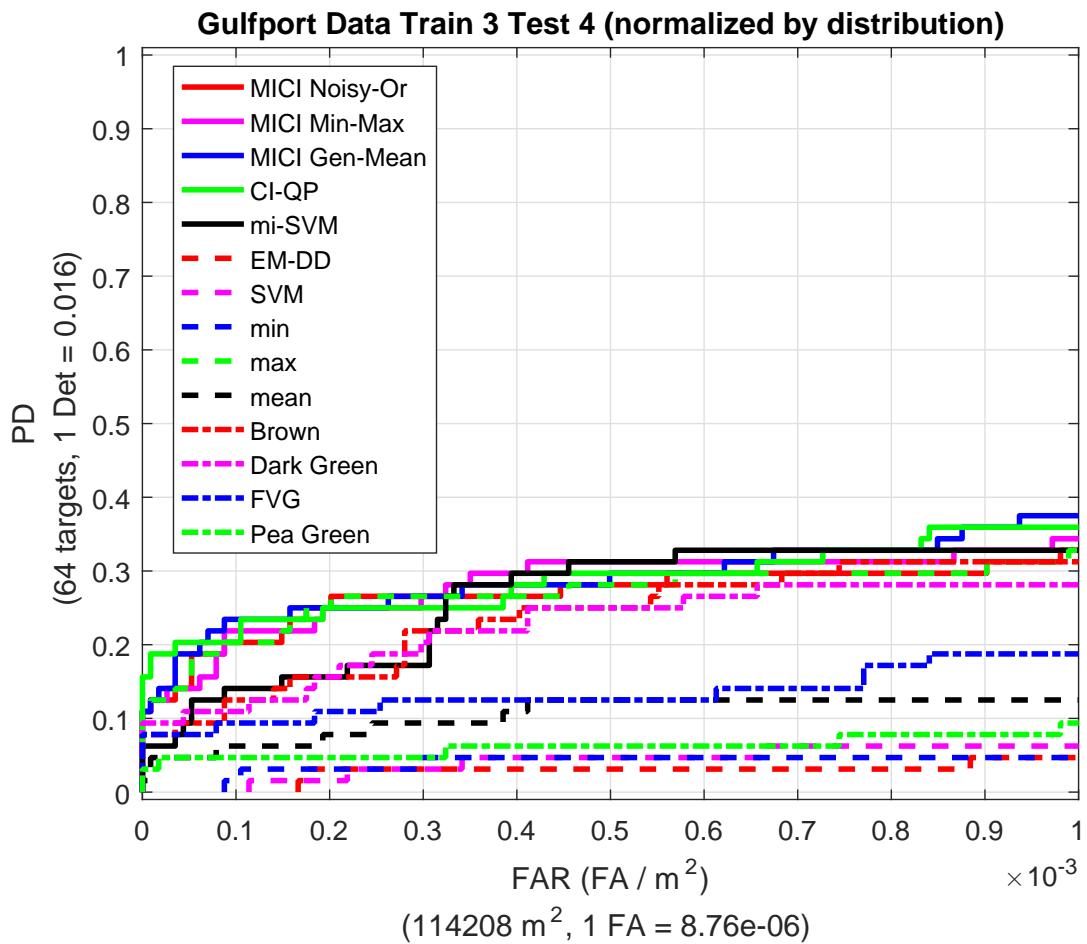


Figure 6.29: ROC curve results for the MUUFL Gulfport data when training on Campus 3 and testing on Campus 4. The HSI data were normalized by the mean and standard deviation.

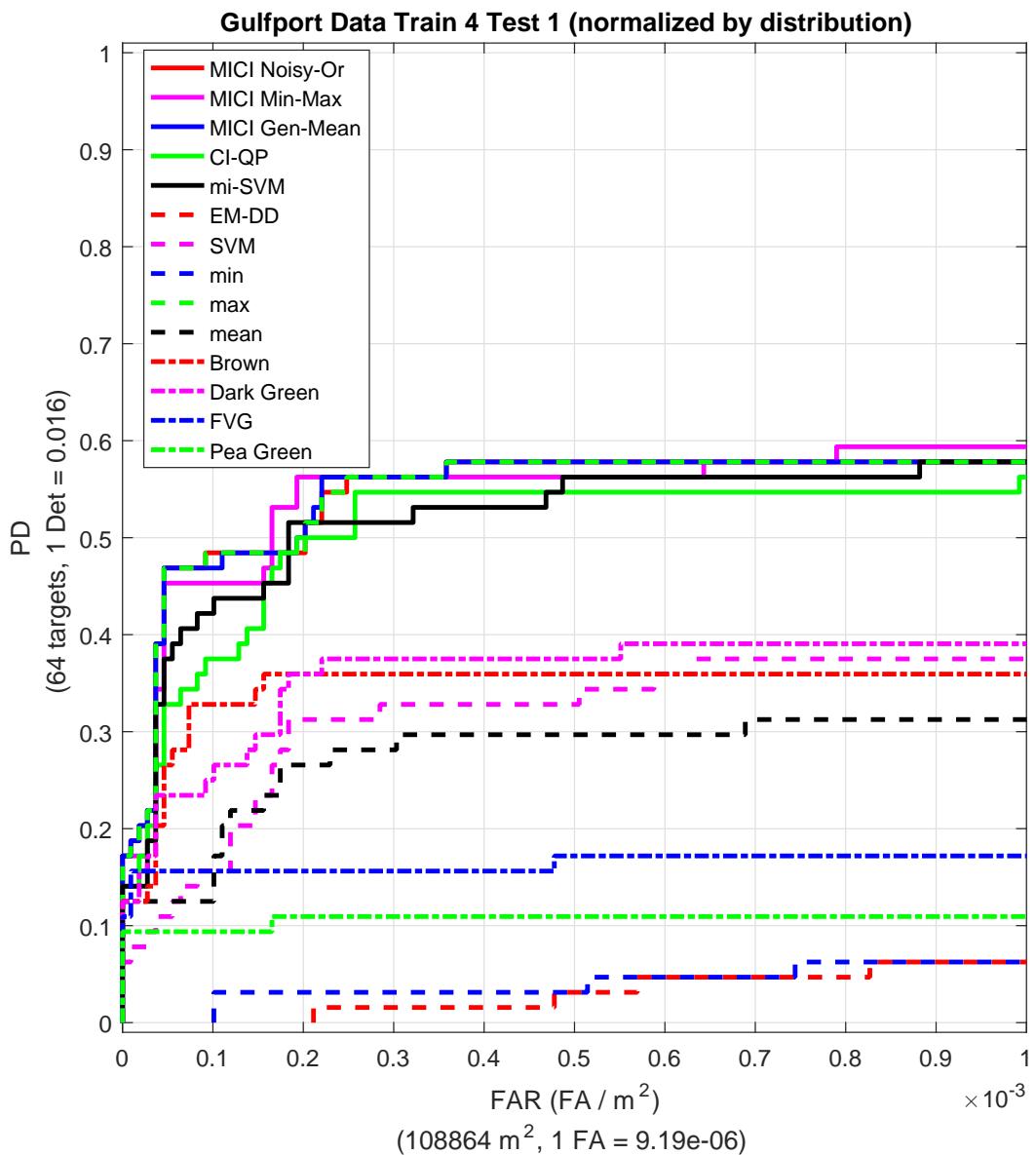


Figure 6.30: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 1. The HSI data were normalized by the mean and standard deviation method.

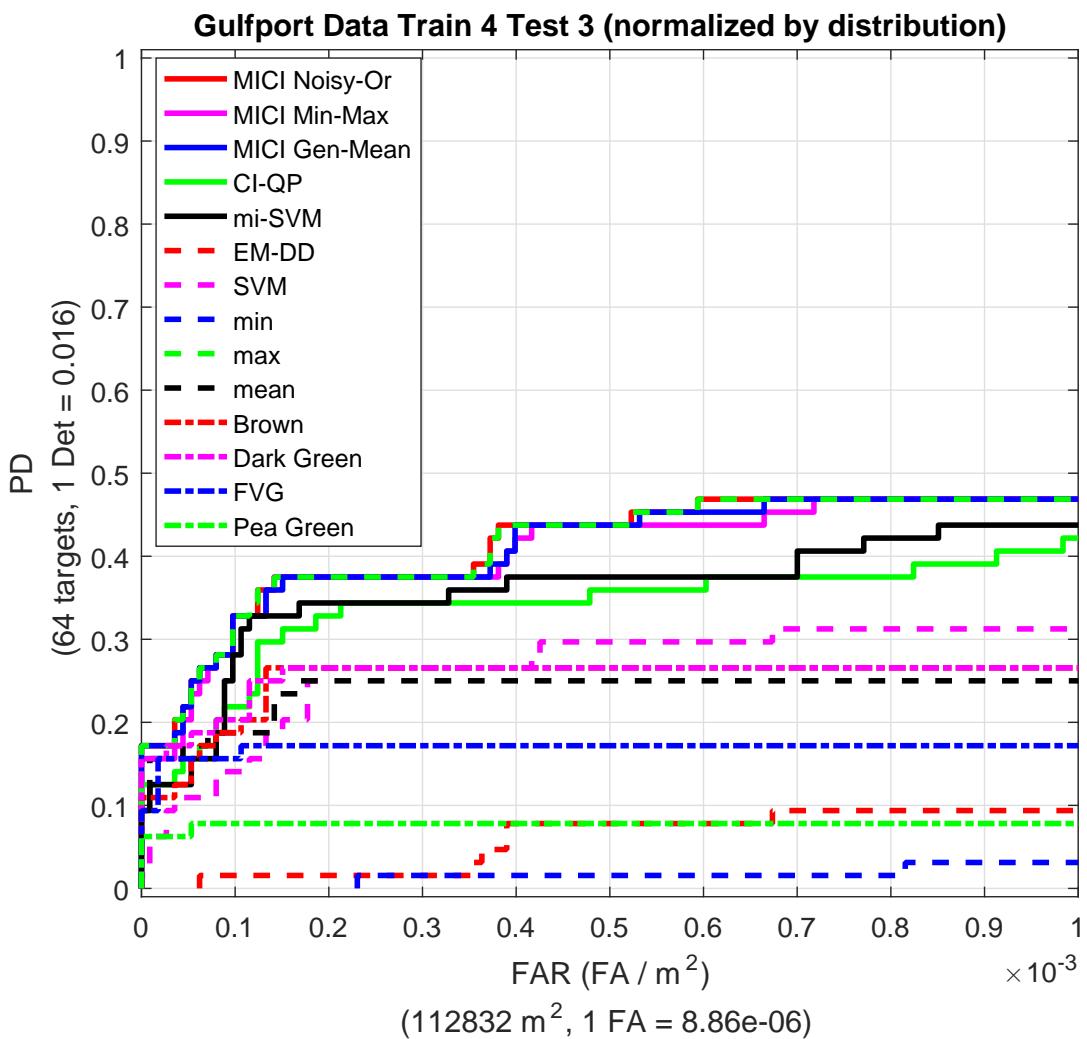


Figure 6.31: ROC curve results for the MUUFL Gulfport data when training on Campus 4 and testing on Campus 3. The HSI data were normalized by the mean and standard deviation method.

6.2 Multi-Resolution Fusion Data Set

6.2.1 Synthetic Multi-Resolution Fusion Data Set

A synthetic 5-source multi-resolution fusion data set is constructed based on the following scenario, similar to the target detection data set in Section 6.1.2, except with consideration of multi-resolution. An “MU” data set is generated with inspiration from the University of Missouri logo. Suppose the two letters are written in the scene, with two different paints for each letter. The goal is to detect both paints (both the “M” and “U” letters). The first two detectors can detect paint type one for the letter “M” and the 3rd and 4th detectors can detect paint type two for the letter “U.” Detector No. 5 can detect background but neither paint. Figure 6.32 shows the ground truth for the “MU” data set.

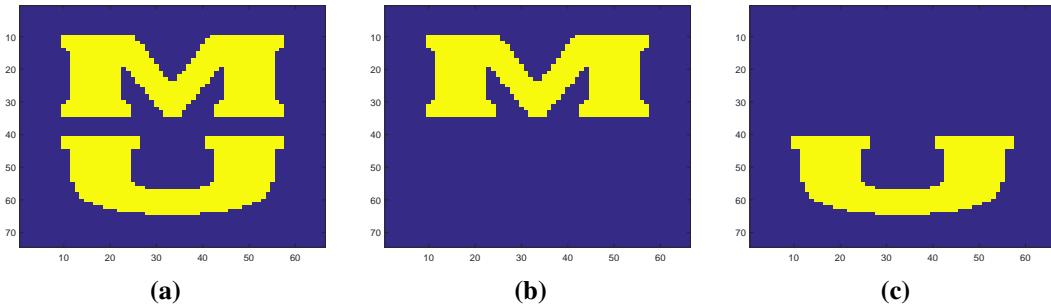


Figure 6.32: Groundtruth for synthetic 5-source dataset for MR-MICI fusion experiments. (a) Full MU data set high resolution ground truth, without noise; (b) Full MU data set with only paint type one (letter “M”-only) in high resolution ground truth, without noise; (c) Full MU data set with only paint type two (letter “U”-only) in high resolution ground truth, without noise. The colors indicate the classification label for each data point according to the colorbar shown in Figure 6.2, bright yellow means label “+1” and deep blue means label “0”.

Suppose five detector results were obtained and MR-MICI was used to perform fusion on the five detector outputs. Suppose these five detector results have different resolutions: Detectors No. 1 and 3 have $1m$ resolution; Detector No. 2 and 4 have $2m$ resolution; and

Detector No. 5 have $4m$ resolution. The size of the five detector maps are of 77×64 , 37×33 , 77×64 , 37×33 , and 25×22 . Figure 6.33 shows one example of the synthetic 5-source multi-resolution “MU” data set. As can be seen from the images, each detector map has clutter (such as the random “i” “z” “o”-shaped letters), in addition to the true targets (“M” and “U” letters). Detector No. 5 has high confidence on the background and not on the targets.

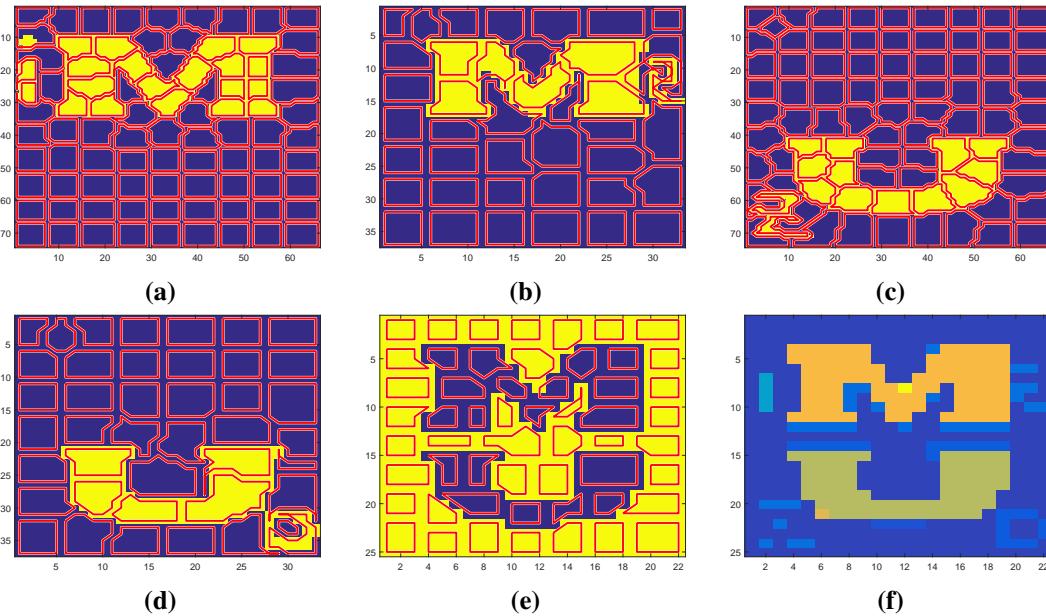


Figure 6.33: One example for synthetic 5-source dataset for MR-MICI fusion experiments. (a) Detector 1, image size 77×64 ; (b) Detector 2, image size 37×33 ; (c) Detector 3, image size 77×64 ; (d) Detector 4, image size 37×33 ; (e) Detector 5, image size 25×22 ; (f) test-on-train map classification result, image size 25×22 . All images are without noise. The colors indicate the detector confidence for each data point according to the colorbar shown in Figure 6.2, bright yellow means “+1” and deep blue means “0”. The red boundaries mark the bags.

The simple linear iterative clustering (SLIC) algorithm [282, 283] is used to generate superpixel as training bags, as marked in red in Figure 6.33. Notice that this is a multi-resolution multiple instance problem as the detector maps have different resolutions and

each pixel in detector map 5 corresponds to multiple pixels in detector maps 1-4. The “correspondence” of pixels across multi-resolution is constructed as follows: Each pixel in each map is assigned to a simulated Easting and Northing coordinate based on their resolutions described above. Based on the coordinates of each pixel in the lowest resolution (in this case Detector No. 5), the pixels in other detector maps within a $1m \times 1m$ window are included as the corresponding training instance for that pixel.

Ideally, to fulfill the goal of detecting true targets (both “M” and “U” letters), the fusion will put emphasis on the intersection of Detectors 1 and 2 and Detectors 3 and 4 and take the union of Detectors 1-4, while ignoring Detector 5 (background). Table 6.12 shows one example of the estimated measure value learned. The red highlights g_{12} and g_{34} , which has high measure values as expected. The blue highlights g_5 on Detector 5 alone, which has low measure values as expected. Figure 6.33f show one example of test-on-train map classification result on the lowest resolution image (size 25×22). As can be seen, the clutter are much weakened after fusion and the target letters “M” and “U” are assigned high confidence, as desired.

Table 6.12: One example estimated measure element values learned for synthetic 5-source multi-resolution classification data set after one run.

	g_1 0.0720	g_2 0.2411	g_3 0.0835	g_4 0.0010	g_5 0.0097				
g_{12} 0.8403	g_{13} 0.3775	g_{14} 0.2624	g_{15} 0.0937	g_{23} 0.3288	g_{24} 0.4145	g_{25} 0.2602	g_{34} 0.7830	g_{35} 0.2293	g_{45} 0.1572
g_{123} 0.8497	g_{124} 0.8927	g_{125} 0.9428	g_{134} 0.9015	g_{135} 0.6035	g_{145} 0.4427	g_{234} 0.8876	g_{235} 0.6740	g_{245} 0.7572	g_{345} 0.8985
	g_{1234} 0.9414	g_{1235} 0.9991	g_{1245} 0.9790	g_{1345} 0.9217	g_{2345} 0.9504				

6.2.2 MUUFL Gulfport Building Detection – Sub-image

Three subimages were chosen from the MUUFL Gulfport data set [275] as the fusion data sets. The MR-MICI is used to perform fusion based on the hyperspectral imagery and raw LiDAR point cloud data.

Suppose building is the target (positive) class and non-building has non-target (negative) labels. Figure 6.34 shows three sub-images extracted from the scene. The simple linear iterative clustering (SLIC) algorithm [282, 283] is used to generate superpixel as training bags, as marked in red in Figure 6.34d to 6.34f. In this experiment, the bags were manually labeled to be positive if it contains buildings and negative if not. Figure 6.35 shows one result on building classification on sub-image 1 (test-on-train) after using MR-MICI fusion, compared with MICI (based on rasterized data) and CI-QP (based on rasterized data). The MR-MICI-sampling refers to the method of picking points by sampling from a multinomial based on the CI values, instead of based on max or min as described in Chapter 5. If the bag-level labels were known, the test map was constructed by picking the combination of sources that gives the max CI value for positive bags and min CI value for negative bags. If not knowing bag-level labels, we randomly sample a combination to form the test map (marked as “multiresMinMaxR” and “multires-sampleMinMaxR” in the plot legend). Figure 6.36 shows the receiver operating characteristic (ROC) curve. The ROC curve results show higher classification accuracy of building using MR-MICI-sampling compared with MICI and CI-QP results, especially at lower false alarm rate (FAR). Similarly, Figure 6.37 and Figure 6.38 show the classification map and ROC curve results for training on sub-image 1 and testing on sub-image 3 after using MR-MICI fusion, compared with MICI (based on rasterized data) and CI-QP (based on rasterized data). The performance when knowing the bag-level labels is higher than randomly sampling, which

makes sense as the bag-level labels provide information to help pick the correct point in testing.

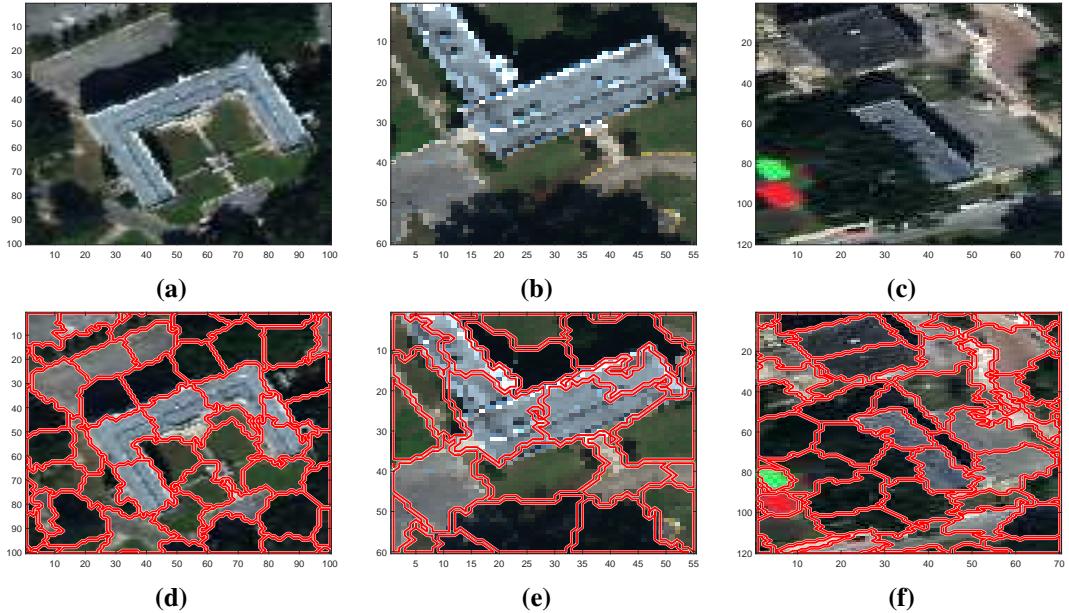


Figure 6.34: Three subimages of buildings in MUUFL Gulfport campus 1 data set. (a) sub-image 1 of size 100×100 , with $Y = [151 : 250]$, $X = [16 : 115]$; (b) sub-image 2 of size 60×55 , with $Y = [91 : 150]$, $X = [56 : 110]$; (c) sub-image 3 of size 120×70 , with $Y = [71 : 200]$, $X = [156 : 225]$; (d) sub-image 1 after SLIC superpixel bag generation; (e) sub-image 2 after SLIC superpixel bag generation; (f) sub-image 3 after SLIC superpixel bag generation. Red boundaries marks training bags. The Y and X numbers refers to the row and column pixel indices of the sub-images from the original campus 1 full image.

6.2.3 MUUFL Gulfport Scene Understanding: Building, Sidewalk and Road

Section 6.2.2 provides preliminary results on small sub-image patches of the Gulfport data. In this section, the proposed fusion methods were run on the entire MUUFL Gulfport data set for building, sidewalk and road classification.

The MUUFL Gulfport hyperspectral imagery and LiDAR data set [275] used in this experiment contains two flights of hyperspectral and lidar data. The LiDAR raw data was available in point cloud and was collected by the Gemini LiDAR sensor at 35000ft over the scene in November 2011 [275]. Figure 6.39 shows the four sets of LiDAR data collected over the area. The red pin shows the location of the campus. As can be seen from Figure 6.39, LiDAR flights 001 and 002 cover the campus area. Therefore, LiDAR data from these two lines were used for fusion. The scatter plot of the raw LiDAR point cloud data over the entire Gulfpark campus is shown in Figure 6.41.

The RGB images of campus 1 and campus 2 data are shown in Figure 6.40a and Figure 6.40b. Detailed description of MUUFL Gulfport hyperspectral imagery (HSI) can be seen in Section 6.1.4. The “campus 1” HSI imagery used in this experiment is the same hyperspectral image cube as in “muufl_gulfport_campus_w_lidar_1.mat” in Section 6.1.4 and hereinafter denoted as “campus 1” data. The same goes for “campus 2”. The first 220 columns of the hyperspectral data were kept due to highly bright beach sand materials in the lower right corner in the original images and the first and last four bands of the data were removed due to noise. The size of the HSI image is $325 \times 220 \times 64$ for both flights in this experiment. The data set also includes rasterized LiDAR imagery processed by 3001 Inc. and Optech Inc. The rasterized LiDAR imagery for campus 1 and campus 2 are shown in Figure 6.42a and Figure 6.42b. The proposed MR-MICI was used to perform detection on building, sidewalk, and road classes based on the fusion of the hyperspectral imagery and raw LiDAR point cloud data.

The simple linear iterative clustering (SLIC) algorithm [282, 283] was used to generate superpixel as training bags. Bag labels were constructed based on the open map data

from Open Street Map (OSM)³. Information from Google Earth⁴, Google Maps⁵ and geo-tagged photographs from a digital camera taken at the scene are also used as auxiliary data. Figure 6.43 shows the map extracted from Open Street Map (OSM) based on the tags available, such as “highway”, “footway”, “building”, “parking” etc. The blue lines corresponds to asphalt materials, which includes road, highway and parking lot. The magenta lines corresponds to sidewalk/footway. The green lines marks buildings. The black lines corresponds to “other” tags.

Affine transformation was used to merge information from the OSM data and the HSI data coordinates. Two pairs of points were selected on both the OSM map (with longitude and latitude values) and the HSI data (with pixel coordinates) and the affine transformation is used to map the points from OSM map to the HSI imagery. Then, for all the bags formed by SLIC segmentation, if a point tagged by OSM as positive falls into a bag after the affine transformation, the bag is labeled positive, and a bag is labeled negative if no positive points are in the bag.

MUUFL Gulfport Building Detection

Figure 6.44b shows the SLIC segmentation result on the MUUFL Gulfport hyperspectral campus 1 data. Each segmentation is treated as a “bag” in training, where the red marks the positive bags that contain building pixels and the blue marks the negative bags that do not contain building pixels. Here, the “building” class is specific to the buildings with a grey roof, as tagged with green color in OSM. Figure 6.44a shows the ground truth map for the (grey-roofed) buildings. The complete ground truth label map was created for all materials

³Map data copyrighted OpenStreetMap contributors and available from <http://www.openstreetmap.org>

⁴<https://www.google.com/earth/>

⁵<https://www.google.com/maps/>

in the MUUFL Gulfport data set and can be seen in [284].

Three sources were used for fusion. First, building points were extracted from the scene in training data and the mean target signature of these point were computed, as shown in Figure 6.45a. Notice only the “light grey roof” and “medium grey roof” are applicable in this experiment for detecting grey roofs and were used. The ACE detector was applied over the entire HSI image with the grey roof signature as target, and the confidence map yielded by ACE is as shown in Figure 6.45b. As can be seen, the ACE confidence map highlights the buildings, but also the roads which has similar asphalt material. Besides, the top right building has darker grey roofs and has low confidence values by using ACE. It is, thus, useful to fuse other sources such as LiDAR information with the ACE detection results.

Two LiDAR detection maps were generated to be fused with the ACE detector [277–279]. The LiDAR height information of extracted training building points were plotted in a histogram, as shown in Figure 6.46a. It is assumed in this experiment, specific to this scene, that buildings do have different heights and of course would be significantly different than height of the road surface materials. The peaks of the histogram was found by using the MATLAB *findpeaks()* function, as shown in Figure 6.46b. The Euclidean distance of all the LiDAR points in the scene were computed against the peak height values of the building points. As high confidence values were desired on points that have similar height to the training data, a Gaussian function was applied to the distance, as follows:

$$Conf_{lidar} = \exp \left\{ -\frac{d_{lidar}}{2} \right\} \quad (6.5)$$

where d_{lidar} is the Euclidean distance between all the LiDAR points in the scene and the peak height value. The LiDAR confidence value, $Conf_{lidar}$, would thus have high values on points where d_{lidar} is small (where points are similar to training building points) and low values on points with larger distance. The top two were selected as sources. Figure 6.47a

and Figure 6.47b shows the confidence maps based on rasterized LiDAR maps. As can be seen, one LiDAR source highlights the three buildings that are the tallest and the other LiDAR source highlights the topleft building. These two maps were used as the second and third training sources in the (rasterized) fusion experiments. The confidence values were computed in the same way for all points in the LiDAR point cloud as well and were used for the MR-MICI fusion experiment.

The comparison fusion methods were similar to those described in Section 6.1.4. In addition, the proposed MICI algorithms were applied to the rasterized data as a comparison. After training on one campus and obtaining the fuzzy measure for fusion, the test sources were obtained in the same way as training and the Choquet integral was computed as the detection results.

Figures 6.48 to 6.51 show the test confidence map on building detection with all comparison methods. Figures 6.52a and Figures 6.52b show the overall ROC curve on building detection, cross validated between campus 1 and 2. The first two columns of Table 6.13 shows the Area Under Curve (AUC) results for building detection. The ACE, Lidar1 and Lidar2 above the dotted line shows results for the three sources and methods below the dotted line are fusion results. As can be seen, MR-MICI has the best-performing ROC curve with the highest AUC. Table 6.14 shows the comparison of Root Mean Square Error (RMSE) of MICI and MR-MICI methods. These two evaluation methods were chosen as the AUC shows how well the method detects the target class – buildings, and the RMSE shows how the detection results on both the building and non-building points differ from the ground truth. As can be seen, the MR-MICI has a lower RMSE overall than MICI method as well.

Now, the MICI method uses rasterized LiDAR imagery while the MR-MICI uses raw

LiDAR points directly. It would be interesting to see the performance of the fusion methods on where the rasterization may be noisy or mis-aligned. Figure 6.53 and Figure 6.54 show the difference maps between the lidar points that MR-MICI picked and the rasterized LiDAR imagery (hereinafter called “LiDAR edge maps”), and the difference maps between the mean, min, and max of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery. As can be seen, the difference maps highlight mostly edges of buildings and trees, which is understandable as the edges are where rasterization would have possible mis-alignment between a tall object (such as building and trees) and the ground-level surface. Figure 6.55 and Figure 6.58 show the ROC curve the MUUFL Gulfport building detection scored on lidar edge map, mean difference map, min difference map, and max difference map with FAR up to 10^{-3} , cross validated between campus 1 and campus 2. Table 6.15 and Table 6.16 records the AUC scored on the difference maps with FAR up to 10^{-3} and the RMSE of all the fusion methods. As can be seen from the ROC curves and AUC tables, the MR-MICI has a much better performance over comparing methods specifically on the edge areas.

Similar to the MICI models, the Choquet integral has $O(2^m)$ parameters for m sources. Assume N_h is the total number of combinations that the sources can make (so that we need to compute CI for N_h times), the computational complexity of computing the CI in the objective function (which is the main part of the computation) for I iterations across B bags is $O(IN_hB2^m)$.

MUUFL Gulfport Sidewalk and Road Detection

Similarly, the same experiment was conducted for sidewalk and road detection. Figure 6.59 and Figure 6.60 shows the ground truth map and SLIC segmentation results for side-

Table 6.13: The AUC results of building, sidewalk, and road detection using MUUFL Gulfport HSI and LiDAR data. The best two results with the highest AUC were **bolded** and underlined, respectively.

	Building Detection		Sidewalk Detection		Road Detection	
	Train1Test2	Train2Test1	Train1Test2	Train2Test1	Train1Test2	Train2Test1
ACE	0.906	0.952	0.882	0.931	0.896	0.902
Lidar1	0.897	0.880	0.772	0.769	0.752	0.748
Lidar2	0.856	0.839	0.670	0.669	0.784	0.779
SVM	0.694	0.738	0.622	0.663	0.806	0.396
min	0.885	0.867	0.830	0.885	<u>0.896</u>	0.918
max	0.943	0.931	0.754	0.754	0.785	0.779
mean	0.957	0.953	0.831	0.870	0.849	0.856
mi-SVM	0.881	0.800	0.721	0.904	0.791	0.817
CI-QP	0.943	0.931	0.767	<u>0.918</u>	0.801	0.815
MICI	0.952(0.000)	<u>0.956(0.000)</u>	<u>0.838(0.009)</u>	0.908(0.001)	0.873(0.011)	0.824(0.003)
MR-MICI	0.952(0.000)	0.977(0.000)	0.854(0.019)	0.861(0.010)	0.905(0.002)	0.895(0.003)

Table 6.14: The RMSE results of MICI and MR-MICI on building, sidewalk, and road detection.

	Building Detection		Sidewalk Detection		Road Detection	
	Train1Test2	Train2Test1	Train1Test2	Train2Test1	Train1Test2	Train2Test1
MICI	0.403(0.002)	0.382(0.000)	0.485(0.002)	0.466(0.002)	0.480(0.009)	0.514(0.002)
MR-MICI	0.351(0.004)	0.331(0.001)	0.460(0.007)	0.489(0.006)	0.448(0.007)	0.478(0.008)

walk and road detection experiments. Figure 6.61 shows the test confidence map for sidewalk and road detection for the proposed MR-MICI method. Figure 6.62 shows the cross-validated overall ROC curve results on sidewalk detection and Figure 6.63 shows the cross-validated overall ROC curve results on road detection. The AUC results can be seen in Table 6.13. As can be seen, the MR-MICI is able to detect sidewalk and road classes, respectively and achieve a good ROC curve performance. Note that the difference map between the raw and rasterized LiDAR do not have significant difference on sidewalk and road and, therefore, only the overall ROC curves for sidewalk and road classes are presented.

Table 6.15: The AUC and RMSE results of MICI and MR-MICI on building detection, scored on edges. Train on campus 1 and test on campus 2.

	picked points		max diff map		min diff map		mean diff map	
	AUC	RMSE	AUC	RMSE	AUC	RMSE	AUC	RMSE
SVM	0.420	0.040	0.113	0.067	0.141	0.125	0.078	0.063
mi-SVM	0.704	0.031	0.327	0.076	0.448	0.140	0.490	0.071
CI-QP	0.329	0.055	0.135	0.080	0.126	0.147	0.115	0.073
MICI	0.371(0.021)	0.046(0.000)	0.311(0.017)	0.068(0.001)	0.190(0.022)	0.125(0.002)	0.401(0.020)	0.062(0.001)
MR-MICI	0.776(0.004)	0.022(0.001)	0.458(0.022)	0.049(0.000)	0.614(0.025)	0.082(0.001)	0.619(0.027)	0.044(0.000)

Table 6.16: The AUC and RMSE results of MICI and MR-MICI on building detection, scored on edges. Train on campus 2 and test on campus 1.

	picked points diff map		max diff map		min diff map		mean diff map	
	AUC	RMSE	AUC	RMSE	AUC	RMSE	AUC	RMSE
SVM	0.513	0.058	0.413	0.101	0.537	0.176	0.390	0.109
mi-SVM	0.695	0.021	0.488	0.018	0.577	0.031	0.528	0.017
CI-QP	0.094	0.104	0.096	0.113	0.027	0.202	0.000	0.119
MICI	0.683(0.004)	0.077(0.000)	0.451(0.008)	0.091(0.000)	0.375(0.009)	0.162(0.000)	0.448(0.009)	0.096(0.000)
MR-MICI	0.794(0.007)	0.035(0.000)	0.529(0.003)	0.061(0.000)	0.649(0.007)	0.103(0.000)	0.638(0.005)	0.064(0.000)

6.2.4 Soybean and Weed Data Set

Another soybean and weed data set was used to test the performance of MR-MICI. Three imagery were available over a patch of soybean field⁶ and the goal is to detect weed amongst the soybean plants. The height map is 351×1450 in size and the RGB map is 1404×5864 in size. The MR-MICI was used to fuse those images. For comparison, the RGB imagery was downsampled to be the same size as 351×1450 and used for fusion.

Figure 6.65 shows the height map over the soybean-weed field. As can be seen from the figure, weed has different height values than soybean plant, therefore the height map can be a useful source. Four Gabor filters at angles 0° , 45° , 90° , and 135° were applied to the height map and the sum of the filtered images was used as one of the fusion source, as shown in Figure 6.68.

Figure 6.64 shows the downsized RGB map. In this experiment, the RGB values were transformed into the LAB space and the L and B band imagery were used as the other two

⁶Data provided by Precision Silver, LLC.

sources for fusion. The weed pixels seem to have lighter color than the soybean plants, and the L and B band are able to provide distinct features to separate weed from soybean plants. The L dimension provide information about lightness and the B dimension is the color opponent for blue-yellow space. Figure 6.66 and Figure 6.67 show the L and B band imagery, which are useful in highlighting the weed pixels.

Figure 6.69 shows the ground truth map for weed in the data. Figure 6.70 shows the SLIC segmentation and bag construction in the data set. Figure 6.72 and Figure 6.73 show the confidence map obtained after MICI and MR-MICI fusion. Figure 6.71 shows the overall ROC curve for all the comparing fusion methods for one run. The MR-MICI method was able to detect weed in the scene and yield higher AUC performance.

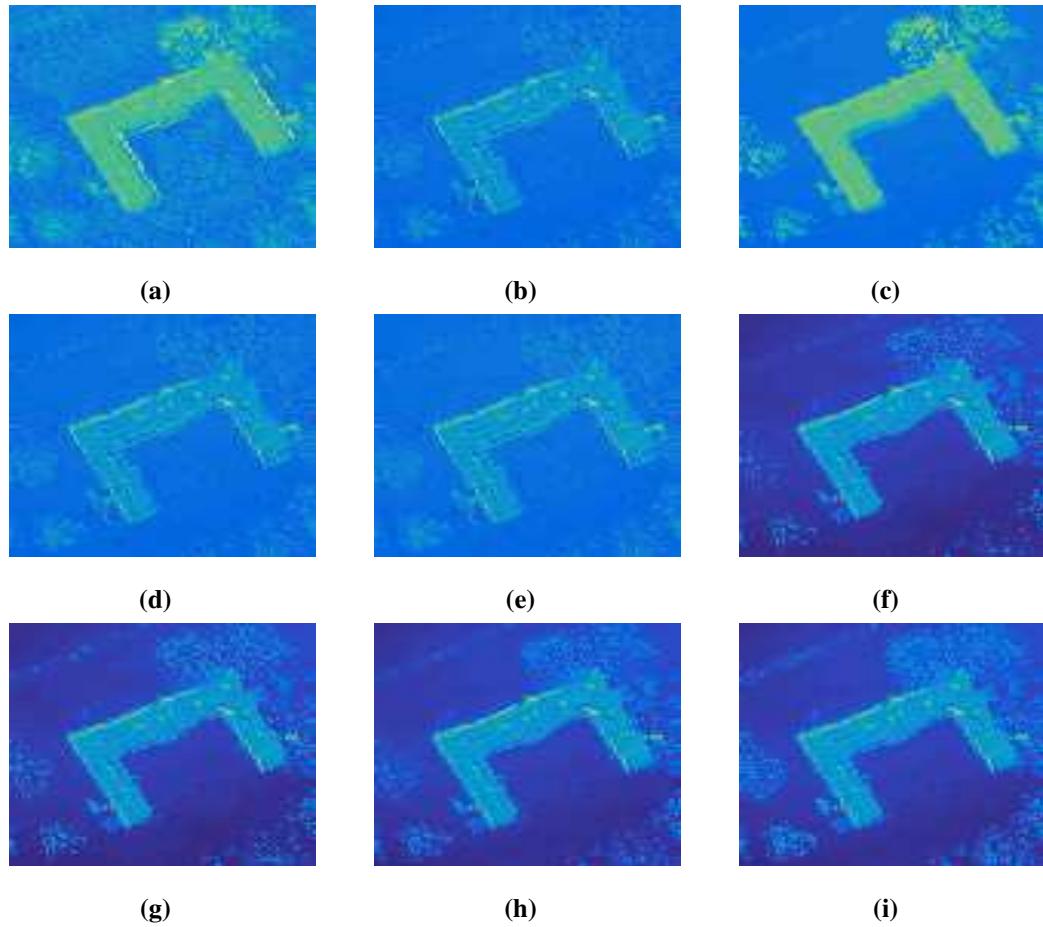
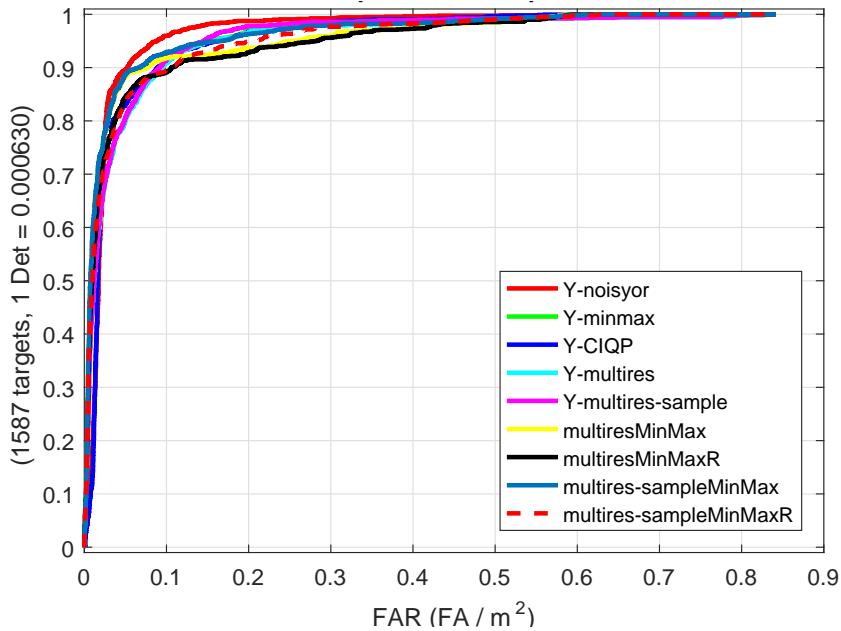
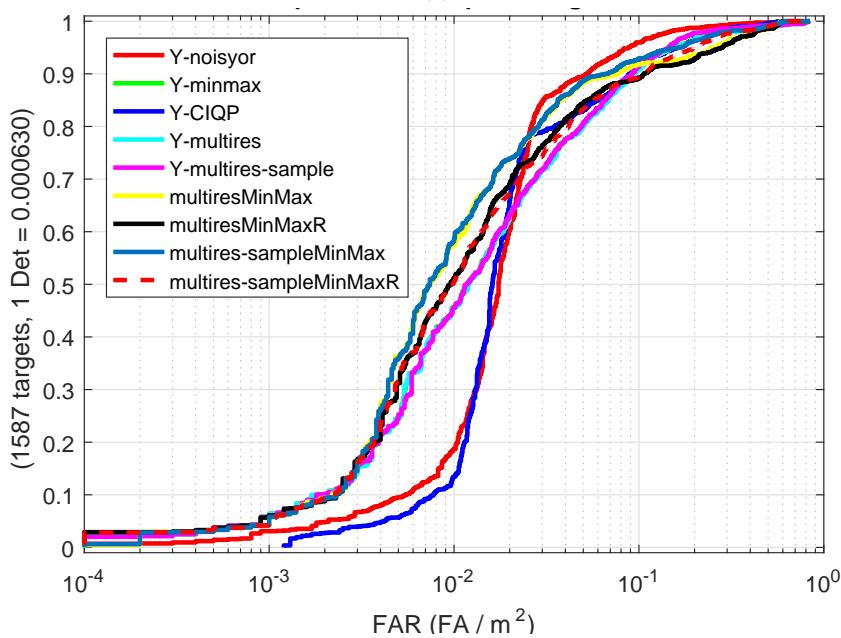


Figure 6.35: Results of building classification, train on sub-image 1, test on sub-image 1. (a) MICI noisy-or model; (b) MICI min-max model; (c) CI-QP; (d) MR-MICI, plot test map on rasterized data; (e) MR-MICI-sampling, plot test map on rasterized data; (f) MR-MICI, plot test map if knowing bag-level labels; (g) MR-MICI, plot test map if bag-level labels are unknown; (h) MR-MICI-sampling, plot test map if knowing bag-level labels; (i) MR-MICI-sampling, plot test map if bag-level labels are unknown.



(a)



(b)

Figure 6.36: ROC curve results of building classification results, train on sub-image 1, test on sub-image 1. (a) ROC curve; (b) ROC curve on log scale.

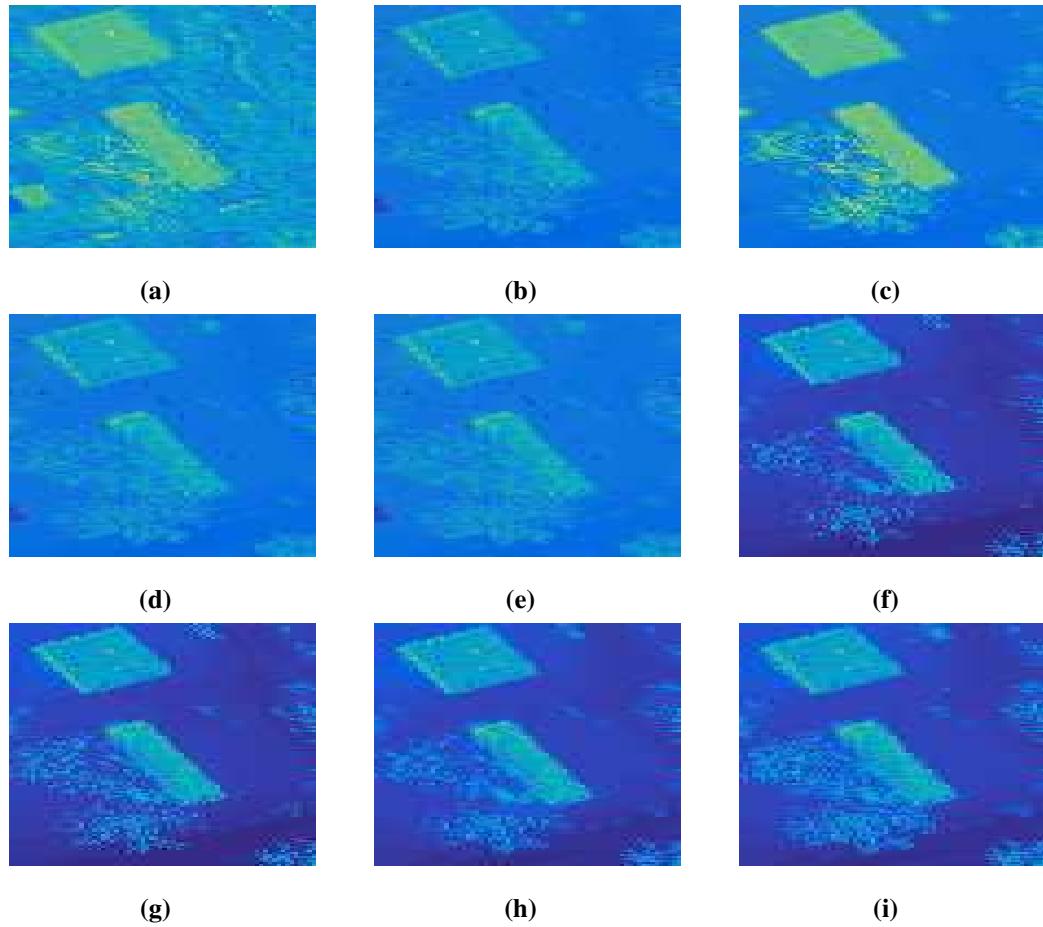
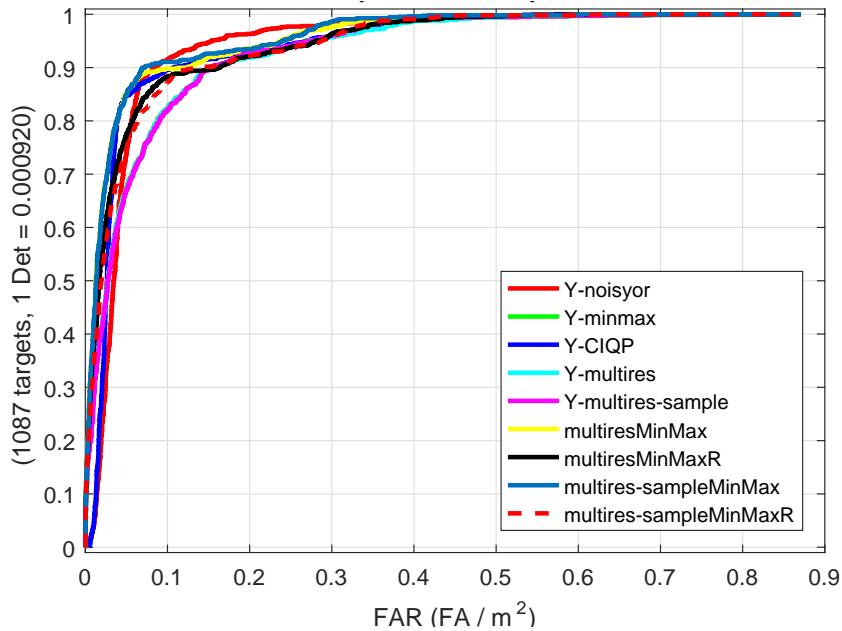
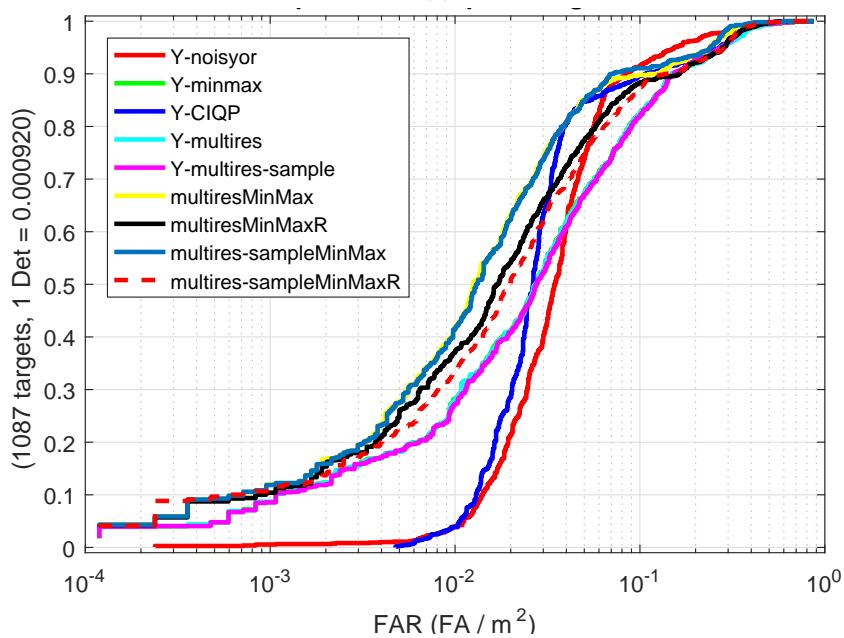


Figure 6.37: Results of building classification, train on sub-image 1, test on sub-image 3. (a) MICI noisy-or model; (b) MICI min-max model; (c) CI-QP; (d) MR-MICI, plot test map on rasterized data; (e) MR-MICI-sampling, plot test map on rasterized data; (f) MR-MICI, plot test map if knowing bag-level labels; (g) MR-MICI, plot test map if bag-level labels are unknown; (h) MR-MICI-sampling, plot test map if knowing bag-level labels; (i) MR-MICI-sampling, plot test map if bag-level labels are unknown.



(a)



(b)

Figure 6.38: ROC curve results of building classification results, train on sub-image 1, test on sub-image 3. (a) ROC curve; (b) ROC curve on log scale.



Figure 6.39: Four LiDAR lines in MUUFL Gulfport data, shown in Google Earth.

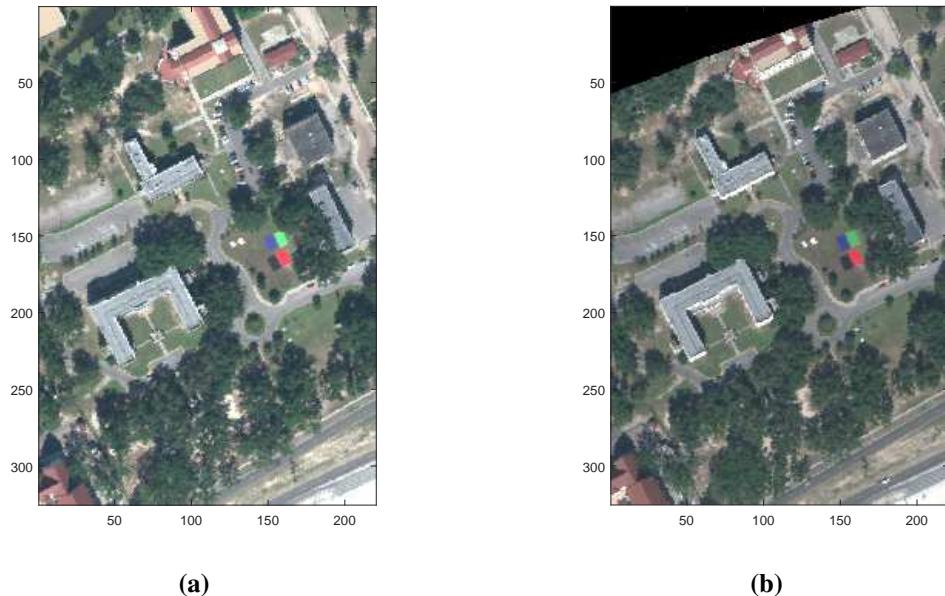


Figure 6.40: RGB image of MUUFL Gulfport data. (a) Campus 1; (b) Campus 2. Notice the black regions in (b) are invalid regions from the data collection and were excluded from any training or testing process in the experiment.

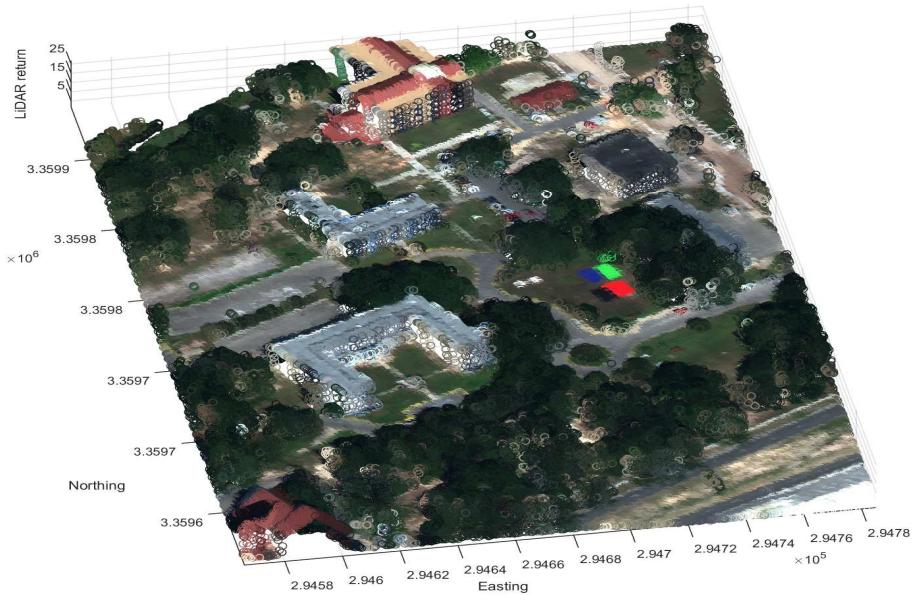


Figure 6.41: Scatterplot of LiDAR line 1 point cloud in MUUFL Gulfport campus 1 data. The color corresponds to the RGB color over the scene. “Z”-axis is the height of each point. “X” and “Y” are UTM coordinates.

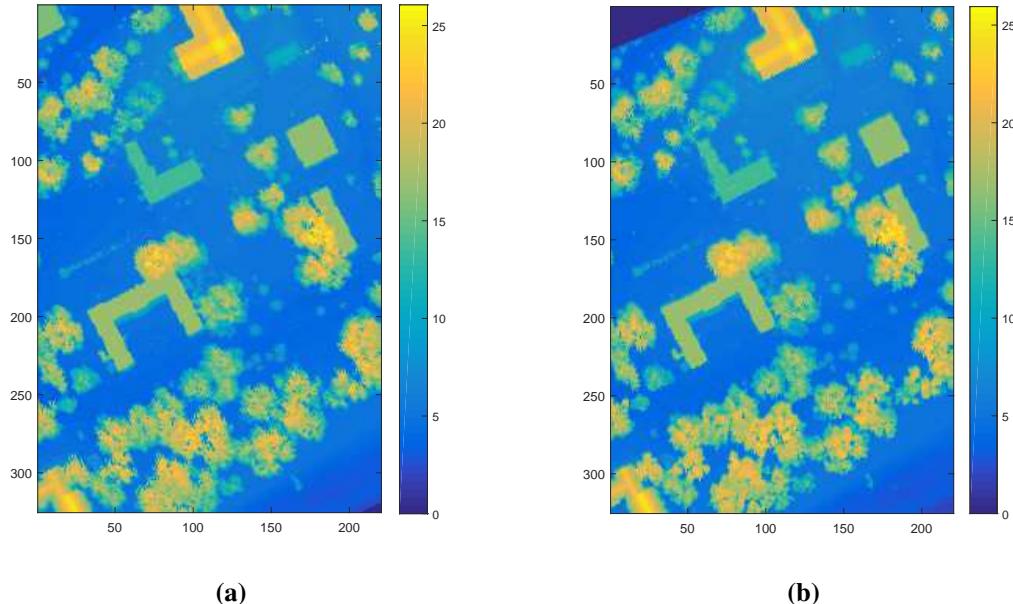


Figure 6.42: Raster image of the first return MUUFL Gulfport LiDAR data. The color represents the lidar height information. The rasterized image is in the same size as the hyperspectral imagery. (a) Campus 1; (b) Campus 2.

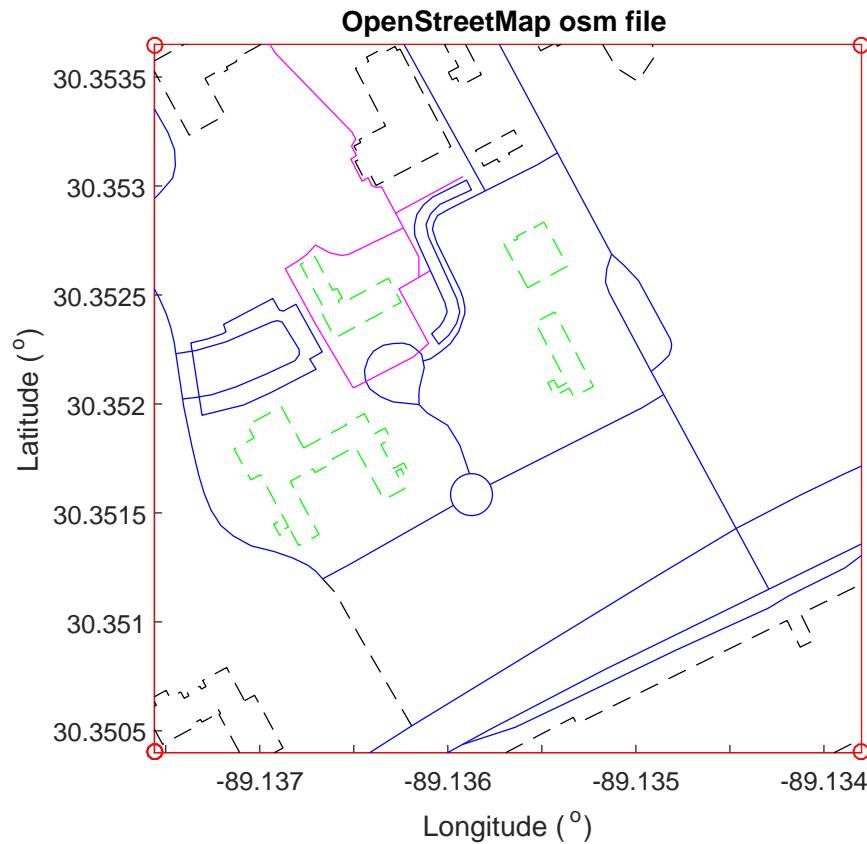


Figure 6.43: Open Street Map imagery over MUUFL Gulfport campus 1. The blue lines corresponds to road, highway, and parking lot (asphalt materials). The magenta lines corresponds to sidewalk/footway. The green lines marks buildings—Notice that in the building detection experiments, only the grey-roof buildings in the center of the map was used as targets. The black lines corresponds to “other” tags.

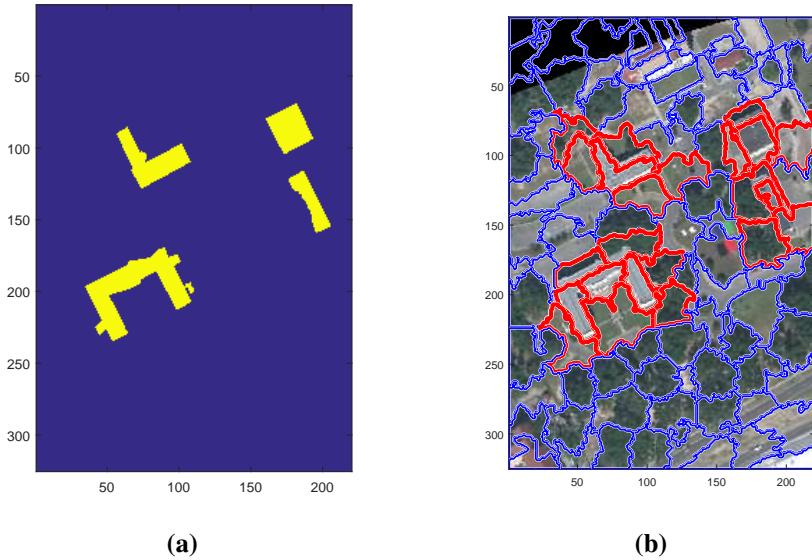


Figure 6.44: The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for building detection. (a) The Ground Truth map of the buildings in MUUFL Gulfport HSI data. (b) The SLIC segmentation result on MUUFL Gulfport HSI data. Red marks positive training bags and blue marks negative bags for building detection experiment.

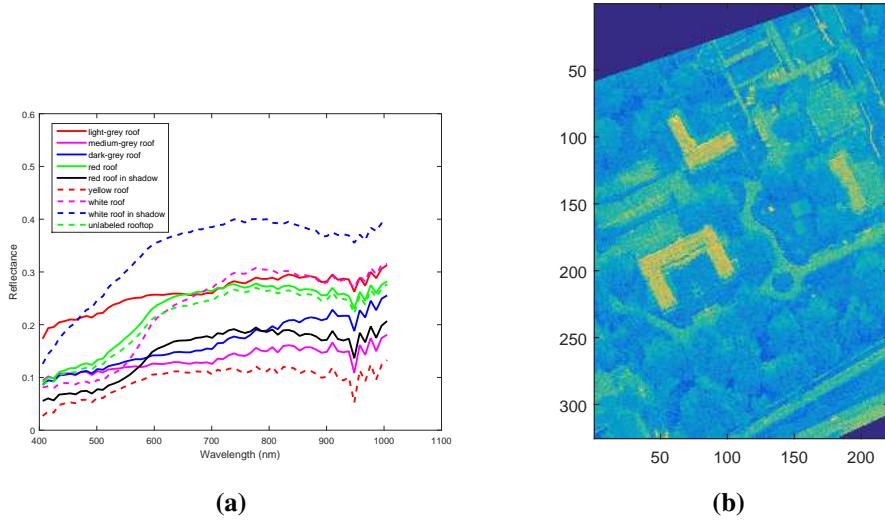


Figure 6.45: The building signature for ACE detector and the ACE detection map for the MUUFL Gulfport HSI data. (a) The building signature from the scene. In the “grey-roof” building detection, only the “light-grey roof” and “medium-grey” roof signatures were used. (b) The ACE detection confidence map.

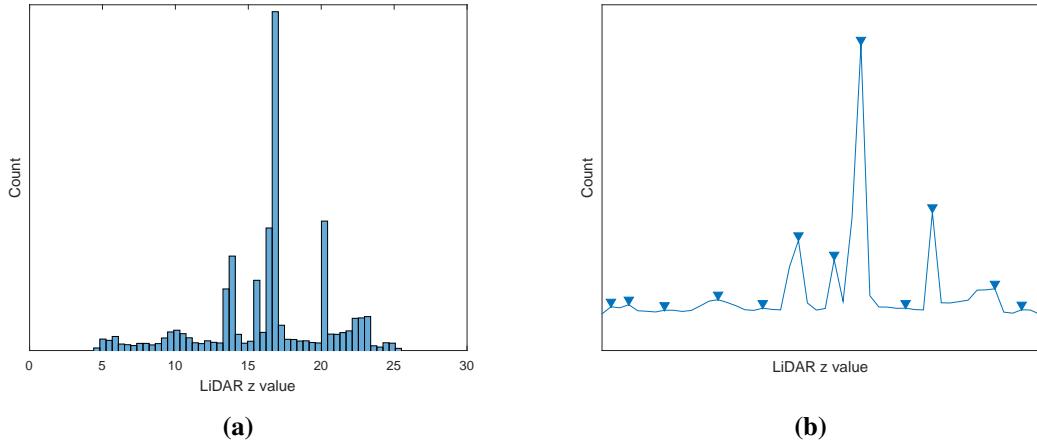


Figure 6.46: The histogram and peaks of the LiDAR values of building points. (a) The histogram of the LiDAR values of building points. (b) The peaks found based on the histogram in (a).

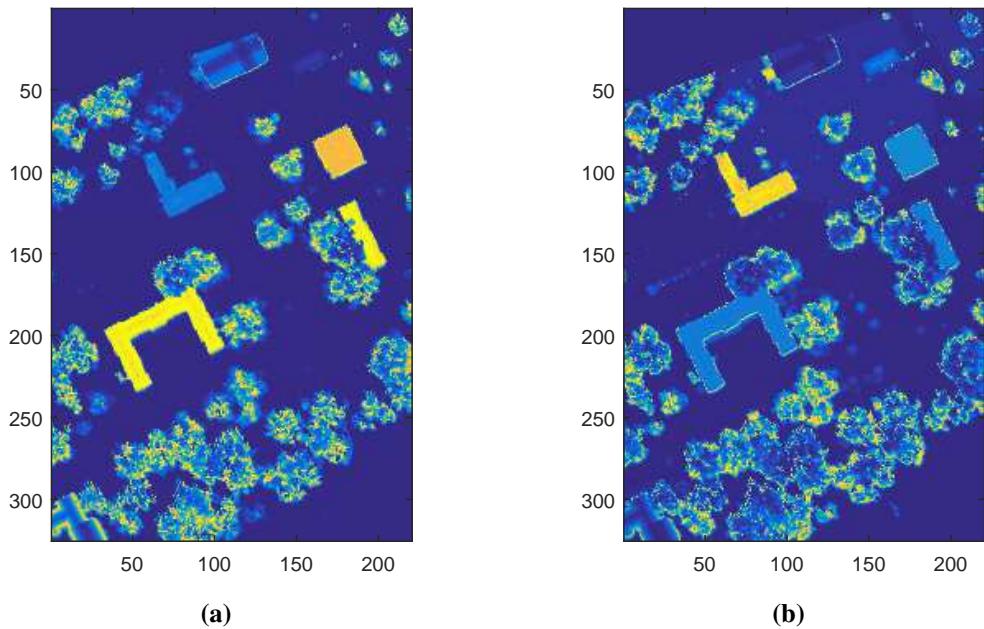


Figure 6.47: The LiDAR confidence maps for building detection in the MUUFL Gulfport HSI data. (a)(b) were distance maps computed against the top two peaks found in Figure 6.46.

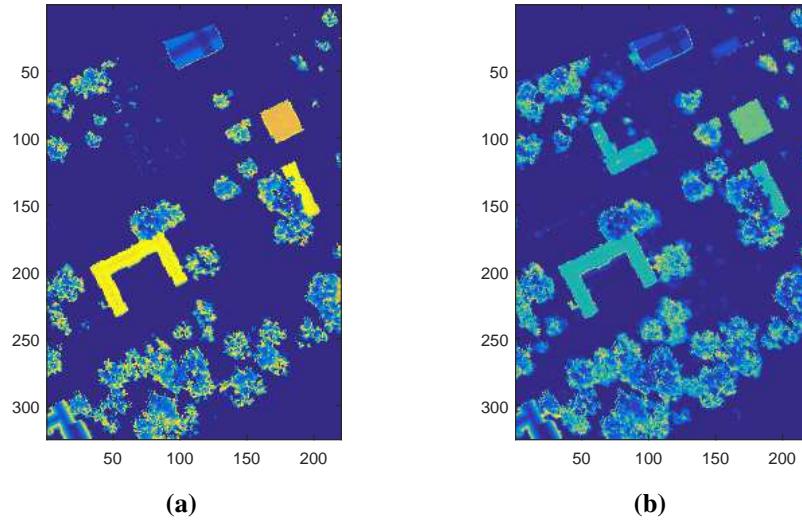


Figure 6.48: The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for SVM and min methods. Train on campus 1 and test on campus 2. (a) Fusion by SVM; (b) Fusion by taking the min of the sources.

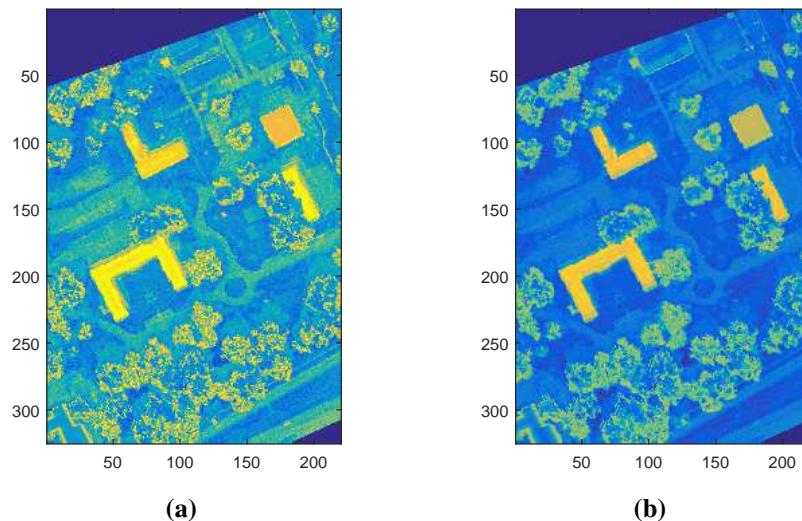


Figure 6.49: The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for taking the max and mean of the sources. Train on campus 1 and test on campus 2. Fusion by taking the (a) max and (b) mean of the sources.

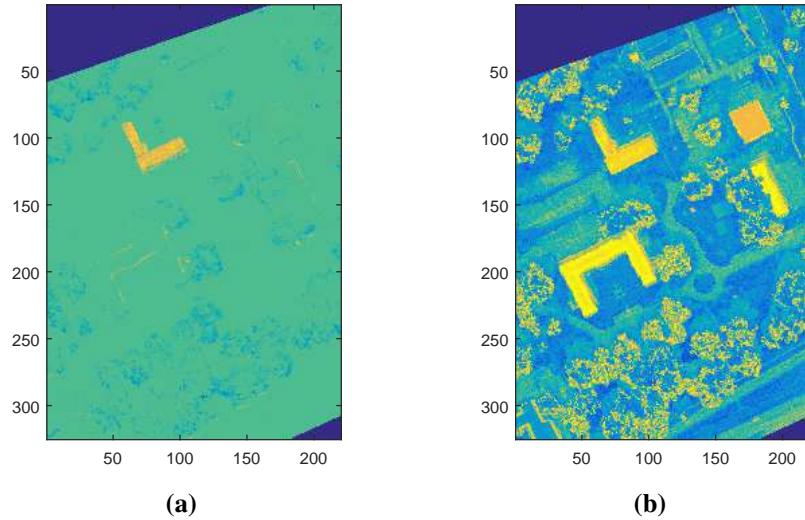


Figure 6.50: The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for the (a) mi-SVM and (b) CI-QP methods. Train on campus 1 and test on campus 2.

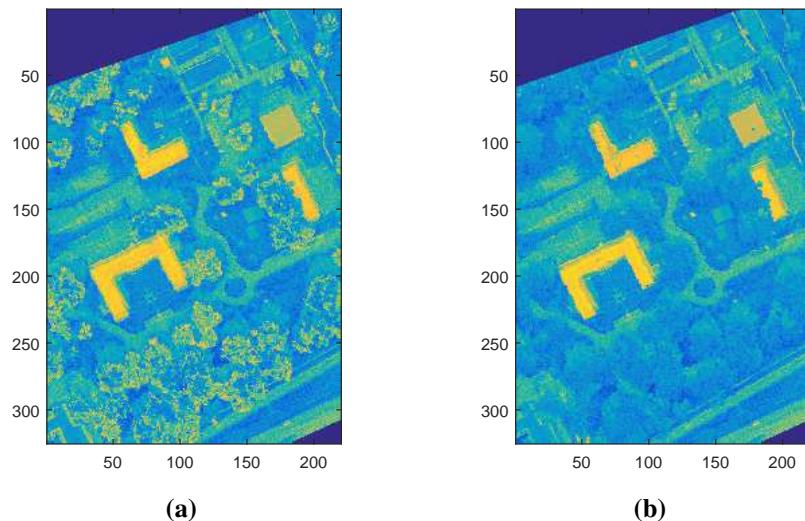
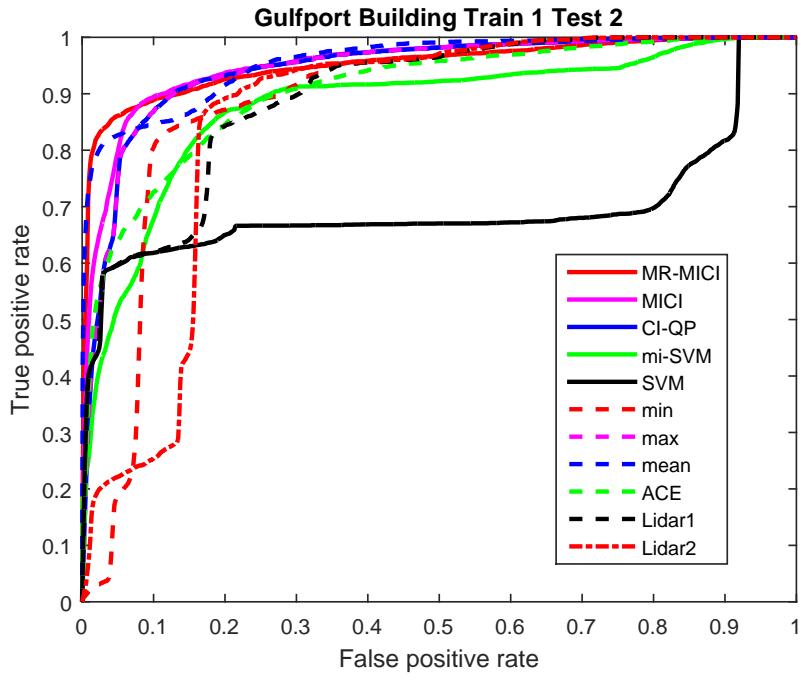
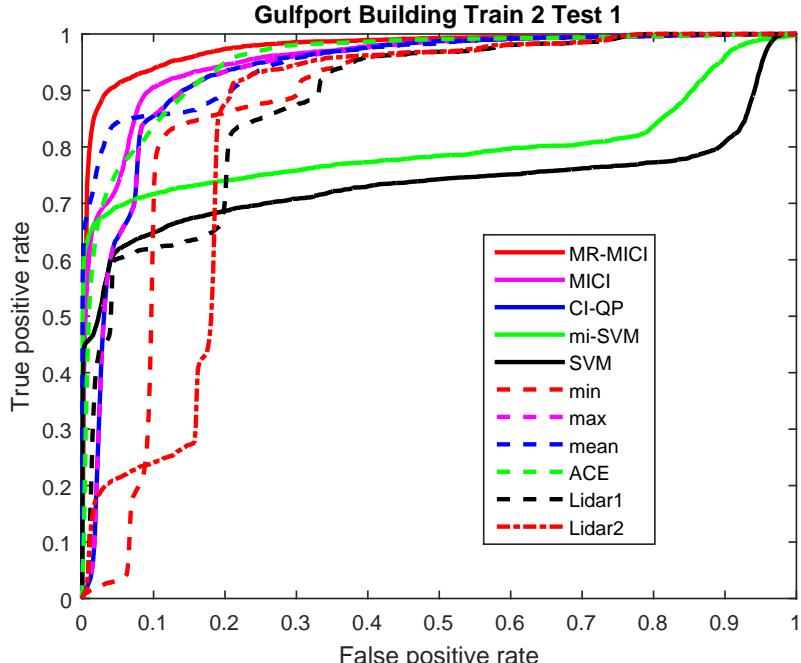


Figure 6.51: The fusion test confidence maps for building detection in the MUUFL Gulfport HSI data for the proposed (a) MICI and (b) MR-MICI methods. Train on campus 1 and test on campus 2.



(a)



(b)

Figure 6.52: The Overall ROC curve for building detection for MUUFL Gulfport data.
 (a) Train on campus 1, test on campus 2; (b) Train on campus 2, test on campus 1.

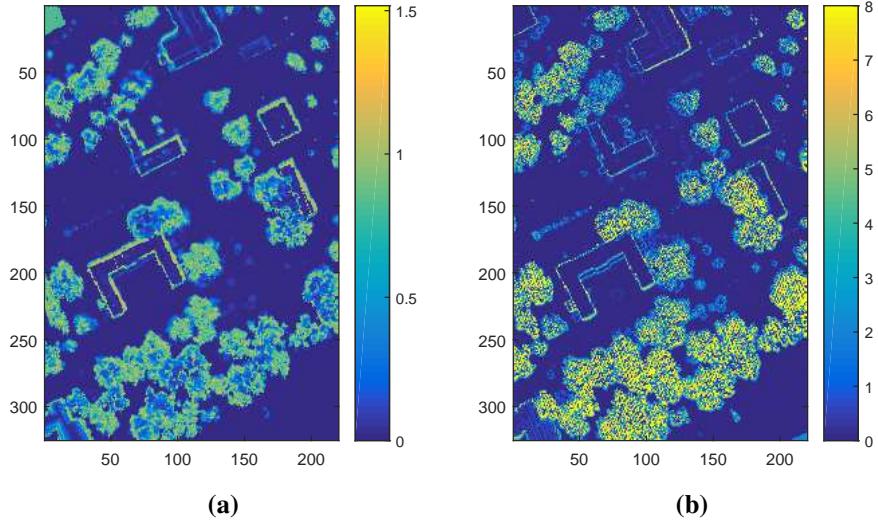


Figure 6.53: The difference map in the MUUFL Gulfport HSI data, (a) between the lidar points that MR-MICI picked and the rasterized LiDAR imagery; (b) between the mean of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery.

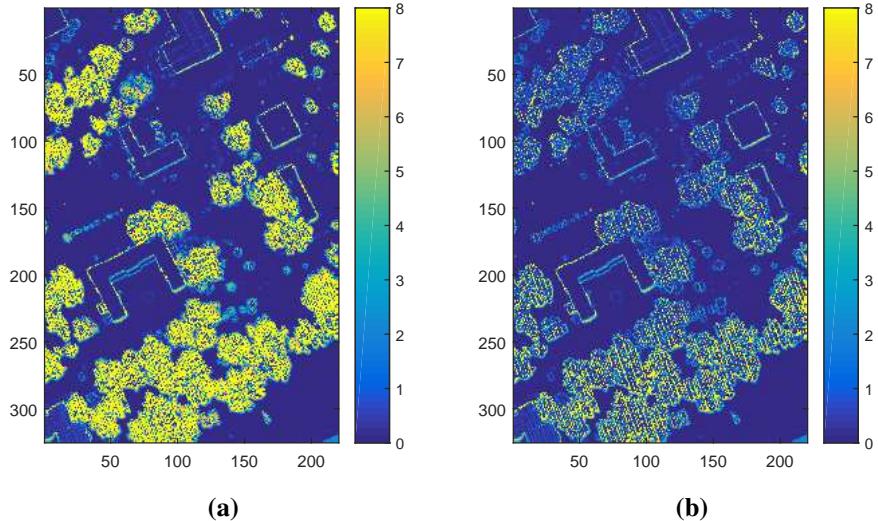


Figure 6.54: The difference map in the MUUFL Gulfport HSI data, (a) between the min of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery;; (b) between the max of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery.

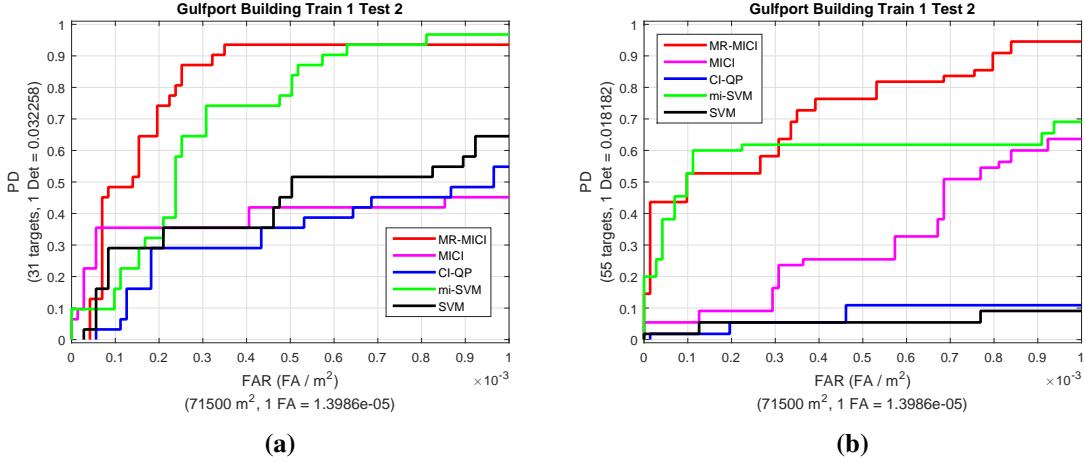


Figure 6.55: The ROC curve for building detection for MUUFL Gulfport data, scored on (a) the difference map between the lidar points that MR-MICI picked and the rasterized LiDAR imagery; (b) the difference map between the mean of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery. Train on Campus 1, Test on Campus 2.

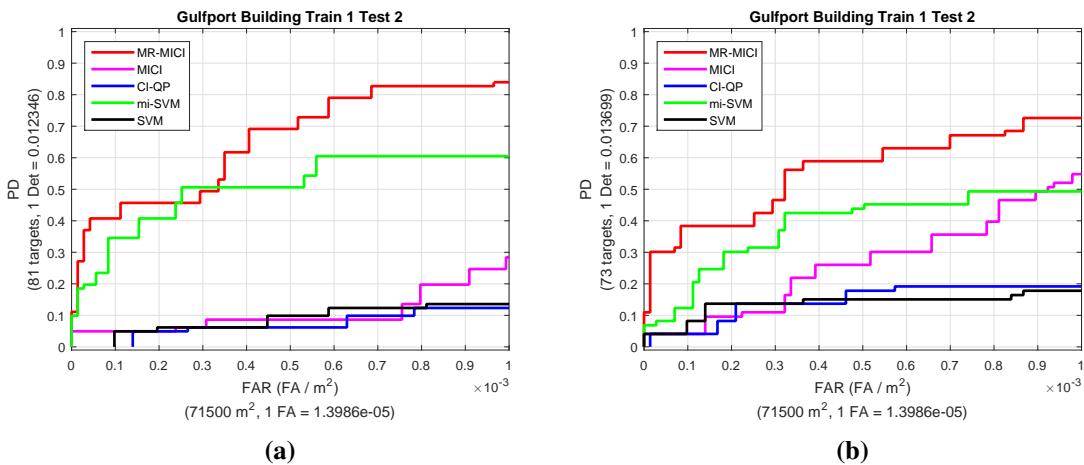


Figure 6.56: The ROC curve for building detection for MUUFL Gulfport data, scored on (a) the difference map between the min of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery; (b) the difference map between the max of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery. Train on Campus 1, Test on Campus 2.

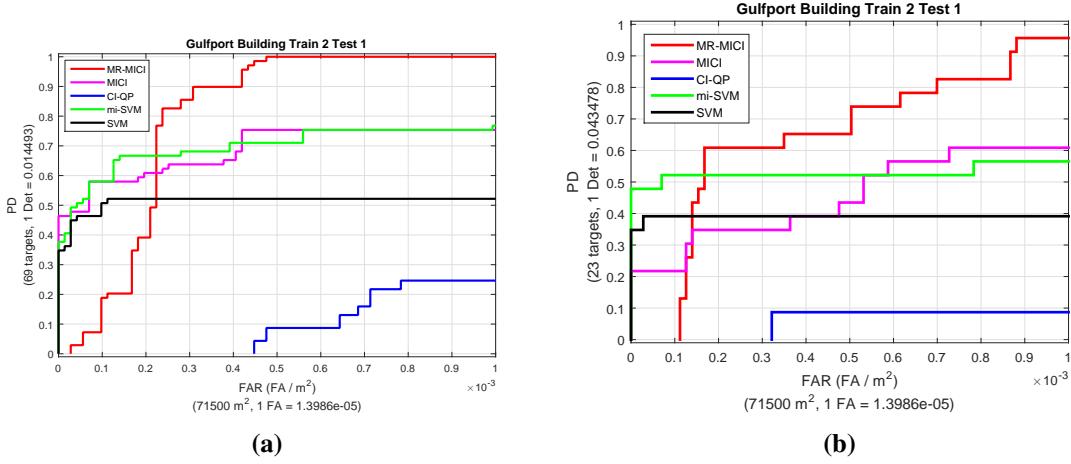


Figure 6.57: The ROC curve for building detection for MUUFL Gulfport data, scored on (a) the difference map between the lidar points that MR-MICI picked and the rasterized LiDAR imagery; (b) the difference map between the mean of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery. Train on Campus 2, test on Campus 1.

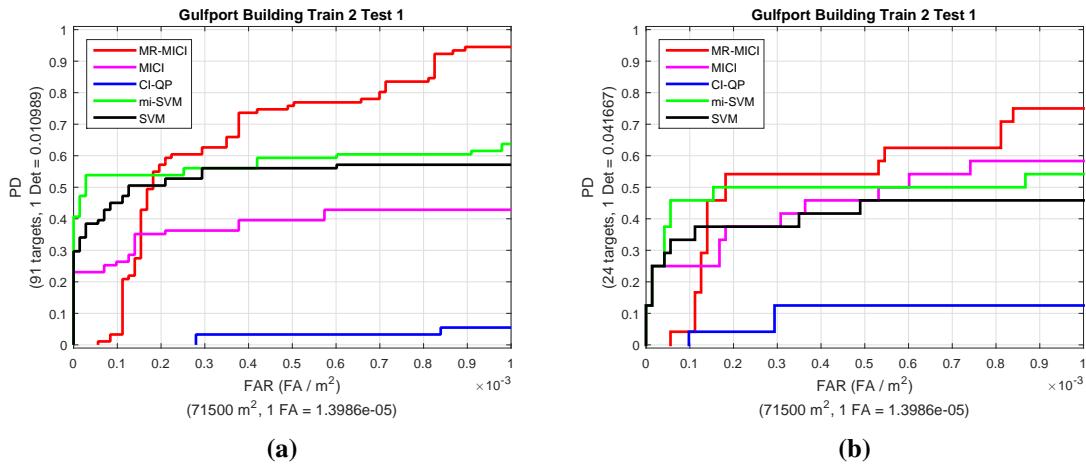


Figure 6.58: The ROC curve for building detection for MUUFL Gulfport data, scored on (a) the difference map between the min of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery; (b) the difference map between the max of the raw LiDAR points corresponding to each pixel and the rasterized LiDAR imagery. Train on Campus 2, test on Campus 1.

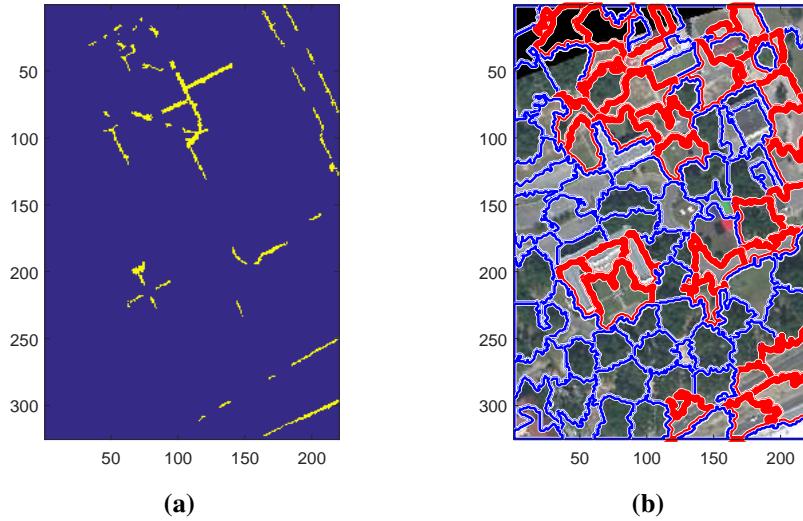


Figure 6.59: The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for sidewalk detection. (a) The Ground Truth map of the sidewalks in MUUFL Gulfport HSI data. (b) The SLIC segmentation result on MUUFL Gulfport HSI data. Red marks positive training bags and blue marks negative bags for sidewalk detection experiment.

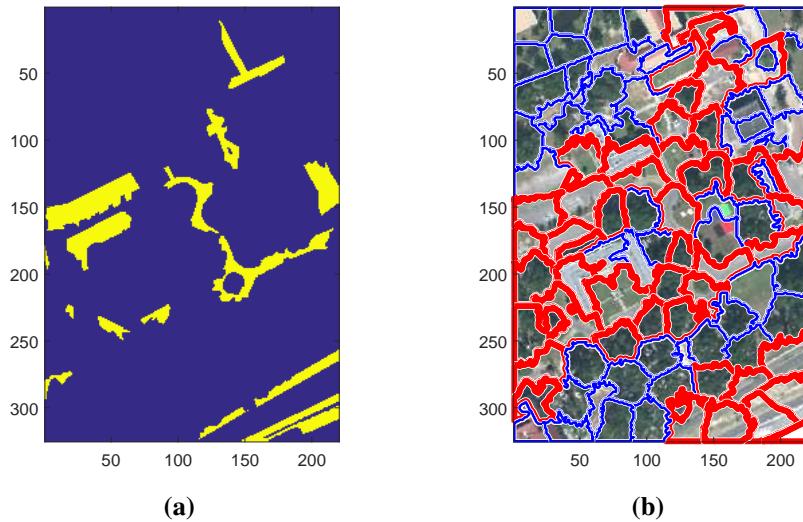


Figure 6.60: The Ground Truth map and the SLIC segmentation map of the MUUFL Gulfport HSI data for road detection. (a) The Ground Truth map of the roads in MUUFL Gulfport HSI data. (b) The SLIC segmentation result on MUUFL Gulfport HSI data. Red marks positive training bags and blue marks negative bags for road detection experiment.

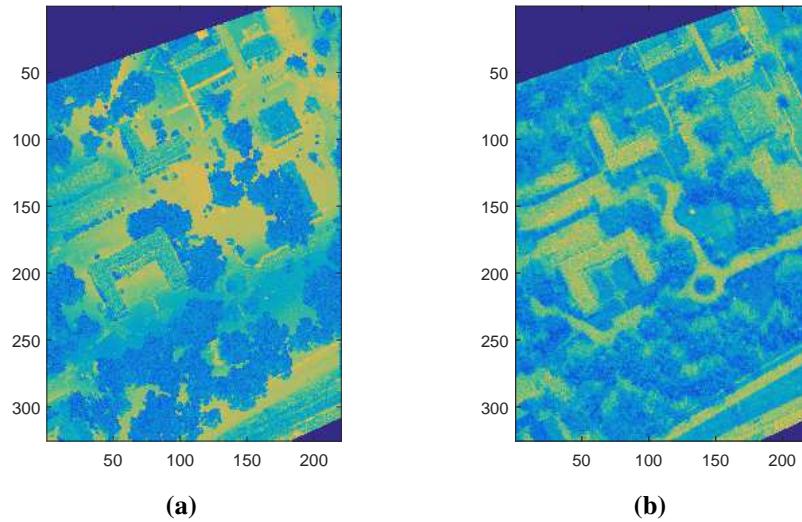


Figure 6.61: The fusion test confidence maps for (a) sidewalk and (b) road detection in the MUUFL Gulfport HSI data for the proposed MR-MICI method. Train on campus 1 and test on campus 2.

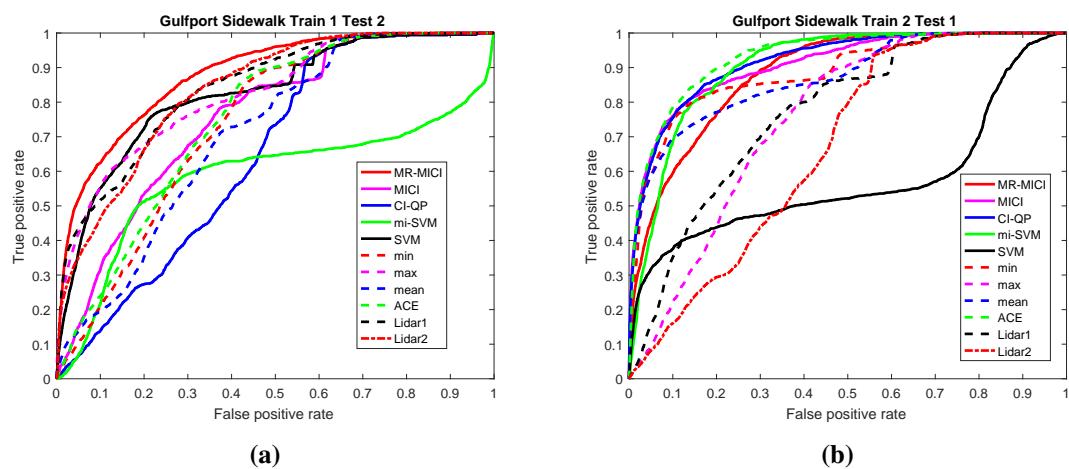


Figure 6.62: The Overall ROC curve for sidewalk detection for MUUFL Gulfport data.
 (a) Train on campus 1, test on campus 2; (b) Train on campus 2, test on campus 1.

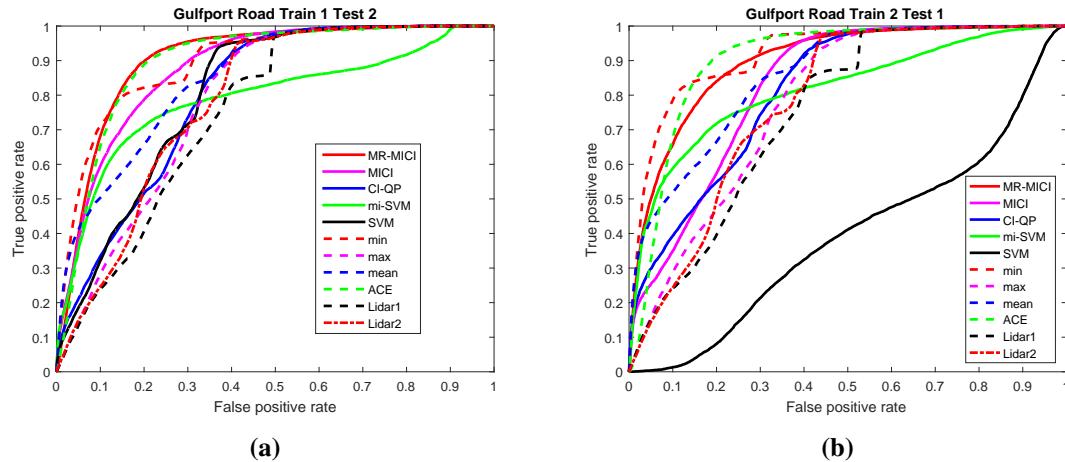


Figure 6.63: The Overall ROC curve for road detection for MUUFL Gulfport data. (a) Train on campus 1, test on campus 2; (b) Train on campus 2, test on campus 1.

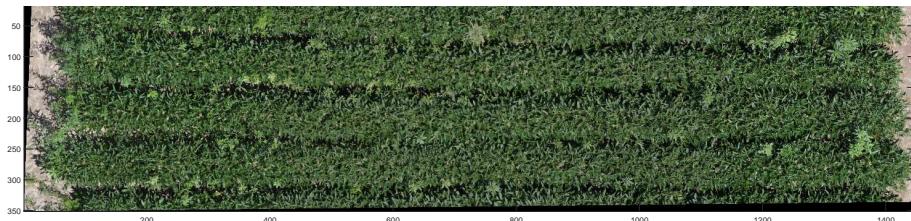


Figure 6.64: The RGB image of the soybean-weed data.

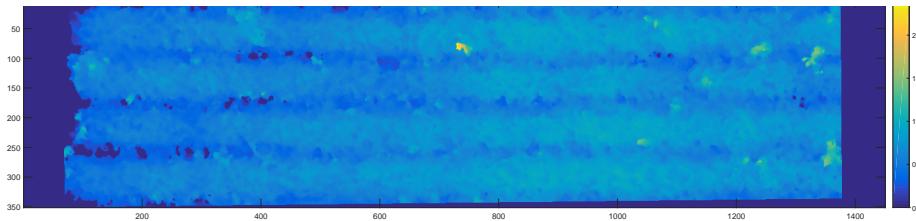


Figure 6.65: The height map of the soybean-weed data.

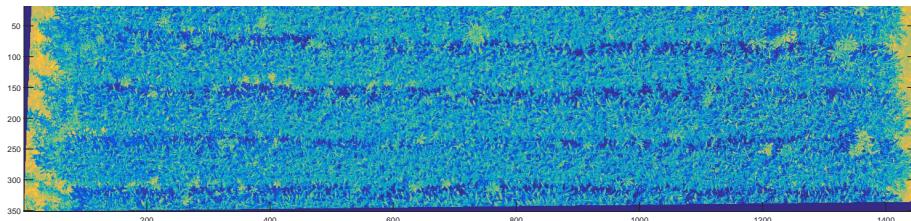


Figure 6.66: The L-band image of the soybean-weed data.

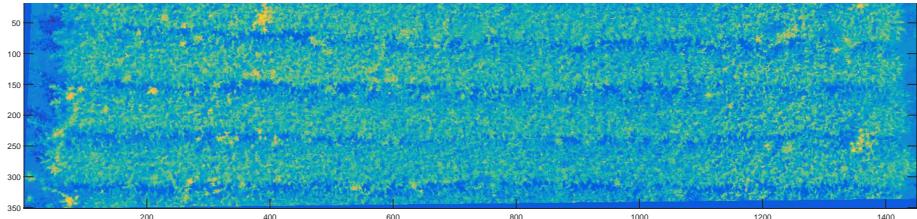


Figure 6.67: The B-band image of the soybean-weed data.

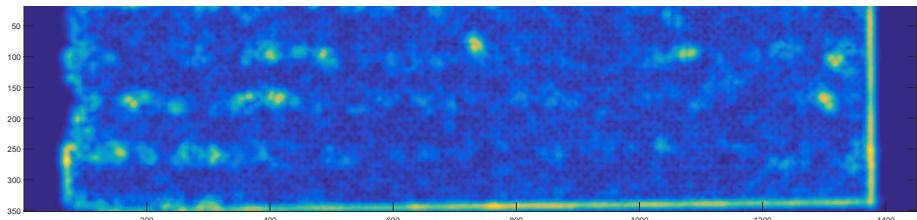


Figure 6.68: The Gabor filtered image of the soybean-weed data height map.

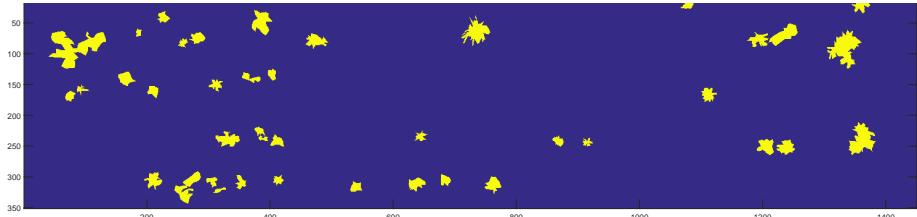


Figure 6.69: The Ground Truth map of weed in the soybean-weed data. The deep blue is the background (soybean plants) and the yellow marks the target (weed).

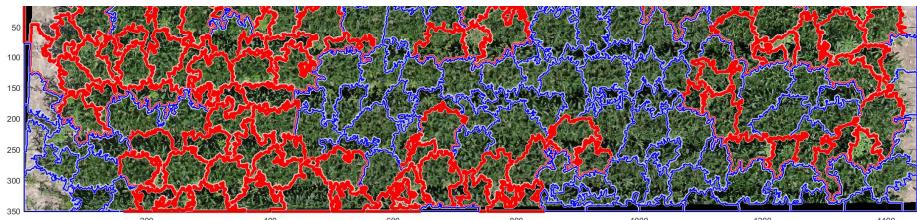


Figure 6.70: The SLIC segmentation map of the soybean-weed data. Red marks positive training bags and blue marks negative bags for weed detection experiment.

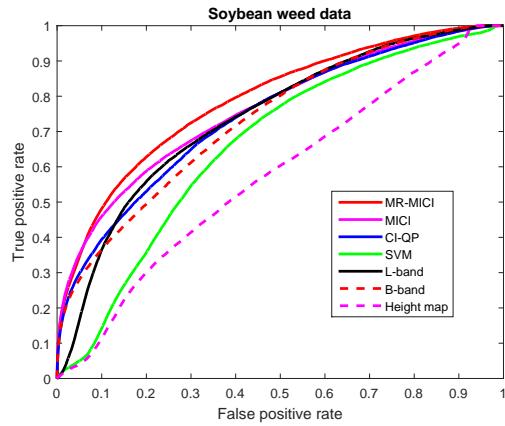


Figure 6.71: The ROC curve for weed detection in the soybean-weed data.

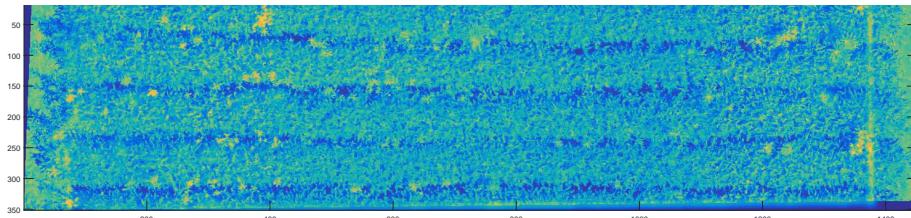


Figure 6.72: The confidence map obtained from MICI fusion for the soybean-weed data.

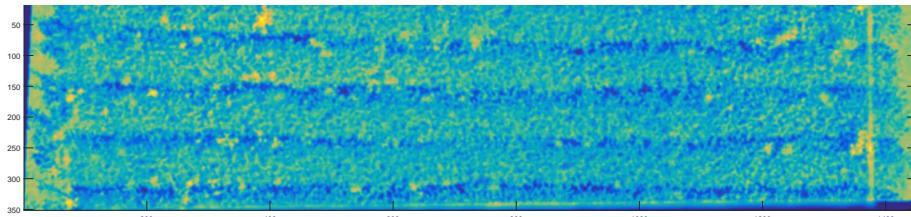


Figure 6.73: The confidence map obtained from the MR-MICI fusion for the soybean-weed data.

6.3 Regression Data Set

This section presents results on fusion with real-valued prediction values (regression) using the proposed Multiple Instance Choquet Integral Regression (MICI Regression) framework. Experiments were conducted on both synthetic regression data set as well as a real crop yield prediction application on a MODIS remote sensing data set provided from the HARVIST (Heterogeneous Agricultural Research Via Interactive, Scalable Technology) project.

6.3.1 Synthetic Regression Data Set

Synthetic 5-source regression data sets were constructed to investigate the performance of MICI regression with changes in the percentage of primary instances in the bag and signal-to-noise ratio (SNR). All data sets used in this section have 1000 data points, each with 5 dimensions and with values between [0, 1]. The data points were grouped into 10 bags, each bag with 100 data points. The data points in this regression data set have real-valued labels between [0, 1]. Relative error [57] is used to evaluate the performance for the MICI Regression:

$$Error_{reg}(y, \hat{y}) = \begin{cases} \left| \frac{y - \hat{y}}{y} \right|, & \text{if } y \in (0, 1] \\ |y - \hat{y}|, & \text{if } y = 0 \end{cases} \quad (6.6)$$

where y is the true label and \hat{y} is the estimated label for each data point.

MICI Regression model operates under the assumption that only one primary instance is associated with the label of each bag [25]. What happens when there are more or less primary instances in the bag? In this experiment, the percentage of primary instances in the bag is changed to observe the performance of MICI. The percentage of primary instances

in the bag takes the value of 0% to 100% with an increment of 10%. For each bag, the “primary instances” have label values that are exactly the same as the bag-level training label. The non-primary instances have randomly generated label values (different from the bag-level labels) that are generated from a completely random measure. Figure 6.74 shows the measure used to generate the data set and labels for each data point and for the bag. Table 6.17 shows the relationship between the percentage of primary instances in the bag and the mean relative error over all the data points across five runs. As can be seen from the table, when the percentage of primary instances in the bag increases, the relative error decreases.

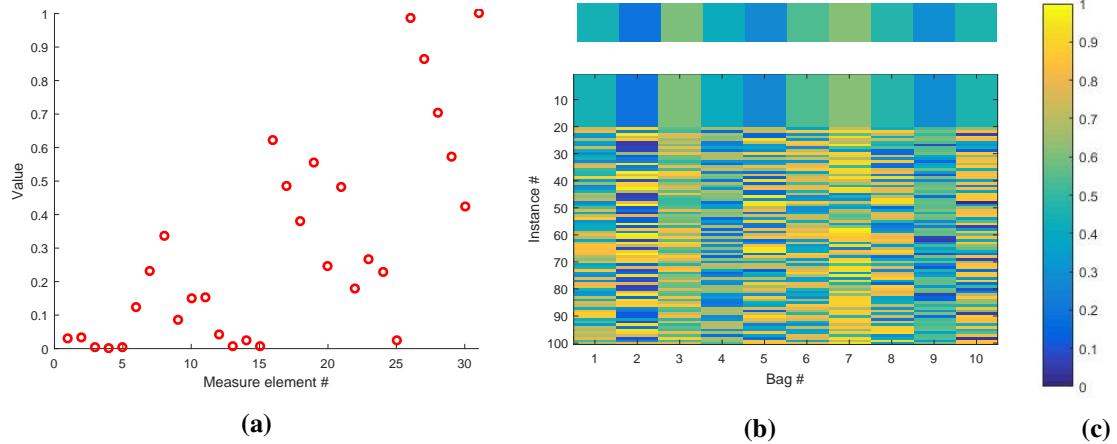


Figure 6.74: Contamination data set for MICI Regression model experiments: (a) True measure values; (b) True labels per bag; (c) Colorbar.

Table 6.17: Relative error versus percentage of primary instances for synthetic regression data set for MICI Regression model across five runs.

	Percentage of primary instances					
	0%	10%	20%	30%	40%	50%
Relative Error	0.730(0.001)	0.488(0.001)	0.492(0.000)	0.451(0.001)	0.365(0.000)	0.301(0.001)
	60%	70%	80%	90%	100%	
Relative Error	0.345(0.000)	0.274(0.001)	0.104(0.001)	0.059(0.002)	0.002(0.002)	

Then, the performance of MICI is observed with varying levels of signal-to-noise ratio

(SNR). In this experiment, 100% of the points in the bag are primary instances and varying amount of noise is add to the entire data set, creating SNR value from 50dB to 0dB with an increment of -5dB. Table 6.18 shows the relationship between the SNR values in the bag and the mean relative error over all the data points across five runs. As can be seen from the table, when the SNR value increases, the relative error decreases.

Table 6.18: Relative error versus SNR for synthetic regression data set MICI Regression model across five runs.

	SNR value					
	0dB	5dB	10dB	15dB	20dB	25dB
Relative Error	0.691(0.027)	0.443(0.029)	0.266(0.013)	0.173(0.025)	0.101(0.013)	0.061(0.012)
	30dB	35dB	40dB	45dB	50dB	
Relative Error	0.044(0.003)	0.021(0.003)	0.014(0.002)	0.008(0.002)	0.005(0.001)	

6.3.2 Crop Yield Data Set

The crop yield data set from the HARVIST (Heterogeneous Agricultural Research Via Interactive, Scalable Technology) project [55–57] is used to test the performance of MICI Regression model. The data set contains MODIS data observations of corn and wheat yield in the states of California and Kansas over 5 years (2001-2005). There are 100 randomly selected pixels included for each county. The surface reflectance values were reported for each pixel containing 92 values: observations in red for 46 timepoints (every 8 days across the year) and observations in infrared (IR) for 46 timepoints (every 8 days across the year). The zeros and “-32767” reflectance values were indicated as “bad values” in the original data set and are removed for this experiment. Only the counties that reported both corn and wheat yield values are considered and Table 6.19 shows the number of counties that are considered in the states of California and Kansas across the years.

This data set suits the multiple-instance framework well as each county can be naturally

Table 6.19: Number of counties (bags) with both corn and wheat yield in the crop yield data set [57].

Year	Training				Testing
	2001	2002	2003	2004	2005
CA	17	16	18	15	13
KS	100	102	100	102	98

regarded as a bag with multiple data collections (instances). In this experiment, the corn and wheat yield values for each county are regarded as the real-valued regression labels. As the Choquet integral works with values between zero and one, the corn and wheat yield values were normalized between zero and one by Equation (6.8):

$$Y_n = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}, \quad (6.7)$$

where Y_n is the normalized corn or wheat yield value that will be used as the regression training labels. Y_{min} and Y_{max} are the min and max yield value in training, respectively.

Table 6.20: RMSE error for CA corn and wheat yield, Training on Years 2001-2004, Test on Year 2005. The two results with the lowest errors were **bolded** and underlined, respectively. The unit is bushels per acre.

Notes	Regression Methods	Wheat CA	Corn CA
Produces instance-level labels. Fusion source.	Linear Regress	8.316(0.000)	3.172(0.000)
	RVM	<u>3.197(0.000)</u>	0.223(0.000)
	SVR	8.601(0.137)	9.378(0.006)
Fusion methods in comparison.	MICI Regression Fusion	1.572(0.268)	<u>1.992(0.001)</u>
	Another layer of RVM/SVR	3.301(2.497)	4.141(1.246)
	Taking the max	11.917(0.666)	12.721(0.609)
	Taking the min	4.019(0.181)	4.229(0.059)
	Taking the mean	7.079(0.093)	4.808(0.122)
Direct bag-level label prediction.	Cluster MIR	10.168(0.000)	11.185(0.000)
	Aggregate MIR	10.052(0.000)	11.185(0.000)
	RFC-MIR	16.573(0.000)	15.631(0.000)

Linear regression, Relevant Vector Machine (RVM) regression [285, 286], and Support

Table 6.21: RMSE error for KS corn and wheat yield, Training on Year 2001-2004, Test on Year 2005. The two results with the lowest errors were **bolded** and underlined, respectively. The unit is bushels per acre.

Notes	Regression Methods	Wheat KS	Corn KS
Produces instance-level labels. Fusion source.	Linear Regress	2.081(0.000)	16.279(0.000)
	RVM	<u>2.151(0.212)</u>	13.479(0.746)
	SVR	3.349(0.366)	20.151(1.392)
Fusion methods in comparison.	MICI Regression Fusion	1.978(0.001)	<u>15.020(0.003)</u>
	Another layer of RVM/SVR	3.351(1.017)	26.732(10.541)
	Taking the max	2.845(0.164)	34.393(10.831)
	Taking the min	9.041(1.838)	20.767(7.122)
	Taking the mean	3.400(0.191)	20.713(1.477)
Direct bag-level label prediction.	Cluster MIR	8.085(0.000)	31.736(0.000)
	Aggregate MIR	6.866(0.000)	29.973(0.000)
	RFC-MIR	6.550(0.000)	33.195(0.000)

Vector Regression (SVR) [287] were applied to the data. These three regressors operate on all instances and each give a set of instance-based labels (these are essentially Instance-MIR methods with different regressors). We used Gaussian kernel for both RVM and SVR methods. Then, the MICI Regression model is applied to fuse these three regressors and compared to use another layer of RVM and/or SVR (whichever one gives better performance) or simply taking the max, min, or mean of the regression sources as a fusion method. The results are also compared with existing multiple instance regression methods such as the Aggregate MIR, Cluster MIR, and Robust Fuzzy Clustering MIR (RFC-MIR). The detailed description of the comparing MIR methods can be seen in Chapter 2.

The test error was computed by computing the root mean squared error (RMSE) between the predicted county-level (bag-level) yield values for the test year and the (known) actual county-level yield values for the test year, as follows:

$$RMSE = \sqrt{\frac{\sum_{b=1}^B (\hat{y}_b - y_b)^2}{B}}, \quad (6.8)$$

where B is the number of (test) bags, \hat{y}_b is the predicted county-level yield values, and y_b is the (known) actual county-level yield values for the test year.

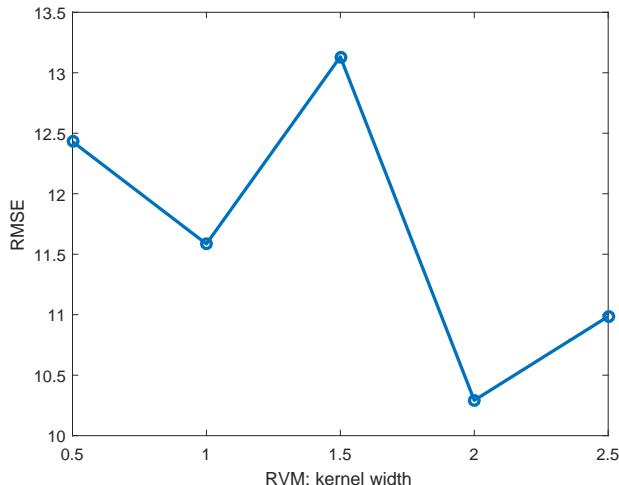


Figure 6.75: The relationship between the Gaussian kernel width and RVM RMSE.

Tables 6.20 and 6.21 present the RMSE prediction error results for corn and wheat yield for the states of California (CA) and Kansas (KS), using crop yield training data from years 2001-2004 and testing on year 2005. The standard deviation is across three runs.

From the table, we can first of all see that all methods are able to predict the yields comparable to results from previous literature such as [55]. The MICI regression produces the lowest or the second lowest error across both states and both crop types. Naturally, the performance of MICI will depend on the performance of input sources (in this case, three instance MIR approaches linear regression, RVM and SVR). The performance of MICI, from Tables 6.20 and 6.21, has surpassed each of its sources as well as other fusion such as taking the max and min. RVM performs well especially for corn, but RVM requires the computation of the kernel matrix, which can be hard when the input data set has higher dimensions. RVM is also highly dependent on the choice of kernel and the parameters

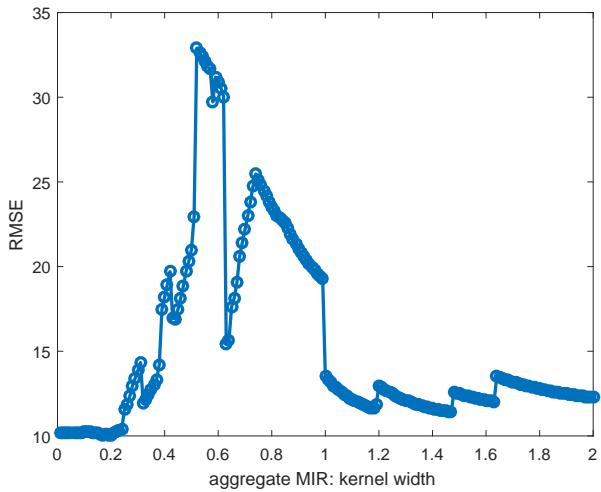


Figure 6.76: The relationship between the Gaussian kernel width and Aggregate MIR RMSE.

such as the kernel width. Figure 6.75 and Figure 6.76 shows the RVM and Aggregate MIR (which also uses RVM regressor) results with kernel width ranging from 0.01 to 2 with an increment of 0.01. Tables 6.20 and 6.21 presented the RMSE prediction of RVM with the minimum error across all the kernel width choices, but it is easy to see that the error of RVM method will depend on the choice of kernel width parameters and can go up to quite high if using a less favorable parameter choice. The Cluster MIR method also depends on the number of clusters and the relationship between the cluster number and the Cluster MIR performance can be seen in Figure 6.77. Again, Tables 6.20 and 6.21 presented the lowest error across these cluster number parameter choices.

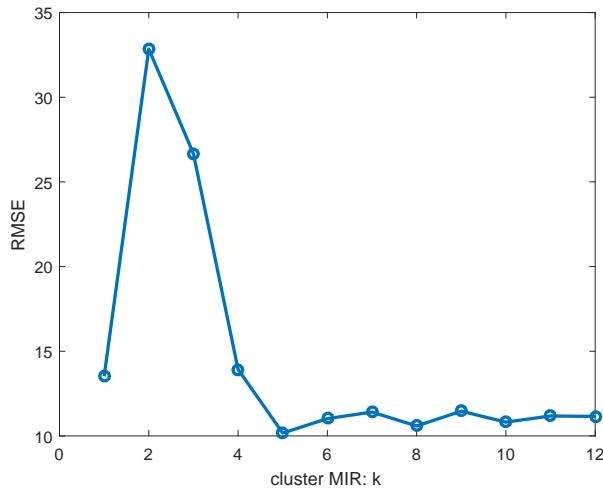


Figure 6.77: The relationship between the cluster number and Cluster MIR RMSE.

6.4 Discussion on Optimization Schemes

A variety of optimization schemes were explored to investigate the effects of optimization schemes on how much objective function fitness and speed of convergence. First, the effects of sampling new measures from “top-down” for initialization versus sampling new measures from “bottom-up” are investigated. Then, during the small-scale and large-scale mutation step of the proposed Multiple Instance Choquet Integral, the performance of sampling according to the size of the valid intervals is compared with sampling according to the number of times a measure element is being used. To make it faster, an alternative approach using binary fuzzy measures was investigated.

6.4.1 “Top-Down” and “Bottom-Up” Initialization

This section investigates the effects of sampling new measures from “top-down” for initialization versus sampling new measures from “bottom-up” as first introduced in Sec-

tion 3.4.1. The two different initialization approaches were applied to the MUUFL Gulfport hyperspectral target detection experiments similar to that in Section 6.1.4. The comparison in the performance and measure values are presented.

Figure 6.78 shows the ROC curves using top-down initialization and bottom-up initialization, training on Gulfport Campus 1 and testing on Gulfport Campus 3. Each initialization was ran five times, with 1000 iterations. As can be seen from the figures, the ROC curves for both initialization methods are almost completely overlapping, with small variations in lower FAR. It can be observed that the ultimate detection performance (after updating the measures) are very similar despite of initialization.

It is interesting to observe the difference of the initial measures generated from these two initialization methods. The “bottom-up” initialization tends to put higher initial values to measure elements from the lower levels of the lattice (as shown in Figure 2.3). Due to monotonicity property of the measure, the “bottom-up” initialization usually forces the measure elements of the upper levels of the lattice to be close to 1. Similarly, while the “top-down” initialization puts lower initial values to measure elements from the upper levels of the lattice, and therefore forcing the measure elements from the lower levels of the lattice to be close to 0. After optimization, the learned measures are very similar and the detection performance measured by ROC curve are also very similar, as shown above.

6.4.2 Sampling according to measure element used

One optimization scheme based on the valid intervals of each measure element was described in Chapter 3 to optimize the proposed Multiple Instance Choquet Integral algorithms. In this section, an alternative optimization scheme is developed and compared with the previously proposed “sampling by the valid interval” scheme.

Similar to the description in Chapter 3, an evolutionary algorithm is used to learn the fuzzy measure for the MICI algorithm. A new measure or a new measure element is updated by large-scale or small-scale mutation. First, for all the instances in the training bags, it is easy to obtain which measure element was used for which instance by sorting all sources (per the definition of Choquet integral). Then, in the small-scale mutation, only one measure element is updated. The element to be updated is chosen by randomly sampling from a multinomial distribution based on the counts of how many times a measure element was used in all the training instances. The probability of sampling a particular measure element g_l is set to

$$P(g_l) = \frac{v_l}{\sum_{o=1}^{2^C-1} v_o}, \quad (6.9)$$

where v_l is the number of times measure element g_l is used in training data. The measure element that was used most frequently by the training data to compute the Choquet integral will have the largest probability to be updated. In the large-scale mutation, all the measure elements are sorted in descending order based on the number of times it was used by the training data and all measure elements are updated according to the sort order. The new measure values are sampled from a truncated Gaussian (TG) distribution, same as previously proposed.

Experiments were conducted based on the MUUFL Gulfport target detection experiment similar to that in Section 6.1.4. The measures were initialized by randomly flipping a coin and select from either top-down and bottom-up initialization methods and then updated using the two different sampling techniques in optimization.

Table 6.22 lists the running times and the number of iterations until convergence for five runs of the two different optimization. The algorithm was considered reaching convergence if the change of fitness is below 10^{-4} . The “measure element” in the table refers

to sampling according to measure element used and “valid interval” refers to sample by sorting the valid intervals of the measure elements. It can be observed from Table 6.22 that the “sampling according to measure element” approach is in general faster in running time and converges in less iterations than the “sort-by-valid-interval” approach, mainly because the valid interval approach has to go through and evaluate the valid interval for all measure elements while the “measure element” approach simply counts for the times a measure element was used in training. Besides, the fitness of the training data depends on the measure elements used, and the measure element that was most frequently used in training is updated most frequently in the “measure element” approach, which encourages the optimization to converge faster.

Table 6.22: Running time (seconds) and number of iterations until convergence for optimization schemes comparison.

	Measure Element		Valid Interval	
	NumIteration	Run Time	NumIteration	Run Time
Run 1	213	79.704s	3488	491.077s
Run 2	61	9.170s	3536	490.432s
Run 3	256	36.970s	2012	280.403s
Run 4	55	49.262s	630	88.477s
Run 5	335	47.479s	3231	448.018s
Summary		44.517(25.376)s		359.735(174.538)s

Figure 6.79 shows the comparison of the new “sampling according to measure element used” scheme compared with the previous “sort-by-valid-interval” scheme. Figure 6.79a shows the fitness value and the speed of convergence in the first 1000 iterations over five runs by the two optimization schemes. It can be observed that the fitness values for both schemes increased rapidly in the first few hundred iterations, and then increased slowly towards convergence. The “sampling according to measure element used” scheme converges faster than the “sort-by-valid-interval” scheme. The ROC curve performance, as shown in

Figure 6.79b, are very similar with very small variations.

6.4.3 Using a binary fuzzy measure

The measures learned by the MICI algorithm for a binary classification problem seem to have a tendency to take values close to $\{0, 1\}$ rather than values in $(0, 1)$, as presented in Table 6.1. Therefore, it is natural to investigate whether it may be more efficient and effective to use a binary fuzzy measure directly, specifically for the binary classification problems.

The binary fuzzy measure (BFM) was explored previously in [288, 289]. It is a variant on the standard fuzzy measure in that the standard fuzzy measure element takes value in $[0, 1]$ while the binary fuzzy measure elements only take values in $\{0, 1\}$. As discussed in [288], the binary fuzzy measure only needs to optimize the space of $\{0, 1\}^{2^C}$ rather than $[0, 1]^{2^C}$, which will be significantly more efficient especially when the number of sources C takes a very large number. Besides, rather than going through the complicated sampling techniques of obtaining a new measure element, the binary fuzzy measure only needs to determine which element(s) has(have) the “first encounter” of the “1” value and the remainder of the measure can be deduced based on the monotonicity property.

Figure 6.80 shows an illustration of a binary fuzzy measure for 4 sources. The red arrows show one path to “climb up the lattice” of the fuzzy measure elements. This figure is similar to Figure 2.3. Suppose we learned that the “first encounter of 1” is g_{23} on the $g_2 \rightarrow g_{23} \rightarrow g_{123} \rightarrow g_{1234}$ path (marked in red). Its subset, $g_2 \equiv 0$ as g_{23} is the first 1 on the path and g_2 can only take the smaller value, which is 0. On the other hand, the elements g_{123} and g_{1234} are in the superset and will have to take the value 1 as it can only be larger or equal to g_{23} , according to the monotonicity property. From this example we can see that it

is only necessary to learn which measure elements first takes the value 1 and the remainder of the measure can be deduced.

Recall the definition of the discrete Choquet integral in Chapter 2. The discrete Choquet integral on instance \mathbf{x}_n given C is then computed as [10, 11, 80]:

$$C_g(\mathbf{x}_n) = \sum_{k=1}^m [h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)] g(A_k), \quad (6.10)$$

where C is sorted so that $h(c_1; \mathbf{x}_n) \geq h(c_2; \mathbf{x}_n) \geq \dots \geq h(c_m; \mathbf{x}_n)$. Since there are only m sources, $h(c_{m+1}; \mathbf{x}_n)$ is defined to be zero. The fuzzy measure element value corresponding to the subset $A_k = \{c_1, \dots, c_k\}$ is $g(A_k)$.

The above definition can be rewritten as [11]

$$\begin{aligned} C_g(\mathbf{x}_n) &= \sum_{k=1}^m [h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)] g(A_k) \\ &= \sum_{k=1}^m h(c_k; \mathbf{x}_n) [g(A_k) - g(A_{k-1})], \end{aligned} \quad (6.11)$$

where $g(A_0)$ is defined to be zero.

Denote k_0 as the (sorted) index for “the first encounter of 1” on a path in the lattice of binary fuzzy measures, i.e. $g(A_{k_0}) = 1$, $g(A_k) = 0$ for all $A(k) \subseteq A(k_0)$ and $g(A_k) = 1$ for all $A(k) \supseteq A(k_0)$ (monotonicity property). The Choquet integral of data point \mathbf{x}_n can be computed as:

$$\begin{aligned} C_g(\mathbf{x}_n) &= \sum_{k=1}^{k_0-1} h(c_k; \mathbf{x}_n) [g(A_k) - g(A_{k-1})] \\ &= \sum_{k=1}^{k_0-1} h(c_k; \mathbf{x}_n) [g(A_k) - g(A_{k-1})] \\ &\quad + h(c_{k_0}; \mathbf{x}_n) [g(A_{k_0}) - g(A_{k_0-1})] \\ &\quad + \sum_{k=k_0+1}^m h(c_k; \mathbf{x}_n) [g(A_k) - g(A_{k-1})]. \end{aligned} \quad (6.12)$$

Under the assumption, $[g(A_k) - g(A_{k-1})] = [0-0] = 0$ for $k \in [1, k_0-1]$ and $[g(A_k) - g(A_{k-1})] = [1-1] = 0$ for $k \in [k_0 + 1, m]$. Therefore, the first and third term in Equation 6.12 equal to zero and the result CI is:

$$C_g(\mathbf{x}_n) = h(c_{k_0}; \mathbf{x}_n) [g(A_{k_0}) - g(A_{k_0-1})] = h(c_{k_0}; \mathbf{x}_n) [1 - 0] = h(c_{k_0}; \mathbf{x}_n). \quad (6.13)$$

In our example of the red path in Figure 6.80, the Choquet integral of \mathbf{x}_n would be equal to the value from the 3rd source (before sorting the sources). This shows that for binary fuzzy measure, the Choquet integral value is equal to the (sorted) k_o^{th} source value.

In binary classification case where the labels are “+1” for positive class (targets) and “0” for negative class (non-targets), the (sorted) k_o^{th} source value must be equal to 1 in order for the estimated label (computed CI) value to be 1.

Then, in this case, we may simply inspect and check the difference between the original (sorted) source value against 1 and learn a binary fuzzy measure.

The binary fuzzy measure is applicable if the label values of a point is associated with either no contribution (measure element value equals 0) or equal worth (measure element value equals 1) to subsets of sources.

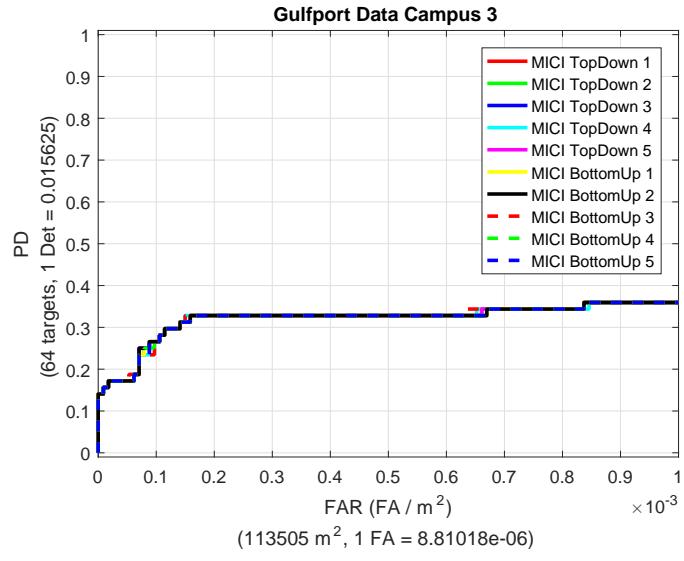
Experiments using the binary fuzzy measures were conducted on MUUFL Gulfport target detection, training on Campus 1 and testing on Gulfport Campus 3, with five runs for each MICI models, similar to Section 6.1.4. The measure values learned by the binary fuzzy measure method are all “1”’s, which is equivalent to the *max* operation. The result fits the desired output of detecting all target types. The classification accuracy performance of the binary fuzzy measure is exactly the same as the performance of the max operation in Table 6.7 and the performance between the standard and binary fuzzy measures are very similar.

For binary fuzzy measures, the number of parameters to be optimized is no longer exponential to the number of sources m , but rather linear ($O(m)$) [289]. For MICI models, the computation complexity of computing the CI in the objective functions using binary fuzzy measure will be $O(INBm)$ given I iterations across B bags and N data points. For MR-MICI model, the computation complexity for computing the CI values will be $O(IN_hBm)$, where N_h is the total number of combinations that the sources can make (so that we need to compute CI for N_h times).

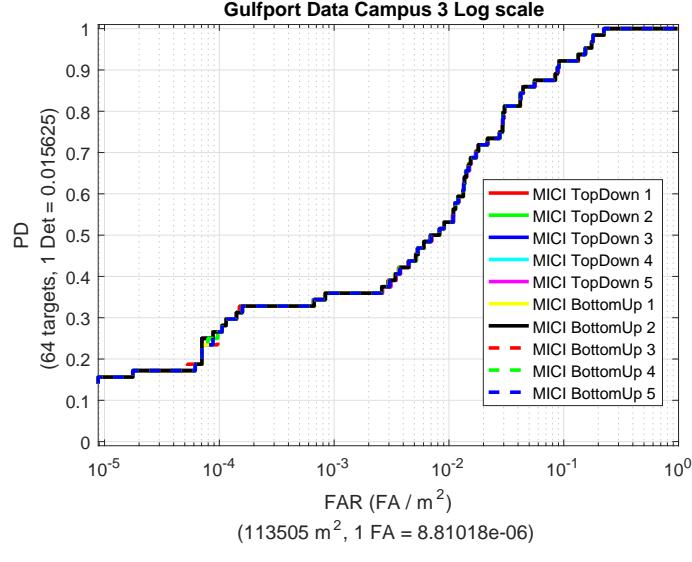
Compared with the running times of MICI models with standard (non-binary) fuzzy measures reported in Table 6.11, the running time of MICI models with binary measures are much shorter as shown in Table 6.23. The maximum iteration number before convergence is enforced to be 100 iterations in this experiment to ensure the fitness does not change for an extended number of iterations, and the binary measure algorithms can in fact reach the optimal solution in even shorter amount of time (sometimes even during initialization). Figure 6.81 shows the comparison of fitness values in MICI models with standard fuzzy measures and with binary measures. The binary fuzzy measure algorithms can reach the optimal fitness almost immediately after initialization and remain the same for the remainder of the iterations. The standard fuzzy measure algorithms rapidly increase in the first few tens of iterations and increase more slowly later, but eventually the fitness reaches convergence (the first 150 iterations are shown in Figure 6.81b). The MICI Noisy-Or model with standard fuzzy measures, in particular, takes much longer to converge overall.

Table 6.23: Running time (seconds) and number of iterations until convergence for MICI models comparison with binary measures.

	MICI noisy-or Binary		MICI min-max Binary		MICI generalized mean Binary	
	NumIteration	Run Time	NumIteration	Run Time	NumIteration	Run Time
Run 1	100	6.746s	100	6.771s	100	10.162s
Run 2	100	6.854s	100	6.327s	100	9.582s
Run 3	100	6.754s	100	6.451s	100	9.355s
Run 4	100	6.897s	100	6.256s	100	9.324s
Run 5	100	6.964s	100	6.385s	100	9.340s
Summary		6.823(0.069)s		6.438(0.200)s		9.553(0.357)s

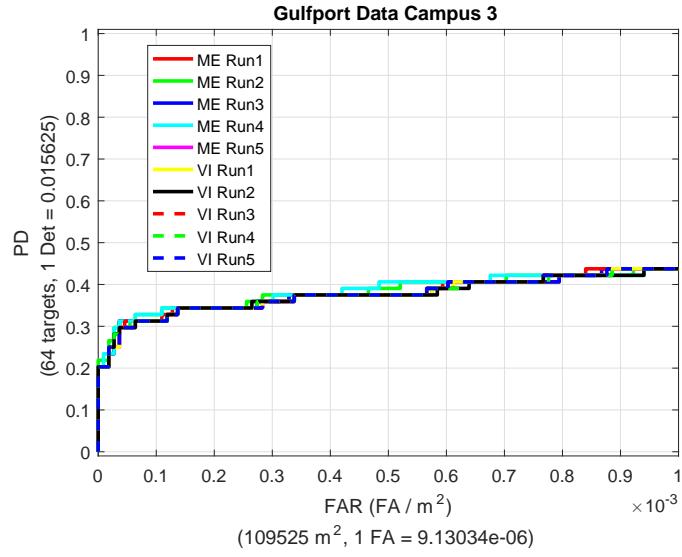


(a)

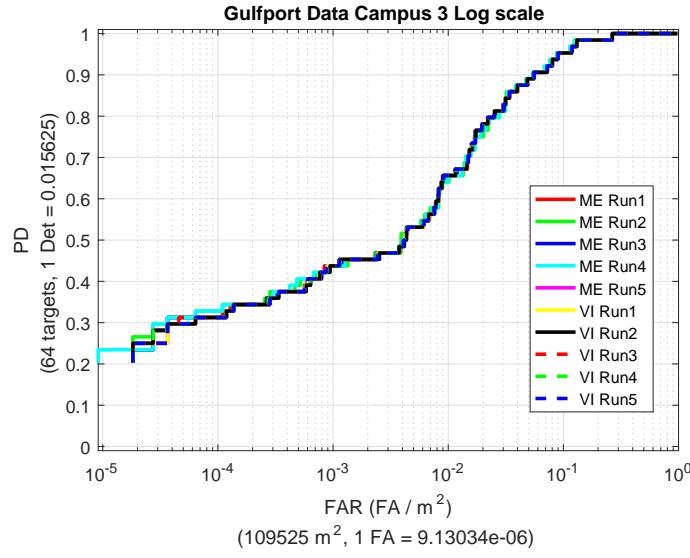


(b)

Figure 6.78: Comparison of ROC curve performance using top-down initialization and bottom-up initialization, testing on Gulfport Campus 3, with five runs for each initialization methods. (a) ROC curve with FAR between 0 and 0.001; (b) ROC curve on log scale with FAR between 0 and 1.



(a)



(b)

Figure 6.79: Comparison of the two optimization schemes: sampling by measure element or sampling according to the valid intervals. Experiments are on MUUFL Gulfport target detection, training on Campus 1 and testing on Gulfport Campus 3, with five runs for each optimization methods. (a) Fitness values in the first 1000 iteration; (b) ROC curve with FAR between 0 and 0.001.

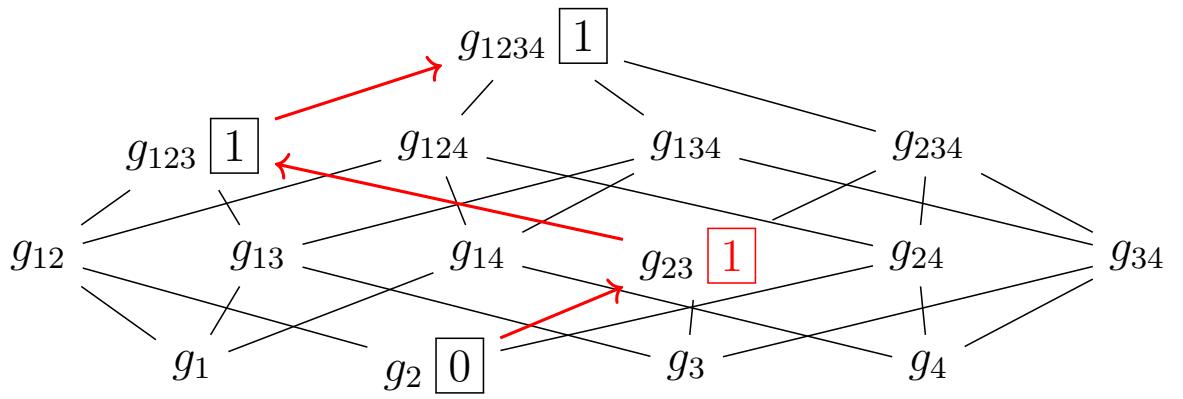
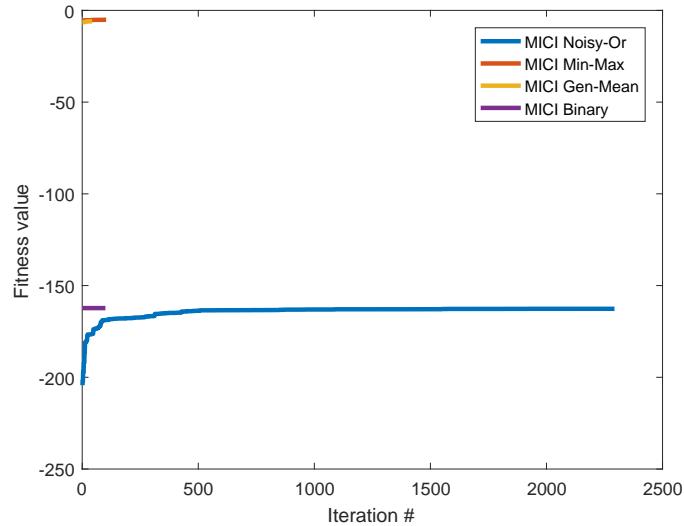
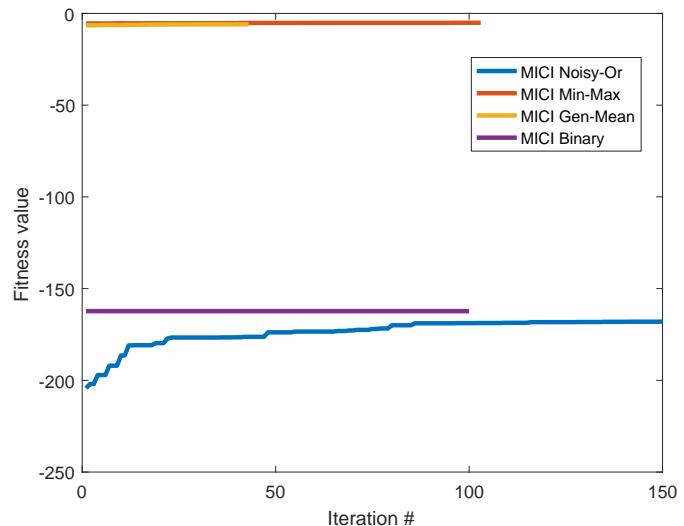


Figure 6.80: An illustration for the subset and superset relationships between fuzzy measure elements given four sources. The red arrows describe one path for “climbing up the lattice”. Suppose $g_{23} = 1$ (marked in red box) and it can be deduced that $g_2 = 0$ and $g_{123} = g_{1234} = 1$.



(a)



(b)

Figure 6.81: Relationship of fitness values vs. number of iterations. The fitness values are shown for MICI noisy-or, min-max and generalized mean models with standard fuzzy measures and MICI models with binary fuzzy measures. Experiments are on MUUFL Gulfport target detection, training on Campus 1 and testing on Gulfport Campus 3. (a) Overall fitness; (b) The first 150 iterations.

Chapter 7

Conclusion

This dissertation proposes a Multiple Instance Choquet Integral (MICI) framework and a Multi-Resolution Multiple Instance Choquet Integral (MR-MICI) framework for multi-sensor fusion. The proposed MICI framework is applicable for classifier fusion and regression with uncertain/imprecise data. The proposed noisy-or, min-max, and generalized-mean models showed successful performance in applications such as target detection. The MR-MICI framework can successfully perform classifier fusion and yield better classification accuracy when compared with traditional approaches on heterogeneous data (multi-resolution data and data of different geospatial types such as HSI and LiDAR) while taking into account uncertain/imprecise labels.

This research has the following contributions:

1. Three variations of the MICI models, i.e. noisy-or, min-max and generalized-mean models, were derived for the MICI classifier fusion framework. This work was extended to regression applications and was able to predict real-valued labels.
2. A novel MR-MICI model was proposed to perform multi-resolution multisensor fu-

sion. Experiments on real data show improved performance against fusion using rasterized data. The proposed model is able to handle mixed geospatial data types (such as HSI and LiDAR) and multi-resolution data, as well as dealing with uncertainty in the training labels. The MUUFL Gulfport data experiment, for example, shows that it is possible to use crowd-sourced data such as OpenStreetMap to guide automatic (uncertain/imprecise) labeling in the supervised training stage and achieve a good classification/detection performance based on such training data.

3. A variety of sampling schemes were investigated for optimizing the proposed algorithms and the binary fuzzy measure was discussed as an alternative approach to improve speed.

4. A complete ground truth label map was created for all materials in the MUUFL Gulfport data set [284], which may benefit future researchers in the area¹.

It would be interesting to continue investigating alternative optimization schemes, such as message passing. More experiments can be conducted to explore the binary fuzzy measure and possibly improve the speed and/or performance of the proposed algorithms. In MR-MICI, it is possible to explore alternative objective function to minimize the number of combinations needed to compute. Alternative features as fusion sources can also be investigated.

¹Scene labels available at <https://github.com/GatorSense/MUUFLGulfport/tree/master/MUUFLGulfportSceneLabels>.

Appendix A

Truncated Gaussian Sampling Method

This appendix describes how to sample from a Truncated Gaussian distribution given a lower and upper bound.

The probability density function (PDF) f of a Truncated Gaussian distribution with lower bound a and upper bound b is defined as:

$$f(x; \mu, \sigma, a, b) = \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}, \quad (\text{A.1})$$

for $a \leq x \leq b$, and $f = 0$ otherwise. μ is the mean and σ is the standard deviation of the Truncated Gaussian distribution. Here, $\phi(\zeta) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}\zeta^2)$ is the probability density function of the standard normal distribution and $\Phi(\zeta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\zeta} e^{-t^2/2} dt = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{\zeta}{\sqrt{2}}\right) \right]$ is the cumulative distribution function of the standard normal distribution. $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ is the "error function" encountered in integrating the normal distribution.

The cumulative distribution function (CDF) of a generic normal distribution with mean μ and standard deviation σ can be computed as $\Phi\left(\frac{\zeta-\mu}{\sigma}\right) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{\zeta-\mu}{\sigma\sqrt{2}}\right) \right]$.

The cumulative density function (CDF) $F_X(x)$ of a continuous random variable X can be expressed as the integral of its probability density function $f_X(x)$ as $F_X(x) = \int_{-\infty}^x f_X(t)dt$. Therefore, the CDF of a Truncated Gaussian distribution can be computed as follows:

$$F(x; \mu, \sigma, a, b) = \int_{-\infty}^x f(t; \mu, \sigma, a, b) dt \quad (\text{A.2})$$

$$= \int_a^x \frac{\phi\left(\frac{t-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} dt \quad (\text{A.3})$$

$$= \frac{\int_a^x \phi\left(\frac{t-\mu}{\sigma}\right) dt}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} \quad (\text{A.4})$$

$$= \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}, \quad (\text{A.5})$$

where $\phi(\zeta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\zeta^2\right)$ is the probability density function of the standard normal distribution and $\Phi(\zeta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\zeta} e^{-t^2/2} dt = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{\zeta}{\sqrt{2}}\right) \right]$ is the cumulative distribution function of the standard normal distribution. $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ is the "error function" encountered in integrating the normal distribution.

The following steps show the solution for x value given a cumulative density function value $F(x; \mu, \sigma, a, b)$:

$$F(x; \mu, \sigma, a, b) = \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} \quad (\text{A.6})$$

$$\begin{aligned} \Phi\left(\frac{x-\mu}{\sigma}\right) &= F(x; \mu, \sigma, a, b) * \left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right) \right] \\ &\quad + \Phi\left(\frac{a-\mu}{\sigma}\right) \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x - \mu}{\sigma \sqrt{2}} \right) \right] &= F(x; \mu, \sigma, a, b) * \left[\Phi \left(\frac{b - \mu}{\sigma} \right) - \Phi \left(\frac{a - \mu}{\sigma} \right) \right] \\ &\quad + \Phi \left(\frac{a - \mu}{\sigma} \right) \\ \operatorname{erf} \left(\frac{x - \mu}{\sigma \sqrt{2}} \right) &= 2 * \left\{ F(x; \mu, \sigma, a, b) * \left[\Phi \left(\frac{b - \mu}{\sigma} \right) - \Phi \left(\frac{a - \mu}{\sigma} \right) \right] + \Phi \left(\frac{a - \mu}{\sigma} \right) \right\} \\ &\quad - 1. \end{aligned} \tag{A.8}$$

(A.9)

Define the right hand side term as $RHTerm$ where

$$RHTerm = 2 * \left\{ F(x; \mu, \sigma, a, b) * \left[\Phi \left(\frac{b - \mu}{\sigma} \right) - \Phi \left(\frac{a - \mu}{\sigma} \right) \right] + \Phi \left(\frac{a - \mu}{\sigma} \right) \right\} - 1. \tag{A.10}$$

$RHTerm$ should be a constant and can be easily computed given a , b , μ , σ and $F(x; \mu, \sigma, a, b)$. Then, x can be solved using the following equation:

$$x = \sigma \sqrt{2} * \operatorname{erf}^{-1}(RHTerm) + \mu. \tag{A.11}$$

erf^{-1} can be computed using the MATLAB *inverf()* function.

Therefore, given a random number $F(x; \mu, \sigma, a, b)$ between $[0, 1]$, its corresponding sample is the x value computed above.

Bibliography

- [1] Jay K. Hackett and Mubarak Shah. Multi-sensor fusion: a perspective. In *IEEE Int. Conf. Robotics and Automation*, volume 2, pages 1324–1330, 1990.
- [2] Jixian Zhang. Multi-source remote sensing data fusion: status and trends. *Int. J. Image and Data Fusion*, 1(1):5–24, 2010.
- [3] Paul Gader, Andres Mendez-Vasquez, Kenneth Chamberlin, Jeremy Bolton, and Alina Zare. Multi-sensor and algorithm fusion with the choquet integral: applications to landmine detection. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, volume 3, pages 1605–1608, Sept. 2004.
- [4] C. Pohl and J. L. Van Genderen. Multisensor image fusion in remote sensing: Concepts, methods and applications. *Int. J. Remote Sens.*, 19(5):823–854, 1998.
- [5] Martin Liggins II, David Hall, and James Llinas. *Handbook of multisensor data fusion*. CRC Press, 2008.
- [6] Jose M. Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.*, 1(2):6–36, 2013.

- [7] Jean-Luc Marichal. An axiomatic approach of the discrete choquet integral as a tool to aggregate interacting criteria. *IEEE Trans. Fuzzy Systems*, 8(6):800–807, Dec 2000.
- [8] Christophe Labreuche and Michel Grabisch. The choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets and Systems*, 137(1):11 – 26, 2003.
- [9] Andres Mendez-Vazquez, Paul D. Gader, James M. Keller, and Kenneth Chamberlin. Minimum classification error training for choquet integrals with applications to landmine detection. *IEEE Trans. Fuzzy Systems*, 16(1):225–238, Feb 2008.
- [10] Xiaoxiao Du, Alina Zare, James Keller, and Derek Anderson. Multiple instance choquet integral for classifier fusion. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1054–1061, Vancouver, BC, July 2016.
- [11] James M. Keller, Derong Liu, and David B. Fogel. *Fundamentals of computational intelligence: Neural networks, fuzzy systems and evolutionary computation*. IEEE Press Series on Computational Intelligence. John Wiley & Sons, Inc., 1st edition, 2016.
- [12] Janos Fodor, Jean-Luc Marichal, and Marc Roubens. Characterization of the ordered weighted averaging operators. *IEEE Trans. Fuzzy Systems*, 3(2):236–240, May 1995.
- [13] Oded Maron. *Learning from Ambiguity*. Ai technical report 1639, Massachusetts Institute of Technology, 1998.

- [14] Gustave Choquet. Theory of capacities. In *Annales de l'institut Fourier*, volume 5, pages 131–295, 1954.
- [15] Matthias Butenuth, Guido v. Gösseln, Michael Tiedge, Christian Heipke, Udo Lipeck, and Monika Sester. Integration of heterogeneous geospatial data in a federated database. *{ISPRS} J. Photogrammetry and Remote Sensing*, 62(5):328 – 346, 2007.
- [16] No-Wook Park, Wooil M. Moon, Kwang-Hoon Chi, and Byung-Doo Kwon. Multi-sensor data fusion for supervised land-cover classification using bayesian and geo-statistical techniques. *Geosciences Journal*, 6(3):193, 2002.
- [17] Huanfeng Shen, Xiangchao Meng, and Liangpei Zhang. An integrated framework for the spatio-temporal-spectral fusion of remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 54(12):7135–7148, 2016.
- [18] Guillaume Brigot, Elise Colin-Koeniguer, Aurélien Plyer, and Fabrice Janez. Adaptation and evaluation of an optical flow method applied to coregistration of forest remote sensing images. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 9(7):2923–2939, 2016.
- [19] Alireza Safdarinezhad, Mehdi Mokhtarzade, and Mohammad Javad Valadan Zoej. Coregistration of satellite images and airborne lidar data through the automatic bias reduction of rpcs. *IEEE J. Sel. Topics. Appl. Earth Observ.*, PP(99):1–14, 2016.
- [20] John A. Hackwell, David W. Warren, Robert P. Bongiovi, Steven J. Hansel, Thomas L. Hayhurst, Dan J. Mabry, Mazaher G. Sivjee, and James W. Skinner. Lwir/mwir imaging hyperspectral sensor for airborne and ground-based remote sens-

- ing. In *Proc. SPIE 2819, Imaging Spectrometry II*, volume 2819, pages 102–107, 1996.
- [21] Yangming Li and Edwin B. Olson. Extracting general-purpose features from lidar data. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 1388–1393, May 2010.
- [22] Aparajithan Sampath and Jie Shan. Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Trans. Geosci. Remote Sens.*, 48(3):1554–1567, Mar. 2010.
- [23] Sen Cao, Xiufang Zhu, Yaozhong Pan, and Qiuyan Yu. A stable land cover patches method for automatic registration of multitemporal remote sensing images. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 7(8):3502–3512, Aug 2014.
- [24] Michael G. Wing, Aaron Eklund, and Loren D. Kellogg. Consumer-grade global positioning system (gps) accuracy and reliability. *J. Forestry*, 103(4):169–173, 2005.
- [25] Soumya Ray and David Page. Multiple instance regression. In *Proc. 18th Int. Conf. Mach. Learn.(ICML)*, volume 1, pages 425–432, 2001.
- [26] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, Jan. 1997.
- [27] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(01):1–25, 2010.
- [28] Nils Weidmann, Eibe Frank, and Bernhard Pfahringer. A two-level learning method

- for generalized multi-instance problems. In *European Conf. Machine Learn.*, pages 468–479. Springer, 2003.
- [29] Nils Weidmann. Two-level classification for generalized multi-instance data. Master’s thesis, Albert Ludwigs University of Freiburg, 2003.
- [30] Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.
- [31] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.
- [32] Qi Zhang and Sally A Goldman. Em-dd: An improved multiple-instance learning technique. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1073–1080, 2001.
- [33] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [34] Oded Maron and Tomas Lozano-Perez. A framework for multiple-instance learning. In *Neural Inform. Process. Syst.*, volume 10, 1998.
- [35] Jun Wang. Solving the multiple-instance problem: A lazy learning approach. In *Proc. 17th Int. Conf. Mach. Learn.*, pages 1119–1125. Morgan Kaufmann, 2000.
- [36] Daniel T Larose. k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, pages 90–106, 2005.
- [37] Liangxiao Jiang, Zhihua Cai, Dianhong Wang, and Harry Zhang. Bayesian citation-knn with distance weighting. *Int. J. Machine Learning and Cybernetics*, 5(2):193–199, 2014.

- [38] Dip Ghosh and Sanghamitra Bandyopadhyay. A fuzzy citation-knn algorithm for multiple instance learning. In *IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2015.
- [39] Pedro Villar, Rosana Montes, Ana María Sánchez, and Francisco Herrera. Fuzzy-citation-knn: a fuzzy nearest neighbor approach for multi-instance classification. In *IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, pages 946–952. IEEE, 2016.
- [40] S. Andrews. Support vector machines for mulitple-instance learning. In *Ann. Conf. Neural Inf. Proc. Systems (NIPS)*, 2002.
- [41] Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1931–1947, Dec. 2006.
- [42] James Foulds and Eibe Frank. Revisiting multiple-instance learning via embedded instance selection. In *Australasian Joint Conf. Artif. Intell.*, pages 300–310. Springer, 2008.
- [43] Xiaoxiao Du, Alina Zare, and James T. Cobb. Possibilistic context identification for sas imagery. In *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, volume 9454, May 2015.
- [44] Oded Maron and Aparna Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *Proc. 15th Int. Conf. Mach. Learn.*, pages 341–349, 1998.
- [45] Zhi-Hua Zhou and Min-Ling Zhang. Multi-instance multi-label learning with application to scene classification. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1609–1616, 2006.

- [46] Saad Ali and Mubarak Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):288–303, 2010.
- [47] Cha Zhang, John C Platt, and Paul A Viola. Multiple instance boosting for object detection. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1417–1424, 2005.
- [48] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 983–990. IEEE, 2009.
- [49] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1619–1632, 2011.
- [50] Peter Torrione, Christopher Ratto, and Leslie M Collins. Multiple instance and context dependent learning in hyperspectral data. In *1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2009.
- [51] Michael I Mandel and Daniel PW Ellis. Multiple-instance learning for music information retrieval. In *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 577–582, 2008.
- [52] Jan Schlüter. Learning to pinpoint singing voice from weakly labeled examples. In *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 44–50, 2016.

- [53] Yasser EL-Manzalawy, Drena Dobbs, and Vasant Honavar. Predicting mhc-ii binding affinity using multiple instance regression. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 8(4):1067–1079, July 2011.
- [54] Zhuang Wang, Vladan Radosavljevic, Bo Han, Zoran Obradovic, and Slobodan Vucetic. Aerosol optical depth prediction from satellite observations by multiple instance regression. In *Proc. SIAM Int. Conf. Data Mining*, pages 165–176, 2008.
- [55] Zhuang Wang, Liang Lan, and Slobodan Vucetic. Mixture model for multiple instance regression and applications in remote sensing. *IEEE Trans. Geosci. Remote Sens.*, 50(6):2226–2237, June 2012.
- [56] Kiri L Wagstaff and Terran Lane. Salience assignment for multiple-instance regression. In *Workshop on Constrained Optimization and Structured Output Spaces (ICML)*, June 2007.
- [57] Kiri L Wagstaff, Terran Lane, and Alex Roper. Multiple-instance regression with structured data. In *IEEE Int. Conf. Data Mining Workshops (ICDMW)*, pages 291–300, Dec. 2008.
- [58] Daniel R. Dooly, Qi Zhang, Sally A. Goldman, and Robert A. Amar. Multiple-instance learning of real-valued data. *J. Mach. Learn. Research*, 3, 2002.
- [59] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [60] Sally A. Goldman and Stephen D. Scott. Multiple-instance learning of real-valued geometric patterns. *Annals Math. and Artif. Intell.*, 39:259–290, 2003.

- [61] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, 100(1):78 – 150, 1992.
- [62] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994.
- [63] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.
- [64] Wang Zhi-Gang, Zhao Zeng-Shun, and Zhang Chang-Shui. Online multiple instance regression. *Chin. Phys. B*, 22(9):098702, 2013.
- [65] Daniel R. Dooly, Sally A. Goldman, and Stephen S. Kwek. Real-valued multiple-instance learning with queries. *J. Comp. System Sciences*, 72:1–15, 2006.
- [66] Pak-Ming Cheung and James T. Kwok. A regularization framework for multiple-instance learning. In *Proc. 23rd Int. Conf. Mach. Learn.*, pages 193–200. ACM, 2006.
- [67] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *Proc. 19th Int. Conf. on Mach. Learn.*, pages 179–186. Morgan Kaufmann, 2002.
- [68] S. R. Gunn. Support vector machines for classification and regression. Technical Report ISIS-1-98, Image Speech and Intelligence Systems Research Group, University of Southampton, 1998.
- [69] Mohamed Trabelsi and Hichem Frigui. Robust fuzzy clustering for multiple instance regression. *J. Pattern Recognition*, Submitted.

- [70] Raghuram Krishnapuram and James Keller. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Systems*, 1(2):98–110, 1993.
- [71] Nikolaos Pappas and Andrei Popescu-Belis. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proc. Conf. Empirical Methods In Natural Language Proc.*, pages 455–466, 2014.
- [72] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Augmented multiple instance regression for inferring object contours in bounding boxes. *IEEE Trans. Image Proc.*, 23(4):1722–1736, April 2014.
- [73] Michel Grabisch. The application of fuzzy integrals in multicriteria decision making. *European J. Operational Research*, 89(3):445–456, Mar. 1996.
- [74] Michel Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. In *Int. Joint Conf. 4th IEEE Int. Conf. Fuzzy Systems and 2nd Int. Fuzzy Eng. Symp.*, volume 1, pages 145–150, Mar. 1995.
- [75] Andres Mendez-Vazquez and Paul Gader. Learning fuzzy measure parameters by logistic lasso. In *Proc. Annual Conf. North American Fuzzy Information Processing Society (NAFIPS)*, pages 1–7, May 2008.
- [76] Derek Anderson, James M. Keller, and Timothy C. Havens. Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. In *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 502–511. Springer Berlin Heidelberg, 2010.
- [77] Derek T. Anderson, Stanton R. Price, and Timothy C. Havens. Regularization-based

- learning of the choquet integral. In *IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, pages 2519–2526, July 2014.
- [78] Michio Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.
- [79] Melvin Fitting and Ewa Orlowska, editors. *Beyond Two: Theory and Applications of Multiple-Valued Logic*. Springer, 2003.
- [80] Andres Mendez-Vazquez. *Information Fusion and Sparsity Promotion Using Choquet Integrals*. PhD thesis, University of Florida, 2008.
- [81] Michio Sugeno. Fuzzy measures and fuzzy integrals: A survey. In Madan M. Gupta, G. N. Saridis, and B. R. Gaines, editors, *Fuzzy automata and decision processes*. Elsevier Science Ltd, Amsterdam: North Holland, first edition, 1977.
- [82] Toshiaki Murofushi and Michio Sugeno. Fuzzy measures and fuzzy integrals. *Fuzzy Measures and Integrals: Theory and Applications*, pages 3–41, 2000.
- [83] Toshiaki Murofushi, Michio Sugeno, and Motoya Machida. Non-monotonic fuzzy measures and the choquet integral. *Fuzzy Sets and Systems*, 64(1):73–86, 1994.
- [84] Yasuo Narukawa and Vicenç Torra. Non-monotonic fuzzy measures and intuitionistic fuzzy sets. In *Int. Conf. Modeling Decisions for Artif. Intell.*, pages 150–160, 2006.
- [85] Michel Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy sets and Systems*, 69(3):279–298, 1995.

- [86] Hong Jiang and J Ronald Eastman. Application of fuzzy measures in multi-criteria evaluation in gis. *Int. J. Geographical Info. Sci.*, 14(2):173–184, 2000.
- [87] Kazuki Ishii and Michio Sugeno. A model of human evaluation process using fuzzy measure. *Int. J. Man-Machine Studies*, 22(1):19–38, 1985.
- [88] Divakaran Liginlal and Terence T. Ow. Modeling attitude to risk in human decision processes: an application of fuzzy measures. *Fuzzy Sets and Systems*, 157(23):3040–3054, 2006.
- [89] Tamalika Chandra. *Fuzzy Measures in Image Processing*, pages 587–606. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [90] James M. Keller and Carl L. Carpenter. Image segmentation in the presence of uncertainty. *Int. J. Intelligent Systems*, 5(2):193–208, 1990.
- [91] James Keller, H. Qiu, and H. Tahani. The fuzzy integral in image segmentation. In *Proc. Annual Conf. North American Fuzzy Information Processing Society (NAFIPS)*, pages 324–338, 1986.
- [92] Lan-Zhen Yang, Ming-Hu Ha, Xian-Jie Wang, and Zi-Ru Zhang. A generalized λ -fuzzy measure and its application in image enhancement. In *Int. Conf. Wavelet Analysis and Pattern Recognition*, volume 1, pages 35–40, Aug. 2008.
- [93] Mihail Popescu, James M. Keller, and Joyce A. Mitchell. Gene ontology automatic annotation using a domain based gene product similarity measure. In *14th IEEE Int. Conf. Fuzzy Systems*, pages 108–113, May 2005.
- [94] Mihail Popescu, James M. Keller, and Joyce A. Mitchell. Fuzzy measures on the

- gene ontology for gene product similarity. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 3(3):263–274, July 2006.
- [95] Eulalia Szmidt and Janusz Kacprzyk. A similarity measure for intuitionistic fuzzy sets and its application in supporting medical diagnostic reasoning. In *Int. Conf. Artif. Intell. and Soft Computing*, pages 388–393, 2004.
- [96] Anna Wilbik and James M. Keller. A fuzzy measure similarity between sets of linguistic summaries. *IEEE Trans. Fuzzy Systems*, 21(1):183–189, Feb. 2013.
- [97] Toshiaki Murofushi and Michio Sugeno. A theory of fuzzy measures: Representations, the choquet integral, and null sets. *J. Math. Analysis and Applications*, 159(2):532–549, 1991.
- [98] Sansanee Auephanwiriyakul and James M. Keller. A comparison of the linguistic choquet and sugeno fuzzy integrals. In *10th IEEE Int. Conf. Fuzzy Systems*, volume 1, pages 312–315, 2001.
- [99] Michel Grabisch and Eric Raufaste. An empirical study of statistical properties of the choquet and sugeno integrals. *IEEE Trans. Fuzzy Systems*, 16(4):839–850, Aug. 2008.
- [100] Erich Peter Klement, Radko Mesiar, and Endre Pap. A universal integral as common frame for choquet and sugeno integral. *IEEE Trans. Fuzzy Systems*, 18(1):178–187, Feb. 2010.
- [101] Paul D. Gader, James M. Keller, and Bruce N. Nelson. Recognition technology for the detection of buried land mines. *IEEE Trans. Fuzzy Systems*, 9(1):31–43, .Feb 2001.

- [102] Michel Grabisch, Hung T. Nguyen, and Elbert A. Walker. *Fundamentals of uncertainty calculi with applications to fuzzy inference*, volume 30. Springer Science & Business Media, 2013.
- [103] James M. Keller, Sansanee Auephanwiriyakul, and Paul D. Gader. Experiments in predictive sensor fusion. In *Proc. SPIE, Detection and Remediation Technologies for Mines and Minelike Targets VI*, volume 4394, pages 1047–1058, Oct. 2001.
- [104] Liang Liu, Mihail Popescu, Marilyn Rantz, and Marjorie Skubic. Fall detection using doppler radar and classifier fusion. In *Proc. IEEE-EMBS Int. Conf. Biomedical and Health Informatics*, pages 180–183, Jan. 2012.
- [105] Qinghua Wang, Canliang Zheng, Hongtao Yu, and Donghua Deng. Integration of heterogeneous classifiers based on choquet fuzzy integral. In *7th Int. Conf. Intelligent Human-Machine Systems and Cybernetics*, volume 1, pages 543–547, Aug. 2015.
- [106] Derek T. Anderson, Paul Elmore, Fred Petry, and Timothy C. Havens. Fuzzy choquet integration of homogeneous possibility and probability distributions. *Inf. Sci.*, 363(C):24–39, Oct. 2016.
- [107] Michel Grabisch. Fuzzy measures and integrals: recent developments. In *Fifty Years of Fuzzy Logic and its Applications*, pages 125–151. Springer, 2015.
- [108] Lequn Hu, Derek T. Anderson, Timothy C. Havens, and James M. Keller. Efficient and scalable nonlinear multiple kernel aggregation using the choquet integral. In *Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 206–215, 2014.

- [109] Anthony Pinar, Timothy C. Havens, Derek T. Anderson, and Lequn Hu. Feature and decision level fusion using multiple kernel learning and fuzzy integrals. In *IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, Aug. 2015.
- [110] Paul D Gader, Wen-Hsiung Lee, and Andres Mendez-Vasquez. Continuous choquet integrals with respect to random sets with applications to landmine detection. In *IEEE Int. Conf. Fuzzy Systems*, volume 1, pages 523–528, 2004.
- [111] Paul D Gader, Magdi A Mohamed, and James M. Keller. Dynamic-programming-based handwritten word recognition using the choquet fuzzy integral as the match function. *J. Electronic Imaging*, 5(1):15–24, 1996.
- [112] Wen-Tsong Chen and Paul Gader. Word level discriminative training for handwritten word recognition. In *Proc. 7th IWFHR*, pages 393–402, 2000.
- [113] Shuchao Feng, Wenqian Shang, and Yuqi Wang. A k-highest expert text classification algorithm based on choquet integral. In *3rd Int. Conf. Applied Computing and Information Technology/2nd Int. Conf. Computational Science and Intelligence (ACIT-CSI)*, pages 499–503, July 2015.
- [114] K. Hirota, H. A. Vu, P. Q. Le, C. Fatichah, Z. Liu, Y. Tang, M. L. Tangel, Z. Mu, B. Sun, F. Yan, D. Masano, O. Thet, M. Yamaguchi, F. Dong, and Y. Yamazaki. Multimodal gesture recognition based on choquet integral. In *IEEE Int. Conf. Fuzzy Systems (FUZZ)*, pages 772–776, June 2011.
- [115] Rong Yang, Zhenyuan Wang, Pheng-Ann Heng, and Kwong-Sak Leung. Fuzzified choquet integral with a fuzzy-valued integrand and its application on tempera-

- ture prediction. *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):367–380, 2008.
- [116] Ronald Stanley, James M. Keller, Charles W. Caldwell, and Paul Gader. Abnormal cell detection using the choquet integral. In *Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf.*, volume 2, pages 1134–1139, July 2001.
- [117] Ya-Li Hsiao, Hsiang-Chuan Liu, Po-Fon Chen, Pei-Chun Chang, and Cheng-Fang Tsai. Choquet integral algorithm for t-cell epitope prediction based on fuzzy measure. In *8th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 3, pages 1588–1591, July 2011.
- [118] M Grabisch and M Schmitt. Mathematical morphology, order filters and fuzzy logic. In *Proc. 4th IEEE Int. Conf. Fuzzy Systems and 2nd Int. Fuzzy Eng. Symp.*, volume 4, pages 2103–2108. IEEE, 1995.
- [119] James M Keller, Paul Gader, Raghu Krishnapuram, Xiaomei Wang, A Koksal Hocaoglu, Hichem Frigui, and Jerry Moore. A fuzzy logic automatic target detection system for ladar range images. In *IEEE World Congress on Computational Intelligence (WCCI)*, volume 1, pages 71–76. IEEE, 1998.
- [120] A Koksal Hocaoglu and Paul D Gader. An interpretation of discrete choquet integrals in morphological image processing. In *12th IEEE Int. Conf. Fuzzy Systems*, volume 2, pages 1291–1295, 2003.
- [121] Jeremy Bolton, Paul Gader, and Joseph N. Wilson. Discrete choquet integral as a distance metric. *IEEE Trans. Fuzzy Systems*, 16(4):1107–1110, Aug. 2008.

- [122] Timothy C. Havens, Derek T. Anderson, Christian Wagner, Hanieh Deilamsalehy, and Dereck Wonnacott. Fuzzy integrals of crowd-sourced intervals using a measure of generalized accord. In *IEEE Int. Conf. Fuzzy Systems (FUZZ)*, pages 1–8, July 2013.
- [123] Michel Grabisch, Ivan Kojadinovic, and Patrick Meyer. A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the kappalab r package. *European J. Operational Research*, 186(2):766–785, 2008.
- [124] Michel Grabisch and Jean-Marie Nicolas. Classification by fuzzy integral: performance and tests. *Fuzzy sets and systems*, 65(2-3):255–271, 1994.
- [125] Jean-Luc Marichal and Marc Roubens. Determination of weights of interacting criteria from a reference set. *European J. Operational Research*, 124(3):641 – 650, 2000.
- [126] Ivan Kojadinovic, Jean-Luc Marichal, and Marc Roubens. An axiomatic approach to the definition of the entropy of a discrete choquet capacity. *Information Sciences*, 172(1):131–153, 2005.
- [127] Ivan Kojadinovic. Minimum variance capacity identification. *European J. Operational Research*, 177(1):498–514, 2007.
- [128] Patrick Meyer and Marc Roubens. Choice, ranking and sorting in fuzzy multiple criteria decision aid. In *Multiple criteria decision analysis: State of the art surveys*, pages 471–503. Springer, 2005.

- [129] James M. Keller and Jeffrey Osborn. Training the fuzzy integral. *Int. J. Approximate Reasoning*, 15(1):1–24, 1996.
- [130] Jia Wang and Zhenyuan Wang. Using neural networks to determine sugeno measures by statistics. *Neural Networks*, 10(1):183–195, 1997.
- [131] Hassiba Nemmour and Youcef Chibani. Neural network combination by fuzzy integral for robust change detection in remotely sensed imagery. *EURASIP J. Applied Signal Processing*, 2005:2187–2195, 2005.
- [132] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
- [133] Keon-Myung Lee and Hyung Leekwang. Identification of λ -fuzzy measure by genetic algorithms. *Fuzzy Sets and Systems*, 75(3):301 – 309, 1995.
- [134] Dayou Wang, Xiaomei Wang, and James M. Keller. Determining fuzzy integral densities using a genetic algorithm for pattern recognition. In *Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 263–267, 1997.
- [135] Wei Wang, Zhenyuan Wang, and George J Klir. Genetic algorithms for determining fuzzy measures from data. *Journal of Intelligent & Fuzzy Systems*, 6(2):171–183, 1998.
- [136] Zhenyuan Wang, Kwong sak Leung, and Jia Wang. A genetic algorithm for determining nonadditive set functions in information fusion. *Fuzzy Sets and Systems*, 102(3):463 – 469, 1999.

- [137] Ting-Yu Chen, Jih-Chang Wang, and Gwo-Hshiung Tzeng. Identification of general fuzzy measures by genetic algorithms based on partial information. *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30(4):517–528, 2000.
- [138] Ting-Yu Chen and Jih-Chang Wang. Identification of λ -fuzzy measures using sampling design and genetic algorithms. *Fuzzy Sets and Systems*, 123(3):321–341, 2001.
- [139] Zhenyuan Wang and Hai-Feng Guo. A new genetic algorithm for nonlinear multiregressions based on generalized choquet integrals. In *IEEE Int. Conf. Fuzzy Systems*, volume 2, pages 819–821, 2003.
- [140] Elías F Combarro and Pedro Miranda. Identification of fuzzy measures from sample data with genetic algorithms. *Computers & Operations Research*, 33(10):3046–3066, 2006.
- [141] Yi-Chung Hu. Fuzzy integral-based perceptron for two-class pattern classification problems. *Information Sciences*, 177(7):1673 – 1686, 2007.
- [142] James Kennedy and Russell Eberhart. Particle swarm optimization. In *IEEE Int. Conf. Neural Networks*, volume 4, pages 1942–1948, Nov. 1995.
- [143] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.
- [144] Tao Xiao, Ning Zhang, and Don-Mei Huang. Solution to the model of nonlinear multiregression based on generalized choquet integral and its application in estimating forest volume. In *Int. Conf. Computer Application and System Modeling (ICCASM)*, volume 12, pages V12–161–V12–165, Oct 2010.

- [145] James Bezdek, James Keller, Raghu Krisnapuram, and Nikhil Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer, 1999.
- [146] Dymitr Ruta and Bogdan Gabrys. An overview of classifier fusion methods. *Computing and Information systems*, 7(1):1–10, 2000.
- [147] Sansanee Auephanwiriyakul, James Keller, and Paul D Gader. Generalized choquet fuzzy integral fusion. *Information Fusion*, 3(1):69 – 85, 2002.
- [148] B. Waske and J. Benediktsson. Fusion of support vector machines for classification of multisensor data. *IEEE Trans. Geosci. Remote Sens.*, 45(12):3858–3866, Dec. 2007.
- [149] Peijun Du, Wei Zhang, Shubi Zhang, and Junshi Xia. Hyperspectral remote sensing image classification based on decision level fusion. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, volume 4, pages IV–940–IV–943, July 2009.
- [150] He Yang, Qian Du, and Ben Ma. Decision fusion on supervised and unsupervised classifiers for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens. Lett.*, 7(4):875–879, Oct. 2010.
- [151] F. Tabib Mahmoudi, F. Samadzadegan, and P. Reinartz. Object recognition based on the context aware decision-level fusion in multiviews imagery. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 8(1):12–22, Jan 2015.
- [152] James M. Keller, Sansanee Auephanwiriyakul, and Paul D. Gader. New fuzzy set tools to aid in predictive sensor fusion. In *Proc. SPIE, Detection and Remediation Technologies for Mines and Minelike Targets V*, volume 4038, pages 1497–1507, Aug. 2000.

- [153] Ajith H Gunatilaka, Brian Baertlein, et al. Feature-level and decision-level fusion of noncoincidentally sampled sensors for land mine detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):577–589, 2001.
- [154] H. Frigui, Lijun Zhang, and P.D. Gader. Context-dependent multisensor fusion and its application to land mine detection. *IEEE Trans. Geosci. Remote Sens.*, 48(6):2528–2543, June 2010.
- [155] Paul Gader, Magdi Mohamed, and James Keller. Fusion of handwritten word classifiers. *Pattern Recog. Lett.*, 17(6):577–584, 1996.
- [156] Zhang Liangbin, Ma Wenjun, Chen Li, and Zhang Su. Research on detection of prostate cancer mr images based on information fusion. In *12th Int. Conf. Signal Proc. (ICSP)*, pages 1094–1098, Oct 2014.
- [157] S. Khazendar, H. Al-Assam, H. Du, S. Jassim, A. Sayasneh, T. Bourne, J. Kaijser, and D. Timmerman. Automated classification of static ultrasound images of ovarian tumours based on decision level fusion. In *6th Comp. Sci. Electron. Eng. Conf. (CEEC)*, pages 148–153, Sept 2014.
- [158] Chao Zhang, Zhuan-Zhao Yang, Zhen-Bao Liu, and Dong Chen. Research on intelligent fault diagnosis method for complex equipment based on decision-level fusion. In *Int. Conf. Mach. Learn. Cybernetics (ICMLC)*, volume 1, pages 386–390, July 2010.
- [159] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karay, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

- [160] Hannu Huhdanpaa, Darryl H. Hwang, Gregory G. Gasparian, Michael T. Booker, Yong Cen, Alexander Lerner, Orest B. Boyko, John L. Go, Paul E. Kim, Anandh Rajamohan, Meng Law, and Mark S. Shiroishi. Image coregistration: Quantitative processing framework for the assessment of brain lesions. *J. Digital Imaging*, 27(3):369–379, Jun. 2014.
- [161] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.
- [162] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical image analysis*, 2(1):1–36, 1998.
- [163] Juheon Lee, Xiaohao Cai, Carola-Bibiane Schönlieb, and David A Coomes. Non-parametric image registration of airborne lidar, hyperspectral and photographic imagery of wooded landscapes. *IEEE Trans. Geosci. Remote Sens.*, 53(11):6073–6084, 2015.
- [164] Achim Roth, M Huber, and Detlev Kosmann. Geocoding of terrasar-x data. *Proc. Int. 20th ISPRS Congr*, pages 840–844, 2004.
- [165] LMH Ulander, A Gustavsson, B Flood, D Murdin, P Dubois-Fernandez, X Depuis, G Sandberg, MJ Soja, LEB Eriksson, JES Fransson, et al. Biosar 2010: Technical assistance for the development of airborne sar and geophysical measurements during the biosar 2010 experiment: Final report. *Eur. Space Agency (ESA), Paris, France, Tech. Rep. Contract No. 4000102285/10/NL/JA/ef*, 2011.
- [166] Enrica Belluco, Monica Camuffo, Sergio Ferrari, Lorenza Modenese, Sonia Silvestri, Alessandro Marani, and Marco Marani. Mapping salt-marsh vegetation by

- multispectral and hyperspectral remote sensing. *Remote sensing of environment*, 105(1):54–67, 2006.
- [167] Adrian Schubert, David Small, Michael Jehle, and Erich Meier. Cosmo-skymed, terrasar-x, and radarsat-2 geolocation accuracy after compensation for earth-system effects. In *IEEE Int. Geosci. Remote Sens. Symp.*, pages 3301–3304, 2012.
- [168] Yong Sun Kim, Jae Hak Lee, and Jong Beom Ra. Multi-sensor image registration based on intensity and edge orientation information. *Pattern recognition*, 41(11):3356–3365, 2008.
- [169] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.
- [170] Qiaoliang Li, Guoyou Wang, Jianguo Liu, and Shaobo Chen. Robust scale-invariant feature matching for remote sensing image registration. *IEEE Geosci. Remote Sens. Lett.*, 6(2):287–291, 2009.
- [171] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. computer vision*, 60(2):91–110, 2004.
- [172] William K. Pratt. Correlation techniques of image registration. *IEEE Trans. Aerospace and Electronic Systems*, AES-10(3):353–358, 1974.
- [173] Youcef Bentoutou, Nasreddine Taleb, Kidiyo Kpalma, and Joseph Ronsin. An automatic image registration for applications in remote sensing. *IEEE Trans. Geosci. Remote Sens.*, 43(9):2127–2137, Sept. 2005.

- [174] Hua-mei Chen, Manoj K. Arora, and Pramod K. Varshney. Mutual information-based image registration for remote sensing data. *Int. J. Remote Sensing*, 24(18):3701–3706, 2003.
- [175] Jordi Ingla. Similarity measures for multisensor remote sensing images. In *IEEE Int. Geosci. Remote Sens. Symp (IGARSS)*, volume 1, pages 104–106, 2002.
- [176] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Trans. Medical Imaging*, 22(8):986–1004, 2003.
- [177] Jon F Nielsen, Nilesh R Ghugre, and Ashok Panigrahy. Affine and polynomial mutual information coregistration for artifact elimination in diffusion tensor imaging of newborns. *Magnetic resonance imaging*, 22(9):1319–1323, 2004.
- [178] Takashi Yokoi, Tsutomu Soma, Hiroyuki Shinohara, and Hiroshi Matsuda. Accuracy and reproducibility of co-registration techniques based on mutual information and normalized mutual information for mri and spect brain images. *Annals of Nuclear Medicine*, 18(8):659–667, 2004.
- [179] Alexander Leemans, Jan Sijbers, Steve De Backer, Everhard Vandervliet, and Paul M. Parizel. Affine coregistration of diffusion tensor magnetic resonance images using mutual information. In *Int. Conf. Advanced Concepts for Intelligent Vision Systems*, pages 523–530, 2005.
- [180] Wim Van Hecke, Alexander Leemans, Emiliano D’Agostino, Steve De Backer, Evert Vandervliet, Paul M Parizel, and Jan Sijbers. Nonrigid coregistration of diffusion

- tensor images using a viscous fluid model and mutual information. *IEEE Trans. Medical Imaging*, 26(11):1598–1612, 2007.
- [181] Sahil Suri, Peter Reinartz, et al. Mutual-information-based registration of terrasar-x and ikonos imagery in urban areas. *IEEE Trans. Geosci. Remote Sens.*, 48(2):939–949, 2010.
- [182] Suma Dawn, Vikas Saxena, and Bhudev Sharma. Remote sensing image registration techniques: A survey. In *Int. Conf. Image and Signal Process.*, pages 103–112, 2010.
- [183] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
- [184] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [185] Hua-mei Chen and Pramod K Varshney. A pyramid approach for multimodality image registration based on mutual information. In *Proc. 3rd Int. Conf. Information Fusion*, volume 1, pages MOD3–9, 2000.
- [186] Ting Hu, Hongyan Zhang, Huanfeng Shen, and Liangpei Zhang. Robust registration by rank minimization for multiangle hyper/multispectral remotely sensed imagery. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 7(6):2443–2457, June 2014.
- [187] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [188] Lan-Rong Dung, Chang-Min Huang, and Yin-Yi Wu. Implementation of ransac

- algorithm for feature-based image registration. *J. Computer and Communications*, 1(06):46, 2013.
- [189] Zahra Hosseini-nejad and Mehdi Nasri. Image registration based on sift features and adaptive ransac transform. In *Int. Conf. Communication and Signal Processing (ICCSP)*, pages 1087–1091, April 2016.
- [190] Thomas Colleu, Jian-Kun Shen, Bogdan Matuszewski, Lik-Kwan Shark, and Claude Cariou. Feature-based deformable image registration with ransac based search correspondence. In *Atlantic Europe Conf. Remote Imaging and Spectroscopy (AECRIS)*, pages 57–64, 2006.
- [191] Shao-Wen Yang, Chieh-Chih Wang, and Chun-Hua Chang. Ransac matching: Simultaneous registration and segmentation. In *Int. Conf. Robotics and Automation (ICRA)*, pages 1905–1912. IEEE, 2010.
- [192] Jesus Selva and Juan M Lopez-Sanchez. Efficient interpolation of sar images for coregistration in sar interferometry. *IEEE Geosci. remote sens. lett.*, 4(3):411–415, 2007.
- [193] Sbastien Leprince, Sylvain Barbot, Franois Ayoub, and Jean-Philippe Avouac. Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. *IEEE Trans. Geosci. Remote Sens.*, 45(6):1529–1558, 2007.
- [194] Ramon Hanssen and Richard Bamler. Evaluation of interpolation kernels for sar interferometry. *IEEE Trans. Geosci. Remote Sens.*, 37(1):318–321, 1999.

- [195] Maria Teresa Merino and Jorge Nunez. Super-resolution of remotely sensed images with variable-pixel linear reconstruction. *IEEE Trans. Geosci. Remote Sens.*, 45(5):1446–1457, 2007.
- [196] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE trans. acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.
- [197] John Tower Ashburner and KJ Friston. *Computational neuroanatomy*. University of London England, 2001.
- [198] Christophe Accadia, Stefano Mariani, Marco Casaioli, Alfredo Lavagnini, and Antonio Speranza. Sensitivity of precipitation forecast skill scores to bilinear interpolation and a simple nearest-neighbor average method on high-resolution verification grids. *Weather and forecasting*, 18(5):918–932, 2003.
- [199] Henri A Vrooman, Chris A Cocosco, Fedde van der Lijn, Rik Stokking, M Arfan Ikram, Meike W Vernooij, Monique MB Breteler, and Wiro J Niessen. Multi-spectral brain tissue segmentation using automatically trained k-nearest-neighbor classification. *Neuroimage*, 37(1):71–81, 2007.
- [200] Neil A Dodgson. Quadratic interpolation for image resampling. *IEEE trans. image processing*, 6(9):1322–1326, 1997.
- [201] Kazuo Toraichi, Sai Yang, Masaru Kamada, and Ryoichi Mori. Two-dimensional spline interpolation for image reconstruction. *Pattern Recognition*, 21(3):275–284, 1988.
- [202] Claus Lamm, Christian Windischberger, Ulrich Leodolter, Ewald Moser, and Herbert Bauer. Co-registration of eeg and mri data using matching of spline interpo-

- lated and mri-segmented reconstructions of the scalp surface. *Brain Topography*, 14(2):93–100, 2001.
- [203] Thomas Martin Lehmann, Claudia Gonner, and Klaus Spitzer. Survey: Interpolation methods in medical image processing. *IEEE trans. medical imaging*, 18(11):1049–1075, 1999.
- [204] Rolf Scheiber and Alberto Moreira. Coregistration of interferometric sar images using spectral diversity. *IEEE Trans. Geosci. Remote Sens.*, 38(5):2179–2191, 2000.
- [205] Pau Prats, Andreas Reigber, and Jordi J Mallorqui. Interpolation-free coregistration and phase-correction of airborne sar interferograms. *IEEE Geosci. Remote Sens. Lett.*, 1(3):188–191, 2004.
- [206] Alan Saalfeld. Conflation automated map compilation. *Int. J. Geographical Info. Syst.*, 2(3):217–228, 1988.
- [207] Alan John Saalfeld. *Conflation: Automated Map Compilation*. PhD thesis, University of Maryland at College Park, College Park, MD, USA, 1993.
- [208] Boris Kovalerchuk and William Q. Sumner. Algebraic relational approach to conflating images. In *AeroSense*, pages 621–630. International Society for Optics and Photonics, 2003.
- [209] Boris Kovalerchuk, Yuliya Kamatkova, and Peter Doucette. Overcoming the combined effect of geometric distortion and object change in image registration and conflation. In *Defense and Security Symp.*, pages 62331S–62331S. International Society for Optics and Photonics, 2006.

- [210] Boris Kovalerchuk. Affine invariant and robust image registration/conflation algorithm. In *Defense and Security Symp.*, pages 656514–656514. International Society for Optics and Photonics, 2007.
- [211] Ching-Chien Chen and Craig A Knoblock. Conflation of geospatial data. In *Encyclopedia of GIS*, pages 133–140. Springer, 2008.
- [212] Andrew R. Buck, James M. Keller, and Marjorie Skubic. A modified genetic algorithm for matching building sets with the histograms of forces. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7, July 2010.
- [213] Andrew R. Buck, James M. Keller, and Marjorie Skubic. A memetic algorithm for matching spatial configurations with the histograms of forces. *IEEE Trans. Evolutionary Computation*, 17(4):588–604, Aug. 2013.
- [214] Wenbo Song, James M. Keller, Timothy L. Haithcoat, and Curt H. Davis. Automated geospatial conflation of vector road maps to high resolution imagery. *IEEE Trans. Image Processing*, 18(2):388–400, Feb. 2009.
- [215] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.
- [216] Manfred Ehlers, Sascha Klonus, Pär Johan Åstrand, and Pablo Rosso. Multi-sensor image fusion for pansharpening in remote sensing. *International J. Image and Data Fusion*, 1(1):25–45, 2010.
- [217] Cheng Shi, Fang Liu, Lingling Li, Licheng Jiao, Yiping Duan, and Shuang Wang. Learning interpolation via regional map for pan-sharpening. *IEEE Trans. Geosci. Remote Sens.*, 53(6):3417–3431, 2015.

- [218] Pengfei Liu, Liang Xiao, Jun Zhang, and Bushra Naz. Spatial-hessian-feature-guided variational model for pan-sharpening. *IEEE Trans. Geosci. Remote Sens.*, 54(4):2235–2253, 2016.
- [219] Qian Du, Nicholas H Younan, Roger King, and Vijay P Shah. On the performance evaluation of pan-sharpening techniques. *IEEE Geosci. Remote Sens. Lett.*, 4(4):518–522, 2007.
- [220] Hussein A Aly and Gaurav Sharma. A regularized model-based optimization framework for pan-sharpening. *IEEE Trans. Image Proc.*, 23(6):2596–2608, 2014.
- [221] Te-Ming Tu, Shun-Chi Su, Hsuen-Chyun Shyu, and Ping S Huang. A new look at ihs-like image fusion methods. *Information fusion*, 2(3):177–186, 2001.
- [222] Bruno Aiazzi, Stefano Baronti, and Massimo Selva. Improving component substitution pansharpening through multivariate regression of ms+ pan data. *IEEE Trans. Geosci. Remote Sens.*, 45(10):3230–3239, 2007.
- [223] Vijay P Shah, Nicolas H Younan, and Roger L King. An efficient pan-sharpening method via a combined adaptive pca approach and contourlets. *IEEE Trans. Geosci. Remote Sens.*, 46(5):1323–1335, 2008.
- [224] Xavier Otazu, María González-Audícana, Octavi Fors, and Jorge Núñez. Introduction of sensor spectral response into image fusion methods. application to wavelet-based methods. *IEEE Trans. Geosci. Remote Sens.*, 43(10):2376–2385, 2005.
- [225] Xiao Xiang Zhu and Richard Bamler. A sparse image fusion algorithm with application to pan-sharpening. *IEEE Trans. Geosci. Remote Sens.*, 51(5):2827–2836, 2013.

- [226] Cheng Jiang, Hongyan Zhang, Huanfeng Shen, and Liangpei Zhang. A practical compressed sensing-based pan-sharpening method. *IEEE Geosci. Remote Sens. Lett.*, 9(4):629–633, 2012.
- [227] Morteza Ghahremani and Hassan Ghassemian. Remote sensing image fusion using ripplet transform and compressed sensing. *IEEE Geosci.*, 12(3):502–506, 2015.
- [228] Morteza Ghahremani and Hassan Ghassemian. A compressed-sensing-based pan-sharpening method for spectral distortion reduction. *IEEE Trans. Geosci. Remote Sens.*, 54(4):2194–2206, 2016.
- [229] Roger L King and Jianwen Wang. A wavelet based algorithm for pan sharpening landsat 7 imagery. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, volume 2, pages 849–851. IEEE, 2001.
- [230] Andrea Garzelli, Filippo Nencini, Luciano Alparone, Bruno Aiazzi, and Stefano Baronti. Pan-sharpening of multispectral images: a critical review and comparison. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, volume 1. IEEE, 2004.
- [231] Yun Zhang and Gang Hong. An ihs and wavelet integrated approach to improve pan-sharpening visual quality of natural colour ikonos and quickbird images. *Information Fusion*, 6(3):225–234, 2005.
- [232] Wei Huang, Liang Xiao, Zhihui Wei, Hongyi Liu, and Songze Tang. A new pan-sharpening method with deep neural networks. *IEEE Geosci. Remote Sens. Lett.*, 12(5):1037–1041, 2015.
- [233] Jamal Saeedi and Karim Faez. A new pan-sharpening method using multiobjec-

- tive particle swarm optimization and the shiftable contourlet transform. *ISPRS J. photogrammetry and Remote Sensing*, 66(3):365–381, 2011.
- [234] Wenqing Wang, Licheng Jiao, and Shuyuan Yang. Novel adaptive component-substitution-based pan-sharpening using particle swarm optimization. *IEEE Geosci. Remote Sens. Lett.*, 12(4):781–785, 2015.
- [235] David M McKeown, Steven D Cochran, Stephen J Ford, J Chris McGlone, Jeffrey A Shufelt, and Daniel A Yocom. Fusion of hydice hyperspectral data with panchromatic imagery for cartographic feature extraction. *IEEE Trans. Geosci. Remote Sens.*, 37(3):1261–1277, 1999.
- [236] Giorgio Antonino Licciardi, Muhammad Murtaza Khan, Jocelyn Chanussot, Annick Montanvert, Laurent Condat, and Christian Jutten. Fusion of hyperspectral and panchromatic images using multiresolution analysis and nonlinear pca band reduction. *EURASIP J. Advances in Signal processing*, 2012(1):1–17, 2012.
- [237] Richard B Gomez, Amin Jazaeri, and Menas Kafatos. Wavelet-based hyperspectral and multispectral image fusion. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 36–42. International Society for Optics and Photonics, 2001.
- [238] Zhao Chen, Hanye Pu, Bin Wang, and Geng-Ming Jiang. Fusion of hyperspectral and multispectral images: A novel framework based on generalization of pan-sharpening methods. *IEEE Geosci. Remote Sens. Lett.*, 11(8):1418–1422, 2014.
- [239] Stefan Hinz and Albert Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS J. Photogrammetry and Remote Sensing*, 58(1):83–98, 2003.

- [240] Stefan Hinz and Albert Baumgartner. Multi-view fusion of road objects supported by self-diagnosis. In *2nd GRSS/ISPRS Joint Workshop on Remote Sens. and Data Fusion over Urban Areas*, pages 137–141. IEEE, 2003.
- [241] Paolo Gamba, Fabio Dell’Acqua, and Belur V Dasarathy. Urban remote sensing using multiple data sets: Past, present, and future. *Information Fusion*, 6(4):319–326, 2005.
- [242] Xiangchao Meng, Huanfeng Shen, Liangpei Zhang, Qiangqiang Yuan, and Huifang Li. A unified framework for spatio-temporal-spectral fusion of remote sensing images. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 2584–2587. IEEE, 2015.
- [243] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.
- [244] Bin Chen, Bo Huang, and Bing Xu. Constructing a unified framework for multi-source remotely sensed data fusion. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 2574–2577. IEEE, 2016.
- [245] Katia Stankov and Dong-Chen He. Detection of buildings in multispectral very high spatial resolution images using the percentage occupancy hit-or-miss transform. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 7(10):4069–4080, 2014.
- [246] Roger L. King. A challenge for high spatial, spectral, and temporal resolution data fusion. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, volume 6, pages 2602–2604. IEEE, 2000.

- [247] Albert Yu-Min Lin, Alexandre Novo, Shay Har-Noy, Nathan D Ricklin, and Kostas Stamatou. Combining geoeye-1 satellite remote sensing, uav aerial imaging, and geophysical surveys in anomaly detection applied to archaeology. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 4(4):870–876, 2011.
- [248] Michele Dalponte, Lorenzo Bruzzone, and Damiano Gianelle. Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas. *IEEE Trans. Geosci. and Remote Sens.*, 46(5):1416–1427, 2008.
- [249] Christian Debes, Andreas Merentitis, Roel Heremans, Jürgen Hahn, Nikolaos Frangiadakis, Tim van Kasteren, Wenzhi Liao, Rik Bellens, Aleksandra Pižurica, Sidharta Gautama, et al. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 7(6):2405–2418, 2014.
- [250] Renbo Luo, Wenzhi Liao, Hongyan Zhang, Youguo Pi, and Wilfried Philips. Classification of cloudy hyperspectral image and lidar data based on feature fusion and decision fusion. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*. IEEE, 2016.
- [251] Aparajithan Sampath and Jie Shan. Building boundary tracing and regularization from airborne lidar point clouds. *Photogrammetric Engineering & Remote Sensing*, 73(7):805–812, 2007.
- [252] Vivek Verma, Rakesh Kumar, and Stephen Hsu. 3d building detection and modeling from aerial lidar data. In *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2213–2220. IEEE, 2006.
- [253] Yi Sun, Cheng Wang, Jonathan Li, Zongliang Zhang, Dawei Zai, Pengdi Huang, and Chenglu Wen. Automated segmentation of lidar point clouds for building rooftop

- extraction. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 1472–1475. IEEE, 2016.
- [254] Shaohui Sun and Carl Salvaggio. Aerial 3d building detection and modeling from airborne lidar point clouds. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 6(3):1440–1449, 2013.
- [255] Yunsheng Wang, Holger Weinacker, and Barbara Koch. A lidar point cloud based procedure for vertical canopy structure analysis and 3d single tree modelling in forest. *Sensors*, 8(6):3938–3951, 2008.
- [256] Wenkai Li, Qinghua Guo, Marek K Jakubowski, and Maggi Kelly. A new method for segmenting individual trees from the lidar point cloud. *Photogrammetric Engineering & Remote Sensing*, 78(1):75–84, 2012.
- [257] Bao Yunfei, Li Guoping, Cao Chunxiang, Li Xiaowen, Z Hao, H Qisheng, B Linyan, and C Chaoyi. Classification of lidar point cloud and generation of dtm from lidar height and intensity data in forested area. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(7):313–318, 2008.
- [258] Bernhard Höfle. Radiometric correction of terrestrial lidar point cloud data for individual maize plant detection. *IEEE Geosci.*, 11(1):94–98, 2014.
- [259] Tomohiro Matsuki, Naoto Yokoya, and Akira Iwasaki. Hyperspectral tree species classification of japanese complex mixed forest with the aid of lidar data. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 8(5):2177–2187, 2015.

- [260] David R Lyzenga. Shallow-water bathymetry using combined lidar and passive multispectral scanner data. *Int. J. Remote Sens.*, 6(1):115–125, 1985.
- [261] Zhigang Pan, Craig Glennie, Juan Carlos Fernandez-Diaz, Ramesh Shrestha, Bill Carter, Darren Hauser, Abhinav Singhania, and Michael Sartori. Fusion of bathymetric lidar and hyperspectral imagery for shallow water bathymetry. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 3792–3795. IEEE, 2016.
- [262] Zhigang Pan, Craig L Glennie, Juan Carlos Fernandez-Diaz, Carl J Legleiter, and Brandon Overstreet. Fusion of lidar orthowaveforms and hyperspectral imagery for shallow river bathymetry and turbidity estimation. *IEEE Trans. Geosci. Remote Sens.*, 54(7):4165–4177, 2016.
- [263] Mahdi Khodadadzadeh, Jun Li, Saurabh Prasad, and Antonio Plaza. Fusion of hyperspectral and lidar remote sensing data using multiple feature learning. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 8(6):2971–2983, 2015.
- [264] Mauro Dalla Mura, Jón Atli Benediktsson, Björn Waske, and Lorenzo Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Trans. Ge*, 48(10):3747–3762, 2010.
- [265] Pedram Ghamisi, Jon Atli Benediktsson, and Johannes R Sveinsson. Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction. *IEEE Trans. Geosci. Remote Sens.*, 52(9):5771–5782, 2014.
- [266] Benqin Song, Jun Li, Mauro Dalla Mura, Peijun Li, Antonio Plaza, José M Bioucas-Dias, Jon Atli Benediktsson, and Jocelyn Chanussot. Remotely sensed image classi-

- fication using sparse representations of morphological attribute profiles. *IEEE Trans. Geosci. Remote Sens.*, 52(8):5122–5136, 2014.
- [267] Wenzhi Liao, Aleksandra Pižurica, Rik Bellens, Sidharta Gautama, and Wilfried Philips. Generalized graph-based fusion of hyperspectral and lidar data using morphological features. *IEEE Geosci. Remote Sens. Lett.*, 12(3):552–556, 2015.
- [268] Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.*, 42(8):1778–1790, 2004.
- [269] Jun Li, José M Bioucas-Dias, and Antonio Plaza. Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *IEEE Trans. Geosci. Remote Sens.*, 50(3):809–823, 2012.
- [270] Yuhang Zhang and Saurabh Prasad. Multisource geospatial data fusion via local joint sparse representation. *IEEE Trans. Geosci. Remote Sens.*, 54(6):3265–3276, 2016.
- [271] Pieter Kempeneers, Fernando Sedano, Lucia Seebach, Peter Strobl, and Jesús San-Miguel-Ayanz. Data fusion of different spatial resolution remote sensing images applied to forest-type mapping. *IEEE Trans. Geosci. Remote Sens.*, 49(12):4977–4986, 2011.
- [272] P. S. Bullen. *Handbook of Means and Their Inequalities*, chapter 3 The Power Means, page 175. Kluwer Academic Publishers, 2003.
- [273] Norman Johnson, Samuel Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. Wiley-Interscience, 2 edition, Oct. 1994.

- [274] Irving S. Reed and Xiaoli Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Trans. Acoust., Speech, Signal Process.*, 38(10):1760–1770, Oct. 1990.
- [275] P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell. Muufl gulfport hyperspectral and lidar airborne data set. Technical Report Rep. REP-2013-570, University of Florida, Gainesville, FL, Oct. 2013.
- [276] Changzhe Jiao and Alina Zare. Functions of multiple instances for learning target signatures. *IEEE Trans. Geosci. Remote Sens.*, 53(8):4670–4686, Aug. 2015.
- [277] Louis L Scharf and L Todd McWhorter. Adaptive matched subspace detectors and adaptive coherence estimators. In *Proc. 30th Asilomar Conf. Signals Syst.*, pages 1114–1117. IEEE, Nov. 1996.
- [278] Shawn Kraut, Louis L Scharf, and Ronald W Butler. The adaptive coherence estimator: a uniformly most-powerful-invariant adaptive detection statistic. *IEEE Trans. Signal Proc.*, 53(2):427–438, 2005.
- [279] N.B. Pulsone and Michael A. Zatman. A computationally efficient two-step implementation of the glrt. *IEEE Trans. Signal Proc.*, 48(3):609–616, Mar 2000.
- [280] Martin Wilk and Ram Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.
- [281] Xiaoxiao Du, Alina Zare, Paul Gader, and Dmitri Dranishnikov. Spatial and spectral unmixing using the beta compositional model. *IEEE J. Sel. Topics. Appl. Earth Observ.*, 7(6):1994–2003, 2014.

- [282] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2010.
- [283] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.
- [284] Xiaoxiao Du and Alina Zare. Technical report: Scene label ground truth map for muufl gulfport data set. Technical Report 20170417, University of Florida, Gainesville, FL, Apr. 2017.
- [285] Michael E. Tipping. The relevance vector machine. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 652–658, 2000.
- [286] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Research (JMLR)*, 1:211–244, 2001.
- [287] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 155–161, 1997.
- [288] Derek T Anderson, Muhammad Islam, Roger King, Nicolas Younan, Joshua Fairley, Stacy Howington, Frederick Petry, Paul Elmore, and Alina Zare. Binary fuzzy measures and choquet integration for multi-source fusion. In *Int. Conf. Military Technologies (ICMT)*, pages 676–681. IEEE, 2017.
- [289] Muhammad Islam, Derek Anderson, Xiaoxiao Du, Timothy Havens, and Christian

Wagner. Efficient binary fuzzy measure representation and choquet integral learning.

Submitted.

VITA

Xiaoxiao Du received her Bachelor of Engineering degree with Chu Kochen Honors Certificate at Zhejiang University, China in June, 2011. She received her Master of Science degree in December 2013 and Doctor of Philosophy degree in 2017 from the Department of Electrical and Computer Engineering at the University of Missouri, Columbia, MO. She was the recipient of the “Outstanding MS Student” award at the Missouri Honor Awards, 2013.