

AN INVESTIGATION ON HYPERSPECTRAL IMAGE CLASSIFIERS FOR REMOTE
SENSING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

OKAN BİLGE ÖZDEMİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

JUNE 2013

AN INVESTIGATION ON HYPERSPECTRAL IMAGE CLASSIFIERS FOR REMOTE SENSING

Submitted by **Okan Bilge Özdemir** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems, Middle East Technical University** by,

Prof. Dr. Nazife Baykal
Director, **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department, **Information Systems**

Prof. Dr. Yasemin Yardımcı Çetin
Supervisor, **Information Systems, METU**

Examining Committee Members:

Prof. Dr. H. Şebnem Düzgün
Department of Mining Engineering, METU

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU

Asst. Prof. Dr. Erhan Eren
Information Systems, METU

Asst. Prof. Dr. Banu Günel
Information Systems, METU

Asst. Prof. Dr. Alptekin Temizel
Work Based Learning, METU

Date: 19.06.2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Okan Bilge Özdemir

Signature : _____

ABSTRACT

AN INVESTIGATION ON HYPERSPECTRAL IMAGE CLASSIFIERS FOR REMOTE SENSING

Özdemir, Okan Bilge

MS., Department of Information Systems

Supervisor: Prof. Dr. Yasemin Yardımcı Çetin

June 2013, 190 pages

Hyperspectral image processing is improved by the capabilities of multispectral image processing with high spectral resolution. In this thesis, we explored hyperspectral classification with Support Vector Machines (SVM), Maximum Likelihood (ML) and K-Nearest Neighborhood algorithms. We analyzed the effect of training data on classification accuracy. For this purpose, we implemented three different training data selection methods; first N sample selection, randomly N sample selection and uniformly N sample selection methods. We employed Principal Component Analysis (PCA) as pre-processing method and conducted experiments with different number of principal components for all three classification algorithms. As a post-processing method following pixelwise classification, filtering with 3x3 window and majority voting with meanshift segmentation methods are used to incorporate spatial information over spectral information.

The experiments showed that without using pre-processing and post-processing SVM procures better classification accuracies than the other algorithms for all training data sizes. ML is inferior for lower number of training data samples but improves its performance with lower number of principal components. K-NN algorithm provides almost the same accuracies for more than 10 principal components. PCA usage does not improve SVM performance but decreases classification time for larger scenes. Filtering with 3x3 window method improves the classification accuracy by 4-5%. However, spatial information usage by employing majority voting with meanshift segmentation method performs better than filtering 3x3 window. Classification with both pre-processing and post-processing improves classification accuracy and decreases classification time. The largest improvement is for the ML method with lower number of training data.

Keywords: Hyperspectral Classification, Meanshift Segmentation, Support Vector Machines, Maximum Likelihood, K-Nearest Neighborhood

ÖZ

UZAKTAN ALGILAMA İÇİN HİPERSPEKTRAL İMGE SINIFLANDIRICILARI ÜZERİNE BİR İNCELEME

Özdemir, Okan Bilge

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Yasemin Yardımcı Çetin

Haziran 2013, 190 sayfa

Yüksek spektral bilgi sayesinde hiperspektral imge işleme, multispektral imge işlemeye göre daha fazla kapasiteye sahiptir. Bu çalışmada, Destek Vektör Makinaları (DVM), En Büyük Olabilirlik (EBO) ve K-En Yakın Komşu algoritmaları kullanılarak hiperspektral sınıflandırma algoritmaları incelenmiştir. Çalışmada ayrıca öğrenme verisinin sınıflandırma başarımına etkisi de incelenmiştir. Bunun için ilk N örneğin seçimi, rasgele N örnek seçimi ve homojen bir biçimde N örnek seçimi olarak farklı üç öğrenme verisi seçim yöntemi kullanılmıştır. Ön işleme metodu olarak Temel Bileşen Analizi (TBA) kullanılmıştır. Her üç algoritma için farklı temel bileşen sayıları ile deneyler yapılmıştır. Spektral bilgiye ek olarak yersel bilginin kullanımı için piksel bazında sınıflandırma sonrasında, son işleme olarak 3x3 pencere ile filtreleme ve ortalama kaydırmalı bölütleme ile çoğunluk oylaması yöntemleri kullanılmıştır.

Deneylede görüldüğü üzere ön işleme ve son işleme kullanılmadığında SVM'in bütün öğrenme verisi boyutları için diğer algoritmalarından daha iyi sonuç verdiği görülmüştür. EBO yöntemi düşük öğrenme verisi ile kötü sonuçlar aldığı gözlenmiştir. Düşük sayıda temel bileşen kullanıldığında EBO yönteminin performansını artırdığı görülmüştür. K-NN algoritmasının sınıflandırma başarımı 10 temel bileşenden daha fazla kullanıldığında değişmediği tespit edilmiştir. TBA kullanımı DVM algoritmasını etkilemediği gözlenmiştir. TBA kullanımı büyük imgeler için sınıflandırma zamanını düşürdü. 3x3 pencere ile filtreleme yöntemi sınıflandırma yüzdesini %4-5 artırdığı tespit edilmiştir. Ortalama kaydırmalı bölütleme ile çoğunluk oylaması, 3x3 pencere ile filtreleme yönteminden daha iyi sonuç verdiği görülmüştür. Ön işleme ve son işleme yöntemlerinin birlikte kullanılması sınıflandırma yüzdesini artırdığı ve sınıflandırma zamanını düşürdüğü gözlemlendi. Düşük sayıda öğrenme verisi kullanıldığında en büyük gelişimi ML algoritmasının yaptığı görüldü.

Anahtar Kelimeler: Hiperspektral Sınıflandırma, Ortalama Kaydırmalı Bölütleme, Destek Vektör Makinaları, En Büyük Olabilirlik , K-En Yakın Komşu

To my father Mehmet Ali Özdemir

ACKNOWLEDGEMENTS

Special thanks to my supervisor Prof.Dr. Yasemin Yardımcı Çetin for her encouragement, priceless guidance, continuous support and friendly attitude throughout my research.

I would like to thank Prof.Dr. H.Şebnem Düzgün, Asst. Prof. Dr. Erhan Eren, Asst. Prof. Dr. Banu Günel and Asst. Prof. Dr. Alptekin Temizel for reviewing my work.

Special thanks go to my colleagues Umut Cinar, Ersin Karaman and Ekin Gedik for their endless support, encouragement, and suggestions since the beginning of this research. I also thank my family for their love, trust and patience during this study.

I also thank to my friends Elif Gök, Nurcan Alkış, Bilge Sürün, Aykut Mert Yakut, Gökçe Oğuz for their support during my study. I would like to thank to Informatics Institute members Sibel Gülnar, Sibel Sel, Hakan Güler, Murat Yabancı and Ferda Ercan for their support.

Special thanks to Cemzade Kader for her endless support, patience and encouragement.

TABLE OF CONTENTS

ABSTRACT	V
ACKNOWLEDGEMENTS.....	VIII
TABLE OF CONTENTS	IX
LIST OF FIGURES	XVI
LIST OF TABLES.....	XX
CHAPTERS	
1-INTRODUCTION	1
1.1 MOTIVATION	1
1.2 SCOPE AND GOAL	1
1.3 OUTLINE OF THESIS	2
2-BACKGROUND.....	3
2.1 LITERATURE SURVEY	5
2.1.1 Dimensionality Reduction	5
2.1.2 Classification	6
2.1.3 Post-Processing Methods.....	8
3-METHODOLOGY	9
3.1 DATA	9
3.1.1 Pavia University Scene	9

3.1.2	<i>Indian Pines Scene</i>	11
3.1.3	<i>Salinas Scene</i>	13
3.2	INTRODUCTION.....	14
3.3	TRAINING DATA SELECTION	15
3.4	PRE-PROCESSING WITH PCA	17
3.5	POST PROCESSING WITH SPATIAL INFORMATION	18
3.5.1	<i>Filtering with 3x3 Window</i>	18
3.5.2	<i>Majority Voting with Meanshift Segmentation</i>	19
3.6	POST-PROCESSING AND PRE-PROCESSING.....	24
4	EXPERIMENTS	25
4.1	MEASUREMENT METRICS	25
4.2	INDIAN PINES SCENE EXPERIMENTS	26
4.2.1	<i>Indian Pines Scene with No Pre-Processing and No Post-Processing</i>	26
4.2.2	<i>Indian Pines Scene with Pre-Processing Only</i>	28
4.2.3	<i>Indian Pines Scene with Post-Processing Only</i>	30
4.2.4	<i>Indian Pines Scene with Post-Processing and Pre-Processing</i>	32
4.3	SALINAS SCENE EXPERIMENTS.....	33
4.3.1	<i>Salinas Scene without Pre-Processing and Post-Processing</i>	33
4.3.2	<i>Salinas Scene with Pre-Processing</i>	36
4.3.3	<i>Salinas Scene with Post-Processing</i>	38

4.3.4	<i>Salinas Scene with Pre-Processing and Post-Processing</i>	40
4.4	PAVIA UNIVERSITY SCENE EXPERIMENTS	42
4.4.1	<i>Pavia University Scene without Pre-Processing and Post-Processing</i>	42
4.4.2	<i>Pavia University Scene with Pre-Processing</i>	43
4.4.3	<i>Pavia University Scene with Post-Processing</i>	44
4.4.4	<i>Pavia University Scene with Pre-Processing and Post-Processing</i>	46
4.5	THE EFFECTS OF SEGREGATION OF TRAINING AND TESTING DATA	48
5	CONCLUSION	53
5.1	SUMMARY	53
5.2	COMPARISON WITH THE LITERATURE AND DISCUSSION	55
5.3	FUTURE WORK	57
	REFERENCES	58
	APPENDICES	65
	APPENDIX-A INDIAN PINES SCENE RESULTS	65
	<i>First N Sample without Pre-Processing or Post Processing</i>	65
	<i>First N Sample with Pre-Processing (PCA)</i>	67
	<i>First N Sample with Post-Processing (Filtering with 3x3 window)</i>	69
	<i>First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	71
	<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	73

<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	76
<i>Randomly Selected N Sample without Pre-Processing or Post Processing</i>	78
<i>Randomly Selected N Sample with Pre-Processing (PCA)</i>	80
<i>Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	82
<i>Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	84
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	87
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	90
<i>Uniformly Selected N Sample without Pre-Processing or Post Processing</i>	92
<i>Uniformly Selected N Sample with Pre-Processing (PCA)</i>	94
<i>Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	96
<i>Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	98
<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	101
<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	104
APPENDIX-B: SALINAS SCENE RESULTS	107
<i>First N Sample without Pre-Processing or Post Processing</i>	107
<i>First N Sample with Pre-Processing (PCA)</i>	109

<i>First N Sample with Post-Processing (Filtering with 3x3 window).....</i>	<i>111</i>
<i>First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	<i>113</i>
<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i> <i>.....</i>	<i>115</i>
<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with</i> <i>Meanshift Segmentation).....</i>	<i>118</i>
<i>Randomly Selected N Sample without Pre-Processing or Post Processing</i>	<i>120</i>
<i>Randomly Selected N Sample with Pre-Processing (PCA)</i>	<i>122</i>
<i>Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	<i>124</i>
<i>Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift</i> <i>Segmentation).....</i>	<i>126</i>
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with</i> <i>3x3 window)</i>	<i>129</i>
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority</i> <i>Voting with Meanshift Segmentation)</i>	<i>132</i>
<i>Uniformly Selected N Sample without Pre-Processing or Post Processing</i>	<i>134</i>
<i>Uniformly Selected N Sample with Pre-Processing (PCA).....</i>	<i>136</i>
<i>Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	<i>138</i>
<i>Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift</i> <i>Segmentation).....</i>	<i>140</i>
<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with</i> <i>3x3 window)</i>	<i>143</i>

<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	146
APPENDIX-C: PAVIA UNIVERSITY SCENE RESULTS	149
<i>First N Sample without Pre-Processing or Post Processing</i>	149
<i>First N Sample with Pre-Processing (PCA)</i>	151
<i>First N Sample with Post-Processing (Filtering with 3x3 window)</i>	153
<i>First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	156
<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	158
<i>First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	160
<i>Randomly Selected N Sample without Pre-Processing or Post Processing</i>	162
<i>Randomly Selected N Sample with Pre-Processing (PCA)</i>	165
<i>Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	167
<i>Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)</i>	169
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	171
<i>Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	174
<i>Uniformly Selected N Sample without Pre-Processing or Post Processing</i>	176
<i>Uniformly Selected N Sample with Pre-Processing (PCA)</i>	178
<i>Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)</i>	180

<i>Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....</i>	<i>182</i>
<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)</i>	<i>185</i>
<i>Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)</i>	<i>188</i>

LIST OF FIGURES

Figure 1 Reflectance spectra of some minerals adapted from [6]	4
Figure 2 Dimensionality reduction diagram adapted from [12]	6
Figure 3 Flowchart of the proposed spectral–spatial classification scheme from [50]	8
Figure 4 Pavia University Scene RGB Image(top) Groundtruth Image(bottom)(Courtesy of Prof. Paulo Gamba from Pavia University, Italy.)	10
Figure 5 Indian Pines Scene (Left) RGB Image (Right) Groundtruth Image (Retrieved from https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html)(accessed 06.05.2013).....	12
Figure 6 Salinas Scene (Left) Sample Band (Right) Groundtruth Image	14
Figure 7 Training Sample Selection N=700 (top)First N Samples, (middle)Uniformly Selected Samples (bottom) Randomly Selected Samples	16
Figure 8 Flow Chart of Classification with Pre-Processing	17
Figure 9 Weighting methods for Filtering with 3x3 Window	18
Figure 10 Different methods for filtering with 3x3 window method.....	19
Figure 11 Post Processing with Meanshift Segmentation	20
Figure 12 Pavia University Meanshift Segmentation Result (top) and Ground Truth (Bottom)	21

Figure 13 Segmentation Image (left) and Ground Truth (right) of Indian Pines Scene Salinas Scene.....	22
Figure 14 Segmentation Image (top) and Ground Truth (bottom) of Salinas Scene	22
Figure 15 Flow chart of pre-processing and post-processing usage for classification	24
Figure 16 Indian Pines Scene Classification Accuracies.....	27
Figure 17 Indian Pines Scene (left) First N (N=380) Sample Classification Result (right) Uniformly Selected N (N=380) Sample Classification Result.....	28
Figure 18 Indian Pines Scene classification with PCA	29
Figure 19 IPS with different number of principal components	30
Figure 20 IPS with Filtering 3x3 Window	31
Figure 21 IPS Classification Results for Majority Voting with Meanshift Segmentation	31
Figure 22 IPS with Majority Voting with Meanshift Segmentation	32
Figure 23 Salinas Scene Classification Results without Pre-Processing and Post- Processing.....	34
Figure 24 Salinas Scene Classification Results N=900 (top) Uniformly Selected N Samples (middle) First N Samples from right side of the image (bottom) Last N Samples from left side of the image	35
Figure 25 SSC with Pre-Processing.....	36

Figure 26 Salinas Scene classification with PCA results	37
Figure 27 SVM-LNR Classification Results (Randomly Selected N Samples with PCA).....	38
Figure 28 Salinas Scene Filtering with 3x3 Window.....	39
Figure 29 Salinas Scene Majority Voting with Meanshift Segmentation.....	39
Figure 30 The Classification Accuracies for ML (left) and SVM-RBF (right)	40
Figure 31 The Classification Accuracies for SVM-LNR (left) and K-NN (right). 41	
Figure 32 The Classification Results of Pavia University Scene without Pre-Processing and Post-Processing	42
Figure 33 The Classification Accuracies of Pavia University Scene with Pre-Processing.....	43
Figure 34 Pavia University Scene Results for Filtering with 3x3 Window	45
Figure 35 Pavia University Scene Results for Majority Voting with Meanshift Segmentation.....	46
Figure 36 Pavia University Scene Classification Results for Filtering with 3x3 Window Method ML (top-left), SVM-RBF(top-right) SVM-LNR (bottom-left) and K-NN (bottom-right)	47
Figure 37 Pavia University Scene Classification Results for Filtering with 3x3 Window Method ML (top-left), SVM-RBF(top-right) SVM-LNR (bottom-left) and K-NN (bottom-right)	48
Figure 38 Sample Groundtruth for SSC (N=700)	49

Figure 39 Classification Results for SSC with and without Segregation of Training Data.....	50
---	----

Figure 40 Classification Results for SSC with and without Segregation of Training Data by using Majority Voting with Meanshift Segmentation Method	51
--	----

LIST OF TABLES

Table 1 Pavia University Scene Groundtruth Classes and Sample Numbers	9
Table 2 Indian Pines Scene Groundtruth Classes and Sample Numbers	11
Table 3 Salinas Scene Groundtruth Classes and Sample Numbers	13
Table 4 IPS - ML -First N Sample without Pre-Processing or Post Processing	65
Table 5 IPS - SVM-RBF - First N Sample without Pre-Processing or Post Processing.....	66
Table 6 IPS - SVM-LNR - First N Sample without Pre-Processing or Post Processing.....	66
Table 7 IPS – K-NN - First N Sample without Pre-Processing or Post Processing	67
Table 8 IPS – ML - First N Sample with Pre-Processing (PCA).....	67
Table 9 IPS – SVM-RBF - First N Sample with Pre-Processing (PCA)	68
Table 10 IPS – SVM-LNR - First N Sample with Pre-Processing (PCA).....	68
Table 11 IPS – K-NN - First N Sample with Pre-Processing (PCA).....	69
Table 12 IPS - ML - First N Sample with Post-Processing (Filtering with 3x3 window).....	69
Table 13 IPS - SVM-RBF - First N Sample with Post-Processing (Filtering with 3x3 window).....	70

Table 14 IPS - SVM-LNR - First N Sample with Post-Processing (Filtering with 3x3 window)	70
Table 15 IPS - K-NN - First N Sample with Post-Processing (Filtering with 3x3 window)	71
Table 16 IPS - ML - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	71
Table 17 IPS - SVM-RBF - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	72
Table 18 IPS - SVM-LNR - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	72
Table 19 IPS - K-NN - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	73
Table 20 IPS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	73
Table 21 IPS – SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	74
Table 22 IPS – SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	74
Table 23 IPS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation).....	76
Table 24 IPS – SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation).....	77

Table 25 IPS – SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	77
Table 26 IPS – K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	78
Table 27 IPS - ML -Randomly Selected N Sample without Pre-Processing or Post Processing.....	78
Table 28 IPS – SVM-RBF -Randomly Selected N Sample without Pre-Processing or Post Processing	79
Table 29 IPS – SVM-LNR -Randomly Selected N Sample without Pre-Processing or Post Processing	79
Table 30 IPS – K-NN -Randomly Selected N Sample without Pre-Processing or Post Processing.....	80
Table 31 IPS – ML - Randomly Selected N Sample with Pre-Processing (PCA) .	80
Table 32 IPS – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA)	81
Table 33 IPS – SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA)	81
Table 34 IPS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA)	82
Table 35 IPS - ML - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	82

Table 36 IPS – SVM-RBF - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	83
Table 37 IPS – SVM-LNR - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	83
Table 38 IPS – K-NN - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	84
Table 39 IPS - ML - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	84
Table 40 IPS – SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	85
Table 41 IPS – SVM-LNR - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	85
Table 42 IPS – K-NN - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	86
Table 43 IPS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	87
Table 44 IPS – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	88
Table 45 IPS – SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	88
Table 46 IPS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	89

Table 47 IPS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	90
Table 48 IPS – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	91
Table 49 IPS – SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	91
Table 50 IPS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	92
Table 51 IPS - ML - Uniformly Selected N Sample without Pre-Processing or Post Processing.....	92
Table 52 IPS – SVM-RBF - Uniformly Selected N Sample without Pre-Processing or Post Processing	93
Table 53 IPS – SVM-LNR - Uniformly Selected N Sample without Pre-Processing or Post Processing	93
Table 54 IPS – K-NN - Uniformly Selected N Sample without Pre-Processing or Post Processing.....	94
Table 55 IPS – ML - Uniformly Selected N Sample with Pre-Processing (PCA).	94
Table 56 IPS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA)	95
Table 57 IPS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA)	95

Table 58 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA)	96
Table 59 IPS - ML - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	96
Table 60 IPS – SVM-RBF - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	97
Table 61 IPS – SVM-LNR - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	97
Table 62 IPS – K-NN - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	98
Table 63 IPS - ML - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	98
Table 64 IPS – SVM-RBF - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	99
Table 65 IPS – SVM-LNR - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	99
Table 66 IPS – K-NN - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	100
Table 67 IPS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	101
Table 68 IPS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	102

Table 69 IPS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	102
Table 70 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	103
Table 71 IPS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	104
Table 72 IPS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	105
Table 73 IPS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	105
Table 74 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	106
Table 75 SSC - ML -First N Sample without Pre-Processing or Post Processing	107
Table 76 SSC - SVM-RBF -First N Sample without Pre-Processing or Post Processing.....	108
Table 77 SSC - SVM-LNR -First N Sample without Pre-Processing or Post Processing.....	108
Table 78 SSC - K-NN -First N Sample without Pre-Processing or Post Processing	109
Table 79 SSC – ML - First N Sample with Pre-Processing (PCA).....	109
Table 80 SSC – SVM-RBF - First N Sample with Pre-Processing (PCA)	110

Table 81 SSC – SVM-LNR - First N Sample with Pre-Processing (PCA)	110
Table 82 SSC – K-NN - First N Sample with Pre-Processing (PCA)	111
Table 83 SSC - ML - First N Sample with Post-Processing (Filtering with 3x3 window)	111
Table 84 SSC - SVM-RBF - First N Sample with Post-Processing (Filtering with 3x3 window)	112
Table 85 SSC - SVM-LNR - First N Sample with Post-Processing (Filtering with 3x3 window)	112
Table 86 SSC - K-NN - First N Sample with Post-Processing (Filtering with 3x3 window)	113
Table 87 SSC - ML - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	113
Table 88 SSC - SVM-RBF - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	114
Table 89 SSC - SVM-LNR - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	114
Table 90 SSC - K-NN - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	115
Table 91 SSC - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	115
Table 92 SSC - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	116

Table 93 SSC - SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	116
Table 94 SSC - K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	117
Table 95 SSC - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	118
Table 96 SSC - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	119
Table 97 SSC - SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	119
Table 98 SSC - K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	120
Table 99 SSC - ML -Randomly Selected N Sample without Pre-Processing or Post Processing.....	120
Table 100 SSC - SVM-RBF -Randomly Selected N Sample without Pre-Processing or Post Processing	121
Table 101 SSC - SVM-LNR -Randomly Selected N Sample without Pre-Processing or Post Processing	121
Table 102 SSC - K-NN -Randomly Selected N Sample without Pre-Processing or Post Processing.....	122
Table 103 SSC – ML - Randomly Selected N Sample with Pre-Processing (PCA)	122

Table 104 SSC – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA).....	123
Table 105 SSC – SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA).....	123
Table 106 SSC – K-NN - Randomly Selected N Sample with Pre-Processing (PCA).....	124
Table 107 SSC - ML - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	124
Table 108 SSC - SVM-RBF - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	125
Table 109 SSC - SVM-LNR - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	125
Table 110 SSC - K-NN - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	126
Table 111 SSC - ML - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	126
Table 112 SSC - SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	127
Table 113 SSC - SVM-LNR - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	127
Table 114 SSC - K-NN - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	128

Table 115 SSC - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	129
Table 116 SSC - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	130
Table 117 SSC - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	130
Table 118 SSC - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	131
Table 119 SSC - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	132
Table 120 SSC - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	133
Table 121 SSC - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	133
Table 122 SSC - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	134
Table 123 SSC - ML - Uniformly Selected N Sample without Pre-Processing or Post Processing.....	134
Table 124 SSC - SVM-RBF - Uniformly Selected N Sample without Pre-Processing or Post Processing	135
Table 125 SSC - SVM-LNR - Uniformly Selected N Sample without Pre-Processing or Post Processing	135

Table 126 SSC - K-NN - Uniformly Selected N Sample without Pre-Processing or Post Processing	136
Table 127 SSC – ML - Uniformly Selected N Sample with Pre-Processing (PCA)	136
Table 128 SSC – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA).....	137
Table 129 SSC – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA).....	137
Table 130 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA).....	138
Table 131 SSC - ML - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	138
Table 132 SSC - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	139
Table 133 SSC - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	139
Table 134 SSC - K-NN - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	140
Table 135 SSC - ML - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	140
Table 136 SSC - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	141

Table 137 SSC - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	141
Table 138 SSC - K-NN - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	142
Table 139 SSC - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	143
Table 140 SSC - SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	144
Table 141 SSC - SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	144
Table 142 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	145
Table 143 SSC - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	146
Table 144 SSC - SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	147
Table 145 SSC - SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	147
Table 146 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	148
Table 147 PUS - ML -First N Sample without Pre-Processing or Post Processing	149

Table 148 PUS - SVM-RBF -First N Sample without Pre-Processing or Post Processing.....	150
Table 149 PUS - SVM-LNR -First N Sample without Pre-Processing or Post Processing.....	150
Table 150 PUS - K-NN -First N Sample without Pre-Processing or Post Processing	151
Table 151 PUS – ML - First N Sample with Pre-Processing (PCA).....	151
Table 152 PUS – SVM-RBF - First N Sample with Pre-Processing (PCA)	152
Table 153 PUS – SVM-LNR - First N Sample with Pre-Processing (PCA).....	152
Table 154 PUS – K-NN - First N Sample with Pre-Processing (PCA).....	153
Table 155 PUS - ML - First N Sample with Post-Processing (Filtering with 3x3 window)	153
Table 156 PUS - SVM-RBF - First N Sample with Post-Processing (Filtering with 3x3 window)	154
Table 157 PUS - SVM-LNR - First N Sample with Post-Processing (Filtering with 3x3 window)	154
Table 158 PUS - K-NN - First N Sample with Post-Processing (Filtering with 3x3 window)	155
Table 159 PUS - ML - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	156

Table 160 PUS - SVM-RBF - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	156
Table 161 PUS - SVM-LNR - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	157
Table 162 PUS - K-NN - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)	157
Table 163 PUS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	158
Table 164 PUS - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	158
Table 165 PUS - SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	159
Table 166 PUS - K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	159
Table 167 PUS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	160
Table 168 PUS - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	161
Table 169 PUS - SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	161
Table 170 PUS - K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	162

Table 171 PUS - ML -Randomly Selected N Sample without Pre-Processing or Post Processing	162
Table 172 PUS - SVM-RBF -Randomly Selected N Sample without Pre-Processing or Post Processing	163
Table 173 PUS - SVM-LNR -Randomly Selected N Sample without Pre-Processing or Post Processing	163
Table 174 PUS - K-NN -Randomly Selected N Sample without Pre-Processing or Post Processing	164
Table 175 PUS – ML - Randomly Selected N Sample with Pre-Processing (PCA)	165
Table 176 PUS – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA).....	165
Table 177 PUS – SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA).....	166
Table 178 PUS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA).....	166
Table 179 PUS - ML - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	167
Table 180 PUS - SVM-RBF - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	167
Table 181 PUS - SVM-LNR - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	168

Table 182 PUS - K-NN - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)	168
Table 183 PUS - ML - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	169
Table 184 PUS - SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	169
Table 185 PUS - SVM-LNR - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	170
Table 186 PUS - K-NN - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	170
Table 187 PUS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	171
Table 188 PUS - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	172
Table 189 PUS - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	172
Table 190 PUS - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	173
Table 191 PUS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	174
Table 192 PUS - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	175

Table 193 PUS - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	175
Table 194 PUS - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	176
Table 195 PUS - ML - Uniformly Selected N Sample without Pre-Processing or Post Processing	176
Table 196 PUS - SVM-RBF - Uniformly Selected N Sample without Pre-Processing or Post Processing	177
Table 197 PUS - SVM-LNR - Uniformly Selected N Sample without Pre-Processing or Post Processing	177
Table 198 PUS - K-NN - Uniformly Selected N Sample without Pre-Processing or Post Processing	178
Table 199 PUS – ML - Uniformly Selected N Sample with Pre-Processing (PCA)	178
Table 200 PUS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA).....	179
Table 201 PUS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA).....	179
Table 202 PUS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA).....	180
Table 203 PUS - ML - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window).....	180

Table 204 PUS - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	181
Table 205 PUS - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	181
Table 206 PUS - K-NN - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)	182
Table 207 PUS - ML - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	182
Table 208 PUS - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	183
Table 209 PUS - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	183
Table 210 PUS - K-NN - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation).....	184
Table 211 PUS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)	185
Table 212 PUS - SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	186
Table 213 PUS - SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	186
Table 214 PUS - K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window).....	187

Table 215 PUS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation).....	188
Table 216 PUS - SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	189
Table 217 PUS - SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	189
Table 218 PUS - K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)	190

CHAPTER 1

INTRODUCTION

1.1 Motivation

Classification is one of the most prominent research areas in hyperspectral image processing. Hyperspectral classification is used in many applications including city planning, mining and military decision support. They provide invaluable information about the composition of the object of the scene due to their high spectral resolution. The main goal of hyperspectral classification is the assignment of each pixel to the correct class. Hyperspectral classification accuracy highly depends on training data. To investigate this phenomenon, the classification methods are employed with different training data and sample size. Moreover, the improvements on classification performance by using spatial information with spectral information are also studied. The requirement of dimensionality reduction for hyperspectral image processing is also elaborated.

1.2 Scope and goal

This study is devoted to analyzing hyperspectral classification with Maximum Likelihood, Support Vector Machines, and K-Nearest Neighborhood algorithms. The first issue is determining the effect of different training data size and selection technique on classification accuracy. Another issue that is analyzed here is the effect of dimensionality reduction with Principal Component Analysis. Finally, the contribution of spatial information usage by filtering with 3x3 window and majority voting with meanshift segmentation is also analyzed and reported. Three different scenes (2 IKONOS and 1 ROSIS) are used for experimentation stage.

1.3 Outline of thesis

This thesis is organized as five chapters including background, methodology, experiments and conclusion. In Chapter 2, literature survey on hyperspectral image classification, dimensionality reduction and meanshift segmentation is presented. In Chapter 3, proposed classification models, training data selection methods, pre-processing method and post-processing methods are described. Chapter 4 presents experiments and results for each scene and classification model. Lastly, Chapter 5 presents the discussion and future work.

CHAPTER 2

Background

Remote sensing is the science that investigates and aims to improve methods to gather information about physical objects without physical contact. The sensors in this area can be classified into two categories as active and passive sensors. In the active type, sensing system sends and receives the reflected energy from the surface whereas in passive type, sensing device measures emitted energy from the surface [1].

In active remote sensing, source of energy is supplied from the remote sensing systems. Examples of these systems are RADAR (Radio Detection and Ranging), SONAR (Sound Navigation and Ranging) and LIDAR (Light Detection and Ranging). Imaging spectrometers and radiometers are examples of optical passive sensors. These sensors measure the reflected energy from the objects. Gamma rays, Xrays, UV, the visible, near infrared and short wave infrared regions are covered by optical remote sensing [2]. According to their spectral band coverage, optical remote sensing systems can be classified into three different categories. These are panchromatic imaging systems, multispectral imaging systems and hyperspectral imaging systems. One may further classify hyperspectral imaging systems as superspectral imaging systems, hyperspectral imaging systems and ultraspectral imaging systems.

Panchromatic imaging systems: These systems use a single channel sensor which is sensitive to broadband wavelengths. Panchromatic imaging systems measure the brightness of the materials so the output is a monochromatic (gray level) image. SPOT, IKONOS-PAN and QuickBird PAN can be given as examples for panchromatic imaging systems.

Multispectral imaging systems: These systems use three or more channel sensors which are collecting data from several selected bandwidths. Output images contain both color and brightness information. Each of these images can be displayed in different gray scale images. LANDSAT TM, MSS and QuickBird MS can be given as examples for multispectral imaging systems.

Superspectral imaging systems: These systems have typically more than 10 spectral channels. Superspectral imaging systems are generally used for observing land cover or vegetation regions [3]. Like multispectral imaging systems, superspectral imaging systems can be displayed

separately as grayscale images. Superspectral imaging sensors acquire images in narrower bandwidth. MODIS and MERIS can be given as examples for superspectral imaging systems.

Hyperspectral imaging systems: These systems have typically more than 100 spectral bands; the difference between two neighboring spectral bands is often less than 10nm. Furthermore, the spectral bands are consecutive. Hyperspectral imaging systems also save set of images in one data set which is called “data cube”. Hyperion and EO-1 satellites can be given as example for hyperspectral imaging systems [4].

Ultraspectral imaging systems: These systems have typically more than 500 spectral bands and high resolution. Molecular absorption or emission bands can be imaged by these devices [5].

Since the early 1980s, hyperspectral image sensors have been used in order to capture spectral information. Hyperspectral cameras, also called imaging spectrometers, combine properties of digital image cameras and spectroscopy devices. These systems are used for gathering spectral characteristics of materials. Figure 1 shows an example of spectral properties of some materials.

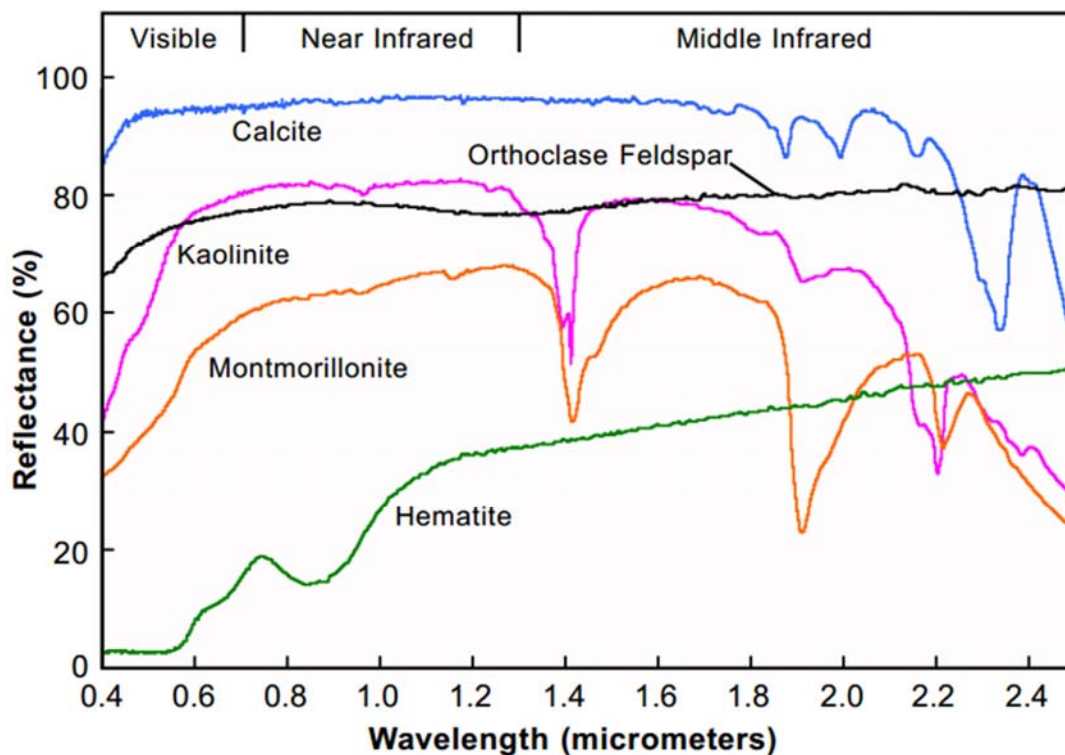


Figure 1 Reflectance spectra of some minerals adapted from [6]

There are many challenges in hyperspectral image processing. First of all due to changes in atmospheric conditions or sensor effect changes, spectral signature of the same material can vary from image to image. Secondly, since hyperspectral images have low spatial resolution and high

spectral resolution one pixel may have a mixture of two or more materials' spectra. Also since hyperspectral images have a multitude of spectral bands, these images may require too much storage capacity or processing power.

These problems can be attacked by different hyperspectral image processing methods. For the high-dimension problem, dimensionality reduction techniques can be used. These methods reduce the dimension with no or little loss of information. Hyperspectral unmixing techniques can be used to solve mixed pixel problem.

2.1 Literature Survey

In this thesis hyperspectral classification methods examined and compared. As a learning algorithms Maximum Likelihood (ML), K-Nearest Neighborhood (K-NN) and Support Vector Machines (SVM) used. Post-processing and pre-processing methods used to improve hyperspectral classification accuracy. In this study, PCA used as a pre-processing method. Majority voting and filter with window (3x3) and majority voting with meanshift segmentation has been used as post-processing.

2.1.1 Dimensionality Reduction

Higher dimensional data generally increases classification accuracy. However, according to Hughes phenomenon [7, 8], the required training sample size for classification grows exponentially as the number of spectral bands increase. Applications of hyperspectral image processing may require data volume or dimensionality reduction without loss of critical information. Dimensionality reduction also improves classification time. Saldju and David in [9], increase of dimension may lower classification accuracy with limited training data size. They conducted experiments with different numbers of training data for a decision tree classifier. Results show that for 100 training data, the classification accuracy is starts to decrease after 10 features. Hyperspectral classification accuracy is mainly dependent on training data size. To reduce training data size, reduction of the number of dimensions is required. Dimensionality reduction is also required to eliminate highly correlated bands.

In the literature, dimensionality reduction can be categorized as feature selection or feature extraction [11]. Feature selection methods are primarily based on eliminating bands that do not contribute to the hyperspectral image processing task. These methods simply eliminate the irrelevant and repetitive features to reduce the dimensionality from the original dimension M to N , where ($N < M$). In the literature, there are different methods for feature selection ([11, 12]). These reduction techniques select features by their relevance for the task and lack of correlation between the features. They differ from each other by their feature selection methods. Feature selection is reducing dimension by eliminating highly correlated bands, the feature extraction uses maximal statistical dependency criterion to select features to eliminate.

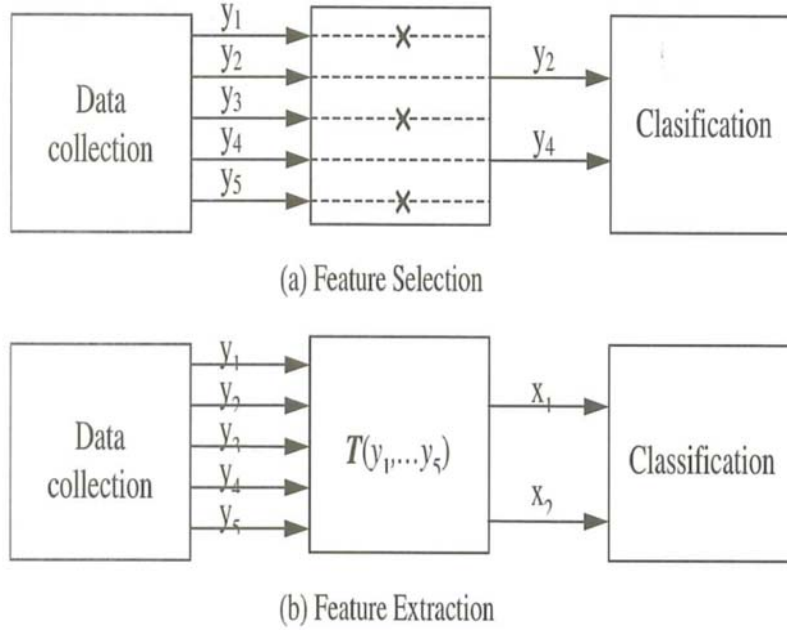


Figure 2 Dimensionality reduction diagram adapted from [12]

According to [14], feature extraction is the transformation of hyperspectral image from M dimensional space to N dimensional space where $N < M$. Principal Component Analysis (PCA) is one of the common feature extraction algorithms. ISOMAP, Factor Analysis, Linear Discriminant Analysis, Sammon mapping, Local Linear Embedding are other methods that are used for dimensionality reduction. Mathematical explanation of PCA is given in [15] and [16]. After applying PCA, the number of principal components can be selected using the principal values generated. These principal vectors can be used for classification. Total processing time is often significantly reduced with lower number of principal components [16].

2.1.2 Classification

Hyperspectral classification can be defined as giving a unique label class to represent each pixel by using its spectral information for every pixel vector in the hyperspectral image. In the literature, hyperspectral classification is grouped in two main categories as supervised and unsupervised classification methods. K-Mean and ISODATA [17] algorithms can be given as examples of unsupervised classification or clustering, algorithms. Clustering algorithms group or cluster the data by using their spatial and spectral properties without prior knowledge. Determining number of different classes in the image is a challenging issue. Supervised classification requires prior knowledge about data. Support Vector Machines (SVM) [18-22], K-Nearest Neighborhood (K-NN) [23-26], Gaussian Classifier (GC) [11], Maximum Likelihood [31-33] and Active Learning methods have been widely used for supervised hyperspectral classification.

2.1.2.1 Maximum Likelihood (ML)

In remote sensing Maximum Likelihood is one of the most popular methods. It has been used for many applications (remote sensing, statistics) since 1940. As a classification method it was firstly proposed by [34]. They used covariance matrices in order to derive likelihood estimate. They extend the work of [35] paper which compares two solutions of [36] and [37]. Maximum likelihood is derived from Bayes Theorem. Bayes theorem states that posterior probability that a pixel t belongs to class:

$$P(i | t) = \frac{P(t | i) P(i)}{P(t)}$$

Where $p(t)$ stands for prior probability of class t and $P(i)$ for prior information.

Maximum likelihood classification is used for multispectral data classification [38]. It is also used for hyperspectral data classification [39-40]. Both of these papers mentioned about dimensionality reduction and its necessity for better maximum likelihood classification accuracy. [39] used correlation between bands and [40] used PCA for dimensionality reduction.

2.1.2.2 K-Nearest Neighborhood (K-NN)

K-NN was introduced by [41] as a non-parametric method for pattern classification. It is very simple and easy to understand. In the literature, K-NN has been used for pattern recognition, speech recognition and remote sensing applications. The algorithm decides the class of the given pixel by majority vote of the pixels within a specific spectral distance from that pixel (in Euclidian distance determined by the k value). For every input vector, all class labels closer than maximum spectral distance is counted and the maximum numbered class is assigned for the input.

2.1.2.3 Support Vector Machines (SVM)

SVM provides high classification rates with small training datasets. In the literature SVM is used for many applications like pattern recognition [42-43], face recognition, handwritten digit recognition [44], object recognition [45, 46], fingerprint recognition [31] etc. Its formulation is presented in [47]. Since it is a very effective and simple method, SVM has been used for hyperspectral classification. SVM training algorithm tries to find the hyperplane which separates the dataset as much as possible. This is an iterative process that tries to separate training data with optimal decision boundary. Its simple form can be considered as a binary linear classifier. Every pixel in the hyperspectral image can be considered as a feature vector. Once the hyperplane is determined, SVM assigns each feature vector input to one of the classes. If the data points are not linearly separable, SVM can be used with non-linear functions named kernels. There are different kernels used in the literature: linear, polynomial, radial basis function (RBF) [48] and sigmoid are examples to that kernels. Formulations [49] of these kernels is given in (1): γ , r and d are kernel parameters.

Linear Kernel : $K(X_i, X_j) = X_i^T X_j$

Polynomial Kernel : $K(X_i, X_j) = (\gamma X_i^T X_j + r)^d, \gamma > 0$ (1)

Radial Basis Function (RBF) : $K(X_i, X_j) = f(-\gamma \|X_i - X_j\|^2), \gamma > 0$

Sigmoid : $K(X_i, X_j) = \tanh(\gamma X_i^T X_j + r)$

2.1.3 Post-Processing Methods

In order to improve hyperspectral classification performance, spatial information may be used with spectral information [50]. Where hyperspectral classification aims to assign a unique value to each pixel, usage of spatial-spectral classification improves classification accuracy by considering neighbor pixels. The main idea is a pixel is probably same material with the neighbor pixels. One approach for using neighbor pixels is using the closest ones. Fixed window based filtering method is used in [51] and morphological profile is used in [52]. They both showed improvements on classification accuracy. Segmentation is another approach for using spatial information. [50] proposed a method for hyperspectral classification with clustering techniques. As is seen in Figure 3, pixel-wise classification results are gathered with the labeled connected components.

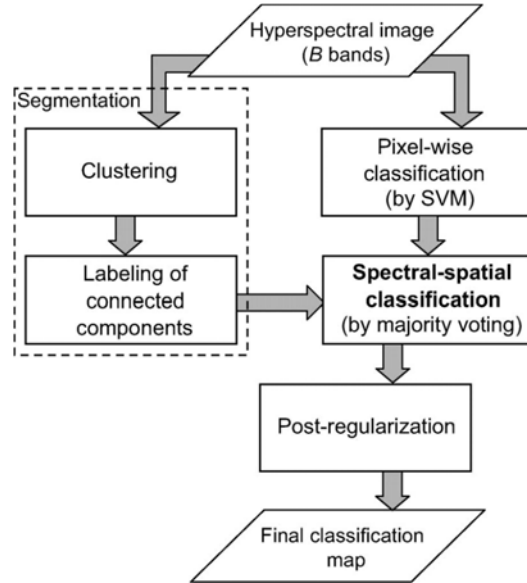


Figure 3 Flowchart of the proposed spectral-spatial classification scheme from [50]

CHAPTER 3

METHODOLOGY

3.1 Data

In this study, one high resolution and two low resolution hyperspectral images are used. Pavia University Scene is acquired by ROSIS sensor. Indian Pines and Salinas Scenes are acquired by AVIRIS sensor.

3.1.1 Pavia University Scene

Pavia University Scene (PUS) is acquired by ROSIS sensor which has a spectral range between 430-860nm in July 2002. This sensor has 115 spectral bands with 4nm bandwidth gap and 1.3 meter geometric resolution. PUS consists of 610x340 pixels. After discarding the image bands containing no information 103 bands are used for classification. PUS groundtruth has nine different classes. These classes and corresponding numbers of samples is presented in Table 1. The RGB image and groundtruth image can be seen in Figure 4. PUS is provided by Prof. Paulo Gamba of Pavia University.

Table 1 Pavia University Scene Groundtruth Classes and Sample Numbers

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

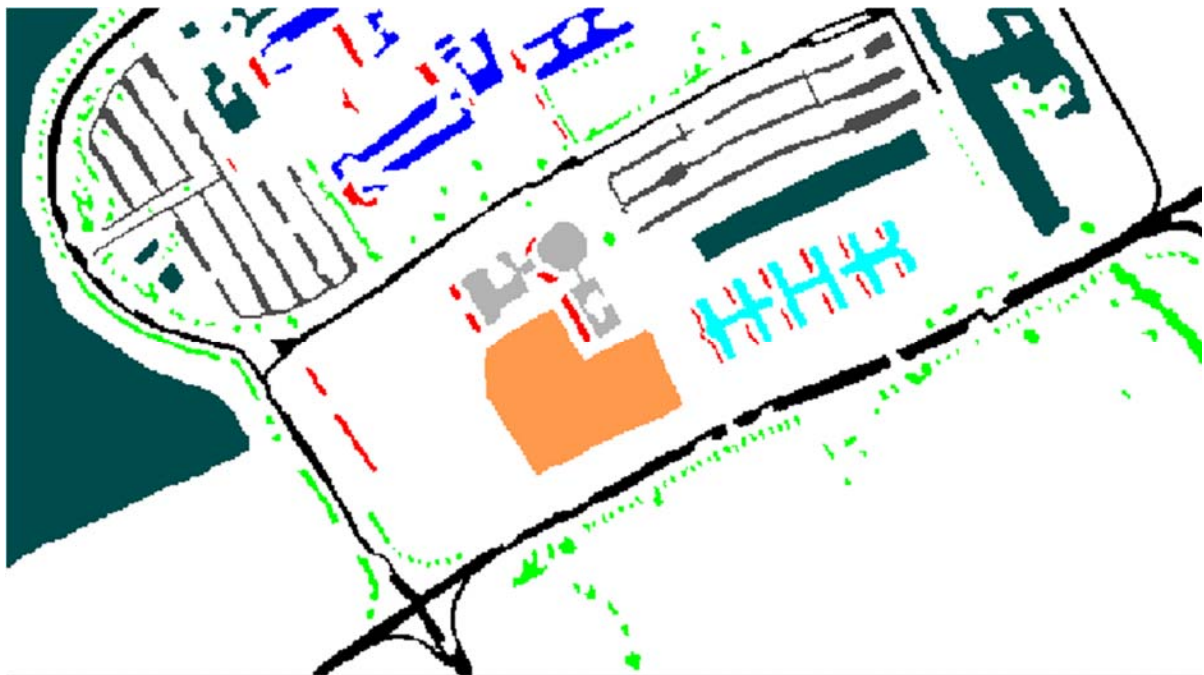


Figure 4 Pavia University Scene RGB Image(top) Groundtruth Image(bottom)(Courtesy of Prof. Paulo Gamba from Pavia University, Italy.)

3.1.2 *Indian Pines Scene*

Indian Pines Scene (IPS) is acquired by AVIRIS sensor which has a spectral range between 400-2500nm in 1992. It can gather images in 224 contiguous spectral channels. IPS has 145x145 pixels and 20 m spatial resolution. After water absorption bands ([108-112] and [154-167]) are removed, 200 bands are used for classification. There are 16 different classes in IPS. These classes and respective sample numbers is presented in Table 2. IPS is retrieved from Purdue University.

Table 2 Indian Pines Scene Groundtruth Classes and Sample Numbers

#	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

Since we examined the effect of increasing the training data size on classification accuracy we do not use classes with less than 380 samples. The RGB image and groundtruth image of IPS can be seen in Figure 5.



Figure 5 Indian Pines Scene (Left) RGB Image (Right) Groundtruth Image (Retrieved from <https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html>)(accessed 06.05.2013)

3.1.3 Salinas Scene

Salinas Scene (SSC) is also acquired by AVIRIS sensor in 1998. Salinas Scene has 512x217 pixels and 3.7 meter spatial resolution. As in the IPS case water absorption bands ([108-112] and [154-167]) are removed and 204 bands are used for classification. Salinas Scene also has 16 classes in its groundtruth. These classes and respective sample number can be seen in Table 3. RGB image and groundtruth of Salinas Scene can be seen in Figure 6. SSC is retrieved from Purdue University.

Table 3 Salinas Scene Groundtruth Classes and Sample Numbers

#	Class	Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

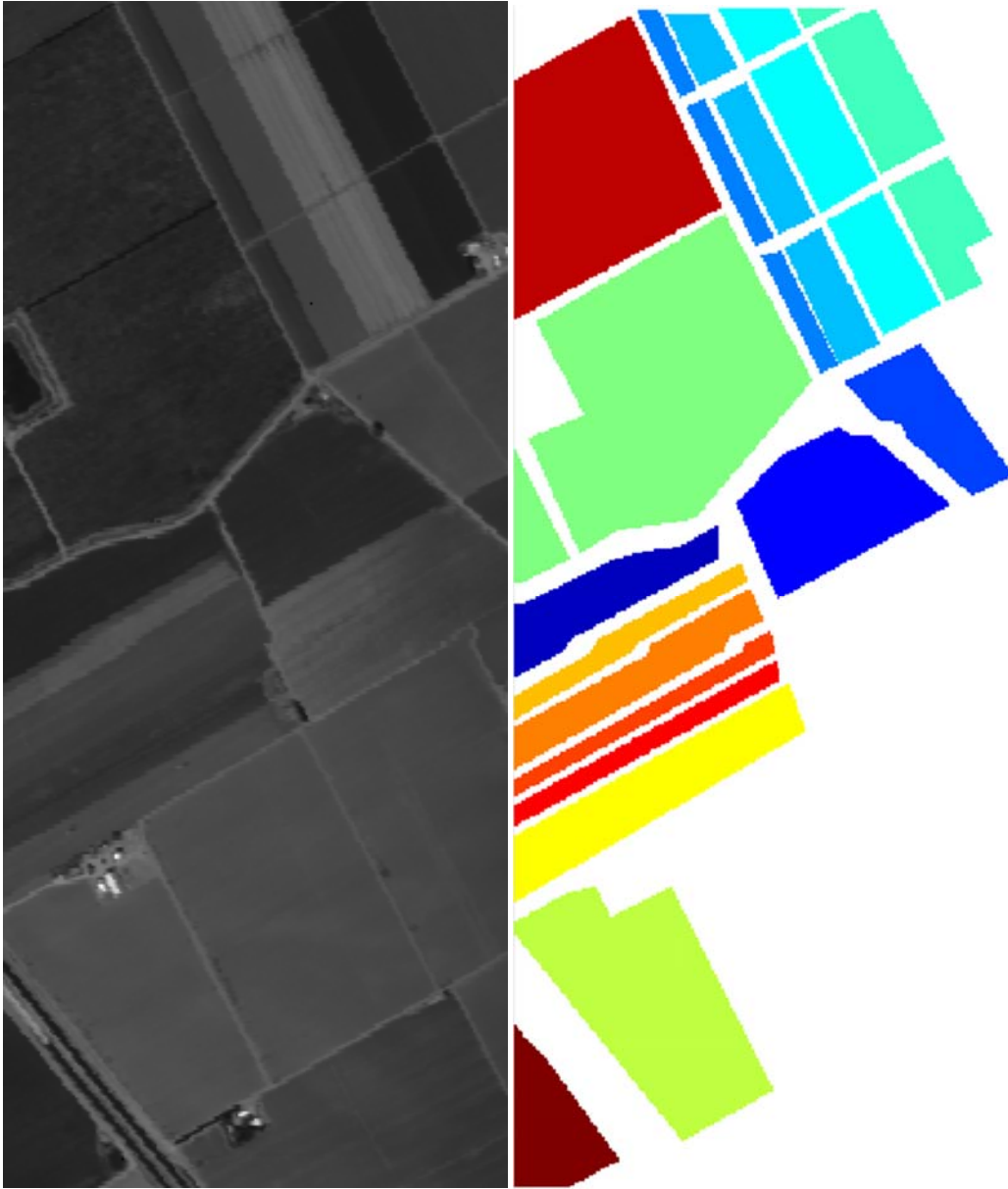


Figure 6 Salinas Scene (Left) Sample Band (Right) Groundtruth Image

3.2 Introduction

The main objective of this research is to investigate the effects of training data to supervised hyperspectral classification algorithms. In order to compare the results with the literature, training data is extracted from the image and used as external training data. Training data

selection is also another aspect of this research. We tried three different train data selection methods. These methods are described in Section 3.3.

The effect of usage of pre-processing with dimensionality reduction and post-processing with filtering with 3x3 window and majority voting with meanshift segmentation methods are also examined. The experiments include the both separate and joint usage of these steps. These experiments are described in Sections 3.4 and 3.5. In this study, we used PCA for dimensionality reduction as the pre-processing stage. We also used for meanshift segmentation and filtering with window as post-processing methods.

3.3 Training Data Selection

We selected training data by using three different methods. These experiments are important to indicate the effects of training data on supervised hyperspectral classification. First we extracted first N samples for all classes. As is seen in Figure 7(top), this method extracts training data from the left columns of the image. The selected training samples are expected to have similar spectral characteristics because the closer pixels have similar spectral signatures.

Second training data selection method is extracting (N) samples from image uniformly. Selected training data with this method is expected to represent all data from the image. The selection with this method can be seen in Figure 7 (middle).

Last training data selection method is extracting (N) samples of training data randomly for each class. This selection method is very similar to uniform selection method. Nevertheless, some small parts of data may not be represented in the training data with this method. The selection with this method can be seen in Figure 7(bottom).

For all these three methods, we also wanted to observe the effect of training sample size. We used and compared the results with different values of training data size, N. For PUS we used different N values that range between 120 and 920, for IPS between 220 and 380 and for SSC ranges between 220 and 900 per class. We incremented training data size by 10 for every iteration and obtained the classification results.



Figure 7 Training Sample Selection $N=700$ (top)First N Samples, (middle)Uniformly Selected Samples
(bottom) Randomly Selected Samples

3.4 Pre-Processing with PCA

In the context of this study, PCA implementation suggested by [53] for dimensionality reduction is employed. In this method, firstly, the hyperspectral image is normalized by subtracting its mean and the covariance matrix is calculated from the Equation (1).

$$Cov(x) = \frac{1}{N-1} X * X^T$$

Then eigendecomposition of the covariance matrix is computed from Equation (2). Let $E = [e_1 e_2 \dots e_n]$ and $\lambda = \lambda_1 \lambda_2 \dots \lambda_n$

$$Cov(x)E = Diag(\lambda)E \quad (2)$$

The eigenvalues and eigenvectors are sorted to get the principal components of the image. We used different numbers of principal components for all three scenes. For Pavia University Scene [2,3.....30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,103] and for Indian Pines and Salinas Scenes [2,3,.....30,35,45,55,65,75,85,95,100,120,140].

In Figure 8, flow chart of classification with Pre-processing usage can be seen. Firstly, the dimensions of hyperspectral image is reduced by PCA. After dimensionality reduction, train data extraction is performed by three different methods. By using these train data, learning step performed. Then, classification process carried out with the output of learning process and lower dimensionality image.

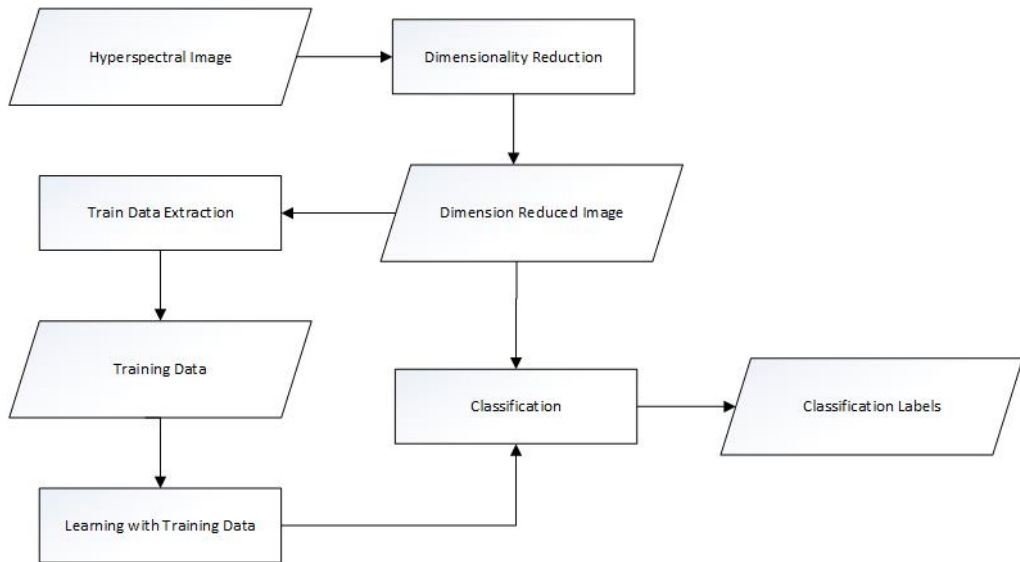


Figure 8 Flow Chart of Classification with Pre-Processing

Normal classification time for ML algorithm is 61.35 seconds. This time reduces to 7.31 seconds if two principal components are used. For SVM-LNR algorithm normal classification time 370 seconds, this time also reduces to 143 seconds. The classification time of SVM-RBF reduces from 501 seconds to 228 seconds.

3.5 Post Processing with Spatial Information

We used two different post processing methods for using spatial information with spectral information. We used majority voting with meanshift segmentation and 3x3 voting filter as post processing methods in this thesis. These methods are applied after performing pixel wise classification.

3.5.1 Filtering with 3x3 Window

We employed filtering with 3x3 window method in order to use spatial information. For every pixel, after we procured assigned class labels from classification algorithm, class labels are weighted as it is in Figure 9. We applied different weighting methods (

Figure 9).

Method 1			Method 2			Method 3			Method 4			Method 5		
1	1	1	0	1	0	1	1	1	0	2	0	1	2	1
1	4	1	1	6	1	1	1	1	2	4	2	2	4	2
1	1	1	0	1	0	1	1	1	0	2	0	1	2	1

Figure 9 Weighting methods for Filtering with 3x3 Window

The overall flow of our majority voting with 3x3 window filtering method algorithms as follows;

Get the classification result

Initialize coordinate (x,y)

Define the neighbor pixels

Count the class numbers by multiplying weights

Select the most frequent and assign it as coordinate value of (x,y)

For each pixel repeat the process

In Figure 10, the classification results for these methods can be seen. Except Method-2 all of these methods improved the classification performance. Although all of them give similar results, we observed that simple averaging (Method-3) is still the best method among all.

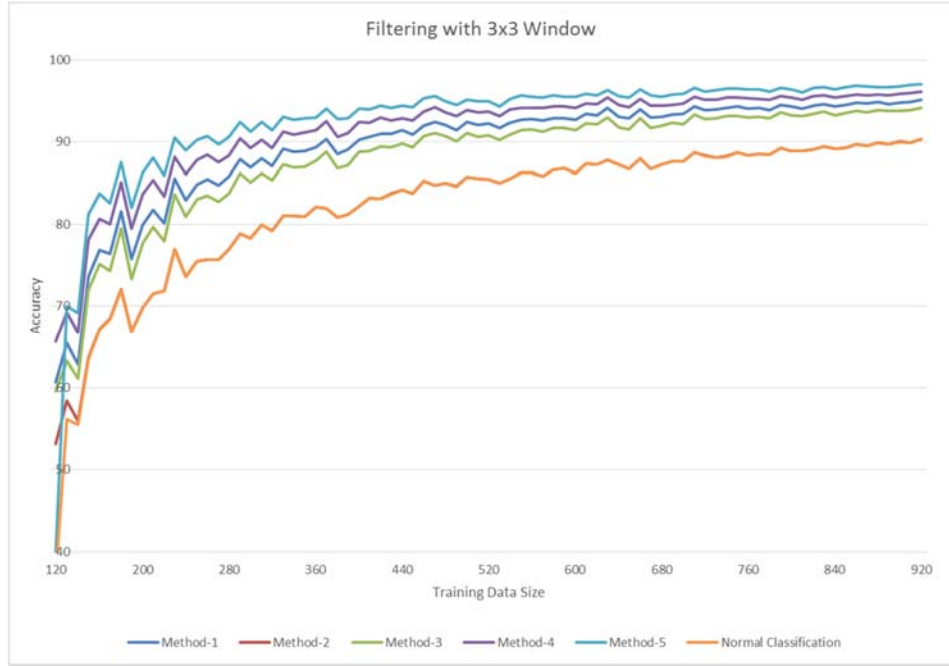


Figure 10 Different methods for filtering with 3x3 window method

3.5.2 Majority Voting with Meanshift Segmentation

We also used majority voting with meanshift segmentation. The majority voting method that we used is similar to [50]. First, we obtained the labels from meanshift segmentation. In order to achieve better segmentation results we used pattern search algorithm (Section 3.5.2.1) to find the best parameters.

The overall flow of majority voting with meanshift segmentation method algorithm as follows;

- Get the classification result
- Get segment labels from meanshift segmentation
- Count the class labels for each segment
- Select the most frequent class label and assign it to all pixels in that segment
- For each label repeat the process

The flow chart of post processing with meanshift segmentation can be seen in Figure 11.

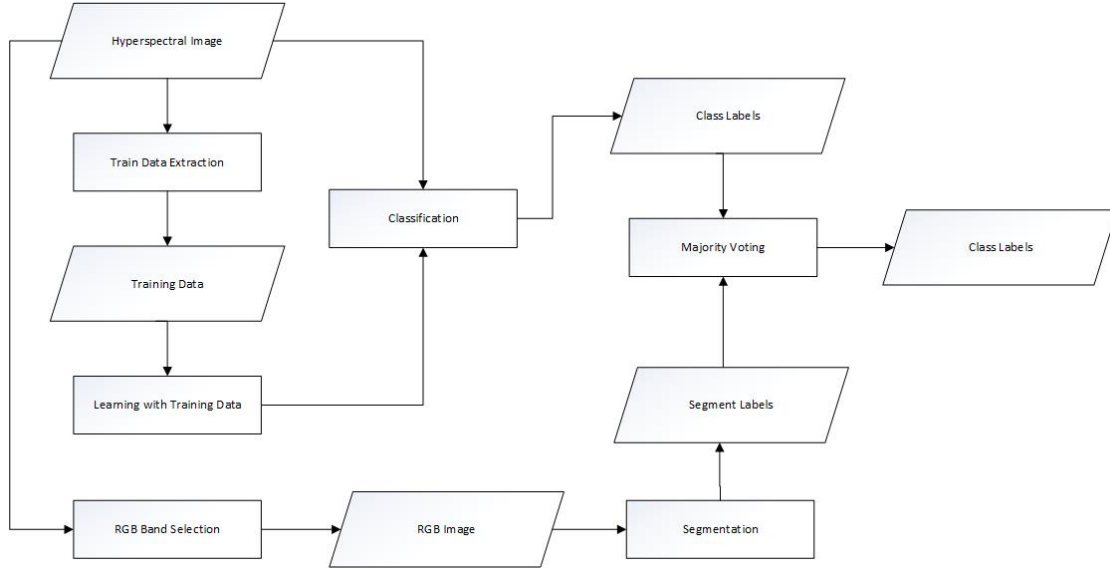


Figure 11 Post Processing with Meanshift Segmentation

For all three scenes, we also adjusted the parameters of the meanshift algorithm. Meanshift operation takes RGB image and feature function as input. Feature function has these parameters:

- SpatialBandWidth - segmentation spatial radius with default integer value 7
- RangeBandWidth - segmentation feature space radius with default float value 6.5
- MinimumRegionArea - minimum segment area with default integer value 20

We conducted experiments to set these parameters for all scenes. For PUS we used the Band 6 (462nm), Band 35 (594nm) and Band 90 (682nm). The best bands are also found by pattern search algorithm. For PUS, the best parameters are selected as “SpatialBandWidth = 18”, “RangeBandWidth = 6” and “MinimumRegionArea = 9”. The sample meanshift output for PUS with these parameters can be seen in Figure 12.

For IPS and SSC RGB bands are selected as 19 (620nm), 27(700nm), 33(760nm). The best parameters of meanshift for IPS are selected as “SpatialBandWidth = 1”, “RangeBandWidth = 9” and “MinimumRegionArea = 13” and for SSC are selected as “SpatialBandWidth = 36”, “RangeBandWidth = 57” and “MinimumRegionArea = 3”. The corresponding segmentation images for IPS (b) in Figure 13 and for SSC Figure 14.

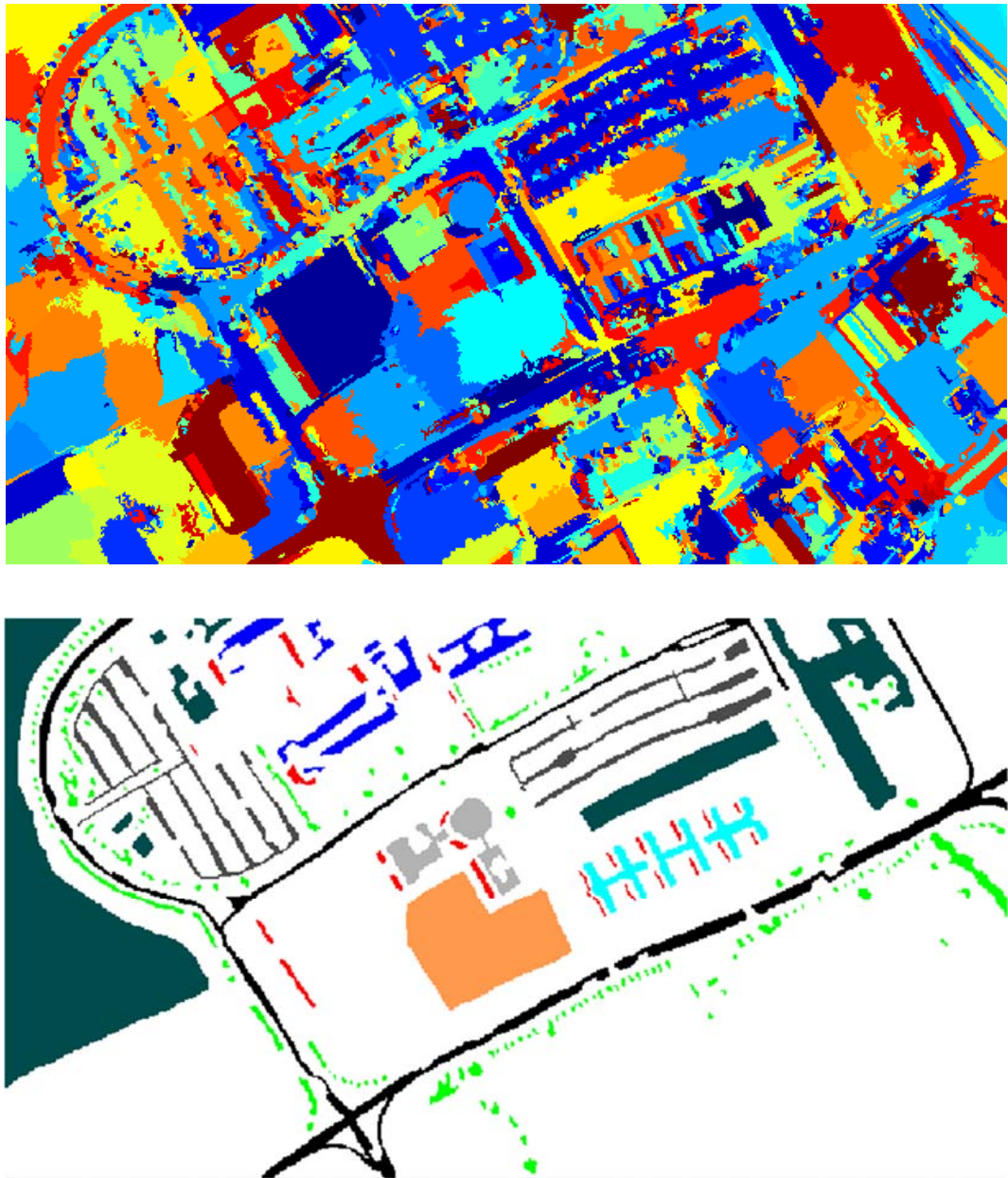


Figure 12 Pavia University Meanshift Segmentation Result (top) and Ground Truth (Bottom)

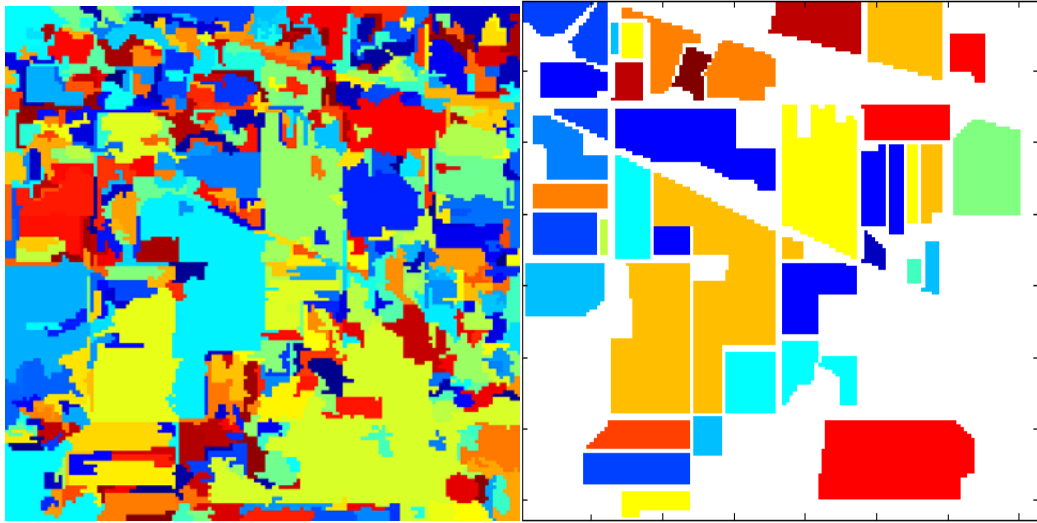


Figure 13 Segmentation Image (left) and Ground Truth (right) of Indian Pines Scene Salinas Scene

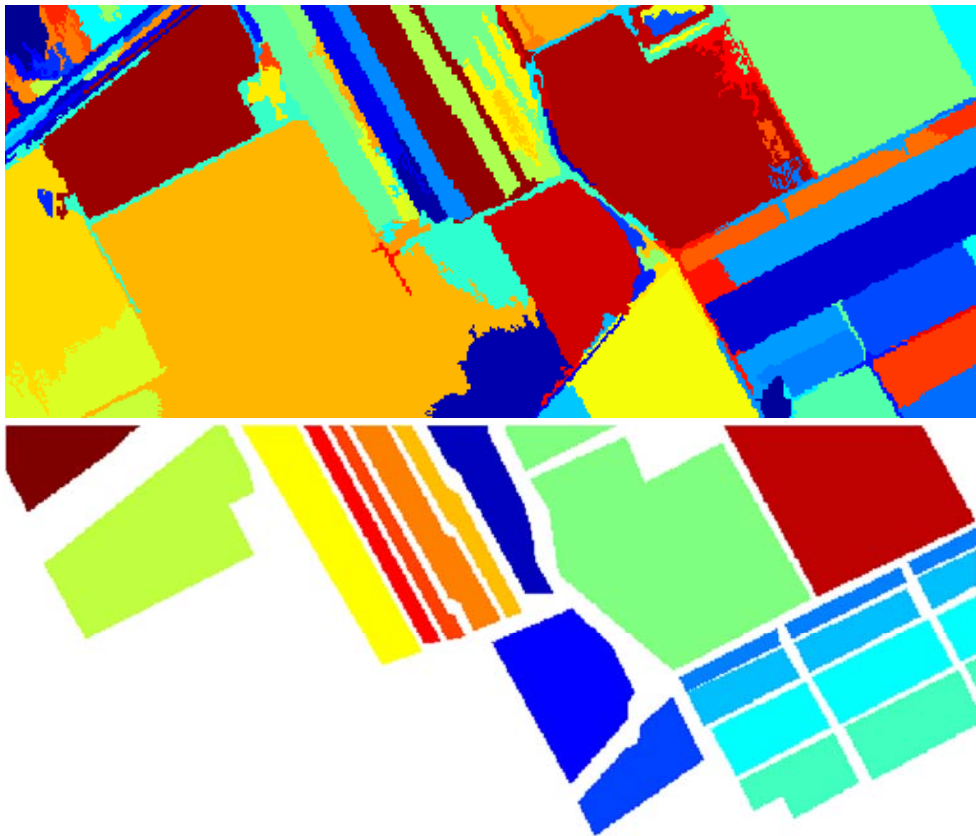


Figure 14 Segmentation Image (top) and Ground Truth (bottom) of Salinas Scene

3.5.2.1 Pattern Search

Pattern search algorithm is a method for direct search numerical optimization method proposed by Hooke and Jeeves [54]. Pattern search does not require the objective function to be continuous or differentiable. The convergence analysis of pattern search and the relation between optimality conditions and the search gradient are further detailed in the paper. [55]

Any linearly constrained optimization problem can be defined by using an objective function, $f(x)$, as given below;

$$\begin{aligned} & \min_{x \in \Omega} f(x), \text{ where } f: R^n \rightarrow R \\ & \text{and } \Omega = \{x \in R^n : \ell \leq Ax \leq u\}, \text{ where } A \in Q^{m \times n} \\ & \text{and } \ell, u \in \{R\}^m \text{ provided that } \ell < u \end{aligned}$$

The main aim of pattern search algorithm is to minimize the objective function in the space of feasible solutions, Ω . Search space is bounded by lower and upper bounds, ℓ and u , respectively.

In the study [56], a barrier function, defined as $f_\Omega(x) = f + \psi_\Omega$, is employed instead of $f(x)$ by introducing an indicator function, ψ_Ω , for feasible solution set Ω . In other words, ψ_Ω has zero value on Ω and equal to ∞ elsewhere, which is shown below;

$$f_\Omega(x) = \begin{cases} f(x) & \text{if } x \in \Omega \\ \infty & \text{otherwise} \end{cases}$$

Since the pattern search is an iterative algorithm visiting instances of solutions, $\{x_n\} \in R^n$, non-increasing objective function values are required to proceed. The optimization procedure is composed of two distinct stages; SEARCH and local POLL.

In SEARCH stage, the algorithm seeks a better solution minimizing the barrier objective function. At each step the objective function is evaluated by a finite number of points on a mesh. In case of reaching a lower objective function value, $f_\Omega(x_{k+1}) < f_\Omega(x_k)$, an improved mesh point is obtained. Otherwise, the algorithm steps into POLL stage in which an optimum solution is searched in the neighborhood of the current mesh point. The current best solution for a mesh local optimizer is identified, unless POLL routine reaches a better solution. Then, the mesh size parameter Δ_k is updated by a pre-defined constant τ as follows;

$$\Delta_{k+1} = \tau^{w_k} \Delta_k$$

As a result, the mesh about to be explored by pattern search at iteration k can be defined as;

$$M_k = \{x_k + \Delta_k D_z : z \in Z^+\}$$

In that formulation, D represents positive spanning directions in R^n .

Moreover, the gradient of the problem is not necessarily needed to reach the global minimum as it is proved in the paper[57] that pattern search algorithm holds the global convergence property.

3.6 Post-Processing and Pre-Processing

Section 3.4 and 3.5 shows the usage of post-processing and pre-processing. We also conduct experiments to see the effect of both post-processing and pre-processing usage. We used both meanshift segmentation and 3x3 voting filter as post-processing methods. The overall flow of the usage pre-processing and post-processing method together as follows;

- Apply PCA as pre-processing
- Get the classification result from principal components
- Get labels from meanshift segmentation
- Count the class labels for all 3x3 window
- Select the most frequent class labels from classification labels
- Assign it for all pixels in that segment
- For each label repeat the process

The flow chart of that process can be seen in Figure 15.

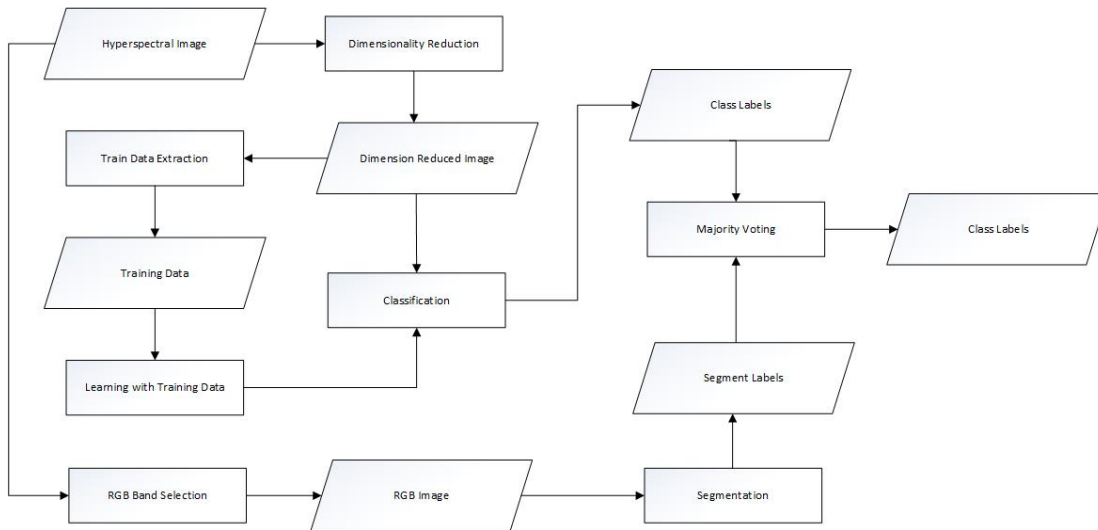


Figure 15 Flow chart of pre-processing and post-processing usage for classification

CHAPTER 4

EXPERIMENTS

In this chapter, empirical classification results are demonstrated. The effects of using pre-processing and post-processing are examined separately. In Sections 4.2, 4.3 and 4.4 the empirical results of PUS, SSC and IPS are shown. We grouped the experiments as no-pre-processing and no-post-processing, pre-processing with no-post-processing, no-pre-processing with post-processing and pre-processing with post-processing methods.

4.1 Measurement Metrics

There are several methods for assessing the performance of hyperspectral classification. In this study, we used classification accuracy as the classification metric. Classification accuracy is the fraction of truly classified data points to the total number of classified pixels. Classification error can be calculated as in equation (4.1)

$$\text{Classification Accuracy} = \frac{\text{Number of Truly Classified Pixels}}{\text{Number of Available Pixels}} \quad (4.1)$$

We also calculated other indicators to assess the performance by material with precision, recall and F-Measure metrics. These indicators are;

True Positives (TP): The number of correctly labeled pixels.

True Negatives (TN): The number of correctly labeled pixels belonging to other classes.

False Positives (FP): The number of incorrectly labeled pixels belonging to class.

False Negatives (FN): The number of incorrectly classified pixels belonging to other class.

Precision is the fraction of the number of correctly labeled pixels to both correctly and incorrectly classified pixels.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is the fraction of correctly classified pixels to correctly classified pixels with incorrectly classified pixels.

$$Recall = \frac{TP}{TP + FN}$$

F-measure is the harmonic mean of precision and recall.

$$F_{\beta} = (1 + \beta) \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$$

The β parameter is the weight of precision and recall. The precision and recall values are deemed more important when β value is below and above 1, respectively.

4.2 Indian Pines Scene Experiments

The first objective of this section is to analyze the effect of train data on classification accuracy. We also analyzed the effect of pre- and post-processing usage on classification accuracy. Firstly, we implemented and tested all the classification algorithms that we used (ML, K-NN, SVM-LNR and SVM-RBF). Detailed explanation of this process is provided in Chapter 3. The confusion matrixes for IPS are given in Appendix A.

4.2.1 Indian Pines Scene with No Pre-Processing and No Post-Processing

We implemented different training data size for all training data selection types. In order to see the effect of training data size to classification algorithms, we conducted the experiments with different training data size. As we stated in Section 3.3, we extracted training data from the scene itself by using groundtruth. We used this extracted data as separate training data.

As the first method, we used the first N samples as the training data. With this method the average classification accuracies are between 55% and 65% for all algorithms. In Figure 16, first N sample classification accuracies can be seen. The training data size affects ML algorithm more than the other algorithms. As the training data size increases, ML algorithms classification results are also increasing. For all algorithms that we used, training data size affected the classification accuracy in a positive manner. By using 220 training data, ML algorithms classification accuracy is 48.10%. When we increased training data size to 380, the accuracy increases to 69.33%. On the other hand, K-NN algorithm increases by approximately 8% (60.10% - 68.02%), SVM-LNR increases by %3 (66.83% - 70.90%) and SVM-RBF increases by %4 (57.71% - 67.71%). SVM-LNR obtained the highest accuracies for all training data sizes with first N sample selection method.

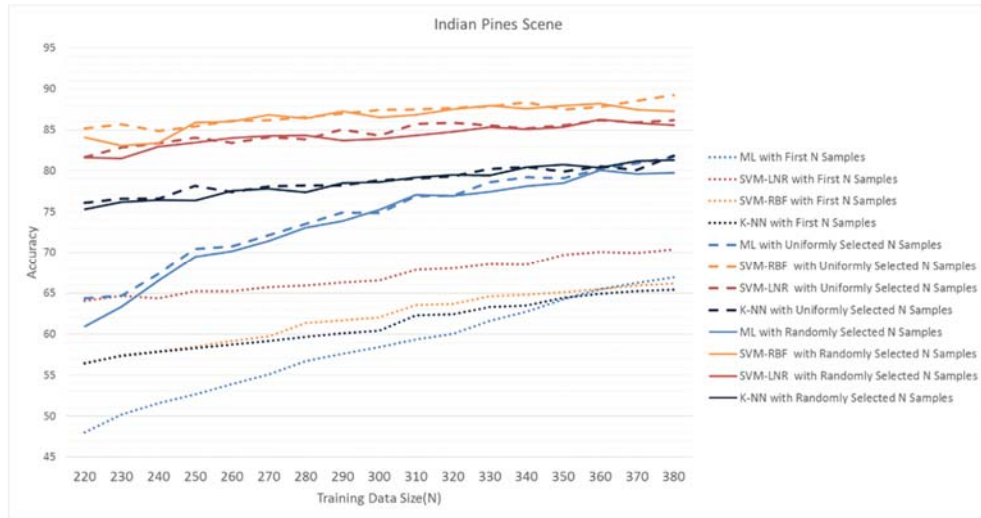


Figure 16 Indian Pines Scene Classification Accuracies

Second training data selection method for IPS is uniformly selected N samples from scene. The change of sample selection method increased the classification accuracies for all algorithms. With ML algorithm and 220 training data, the classification accuracy is 64.39%. By using 380 training data, the classification accuracy is increased to 81.73%. ML algorithm is affected by training data size like first N sample selection method, but training sample selection method change altered the classification result by 15%. Other algorithms are not affected by the increase of the training data size. K-NN algorithm increases approximately by 5% (76.06% - 81.83%), SVM-LNR increases by 3% (81.63% - 86.20%) and SVM-RBF increases by 4% (85.16% - 89.24%). However, the change of training sample selection method mostly improved the SVM-RBF algorithm. The best classification results with SVM-LNR and SVM-RBF are obtained with first N sample selection and uniformly selected N sample methods, respectively. For all training data sizes with uniformly selected N samples, SVM-RBF achieved higher classification accuracies. SVM-LNR and K-NN algorithms improved the classification accuracy approximately 20% with first N sample and randomly selected N sample methods for all training data sizes over first N sample method. The classification accuracies of randomly selected N sample method are similar to uniformly selected N sample method. The classification accuracies may differ by 1%.

In Figure 17, the classification results for ML algorithm with 380 training sample is presented. The misclassified pixels for first N sample selection method (left) generally take place on the right side of the classes as expected. Furthermore, the misclassified pixels in uniformly selected N sample selection method (right) are dispersed uniformly over the classes. First N sample selection and uniformly selected N sample selection methods acquired 66.91% and 81.74% accuracy, respectively.

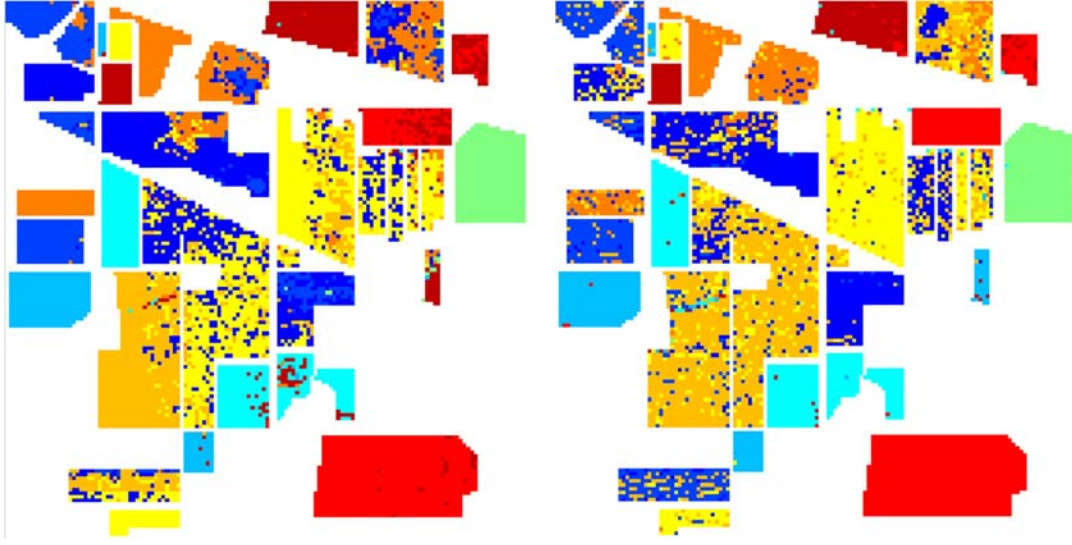


Figure 17 Indian Pines Scene (left) First N (N=380) Sample Classification Result (right) Uniformly Selected N (N=380) Sample Classification Result

4.2.2 Indian Pines Scene with Pre-Processing Only

The second stage of IPS experiments is using PCA as the pre-processing step. We used different number of principal components for classification algorithms. The aim of PCA step is to represent the data more efficiently and reduce classification time by using low number of principal components. In general, PCA usage reduces classification time, but in small scenes like IPS the difference is not significant.

The training size selection method affected the classification accuracy very similar to Section 4.2.1. The accuracies with selection of first N samples method is approximately 10% lower than both uniformly selected N samples and randomly selected N samples methods. In Figure 18, effect of training data selection method can be seen clearly.

As the number of principal components increase, ML algorithm classification accuracy is improved when training data size is increased. The difference between minimum training data size (220) and the maximum training data size (380) is approximately 4% for lower (<20) number of principal components. However this improvement is 10% for 140 principal components. As is seen in Figure 19, the classification accuracy is increasing with the number of principal components. By using ML algorithm, we obtained 50.35% classification accuracy with two principal components.

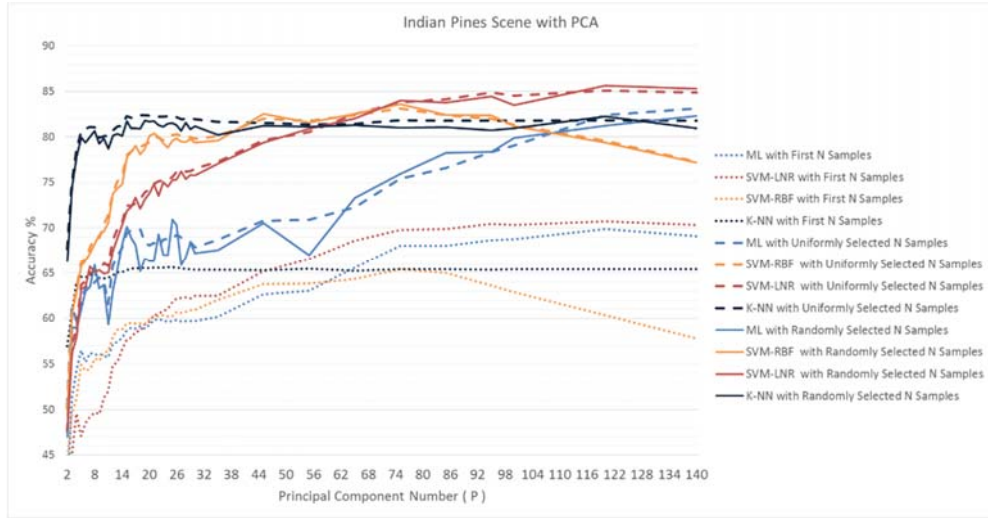


Figure 18 Indian Pines Scene classification with PCA

The accuracy with 140 principal components is 83.12%. Moreover, when we compared the classification accuracies without PCA, the classification accuracies for all training data size are improved by 2-8%. We obtained the best accuracies by using lower number of principal components with K-NN algorithm. After 18 principal components, the accuracy of the K-NN algorithm is not improved significantly with the increase of the number of principal components. In addition, K-NN gives the best results for all principal component numbers. The pre-processing usage does not affect K-NN's classification accuracy over not using pre-processing. The increase of principal component number affects SVM-LNR and SVM-RBF classification accuracy positively. Both algorithms gave better results with higher number of principal components. However, for SVM-RBF and SVM-LNR the classification accuracy for all training data sizes are decreasing if pre-processing with PCA step is used. This decrease can be defined as SVM-RBF on the average 6% and SVM-LNR on the average 2%.

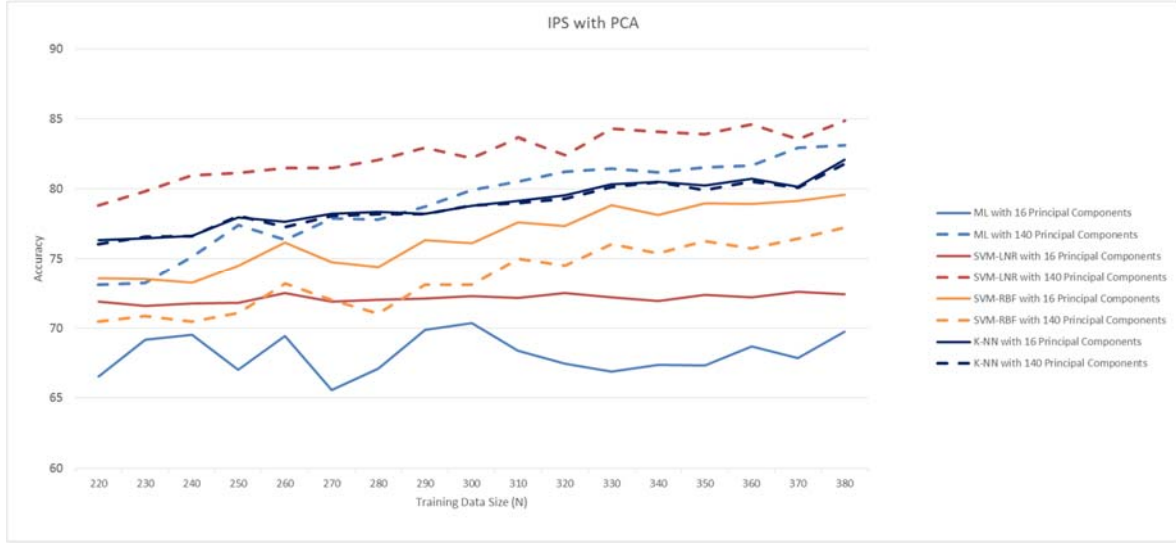


Figure 19 IPS with different number of principal components

4.2.3 Indian Pines Scene with Post-Processing Only

As we stated before, we used majority voting with meanshift segmentation (Section 3.5.2) and filtering with 3x3 window methods (Section 3.5.1) as post processing methods. The aim of this step is to improve hyperspectral classification accuracy by using spatial information. In Figure 20 the results for filtering with 3x3 window results and in Figure 21 the results for majority voting with meanshift segmentation results can be seen. The results show that both methods improve the classification accuracy.

Using majority filtering with 3x3 window as post processing method enhances the classification results for all training data sizes. The enhancement is approximately 16% for lower training data sizes and 11% for higher training data sizes. Usage of this method also enhances the classification results for other algorithms. The best result for ML algorithm is 92.54% with 360, for SVM-LNR algorithm is 93.62% with 360 training data, for SVM-RBF algorithm is 94.29% with 380 training data and for K-NN algorithm is 93.30% with 380 training data. These results show that the usage of spatial information with filtering 3x3 window improves the classification rates 10.8% for ML, 5.05% for SVM-RBF, 7.42% for SVM_LNR and 11.47% for K-NN algorithms.

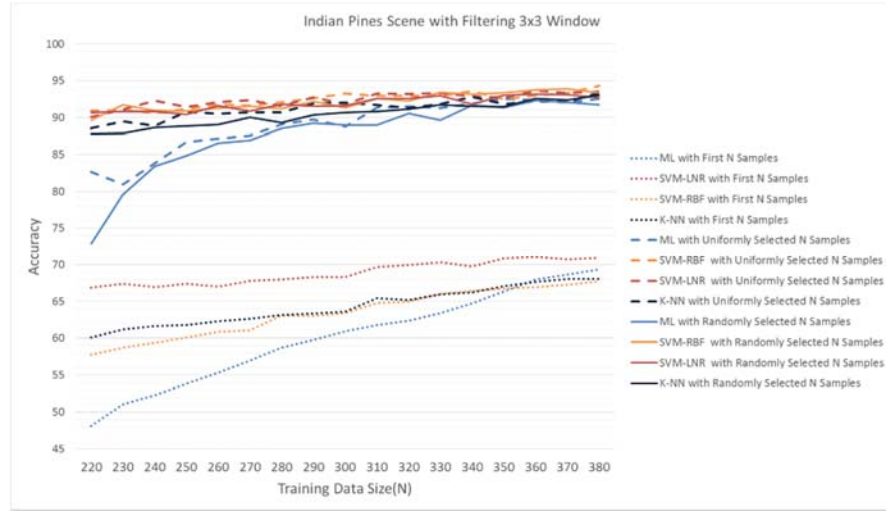


Figure 20 IPS with Filtering 3x3 Window

Moreover, the best results for IPS are obtained by using majority voting with meanshift segmentation as the post processing method. The advantage of this method is that it provides higher accuracy with lower training data size. With 220 training data size, ML algorithm classification accuracy reaches 91.92%. Spatial information usage brings not only higher accuracy but also stable classification accuracy for all number of training data. Majority voting with meanshift segmentation usage improves all algorithms classification accuracy. For example, the best classification accuracy for ML 93.18% with 340 train data, for SVM-LNR 95.05% with 310 training data, for SVM-RBF 94.36% with 370 training data and for K-NN 95.38% with 380 training data. Using this method improves the classification accuracy 11.44% for ML, 5.11% for SVM-LNR, 8.7% for SVM-RBF and 13.55% for K-NN.

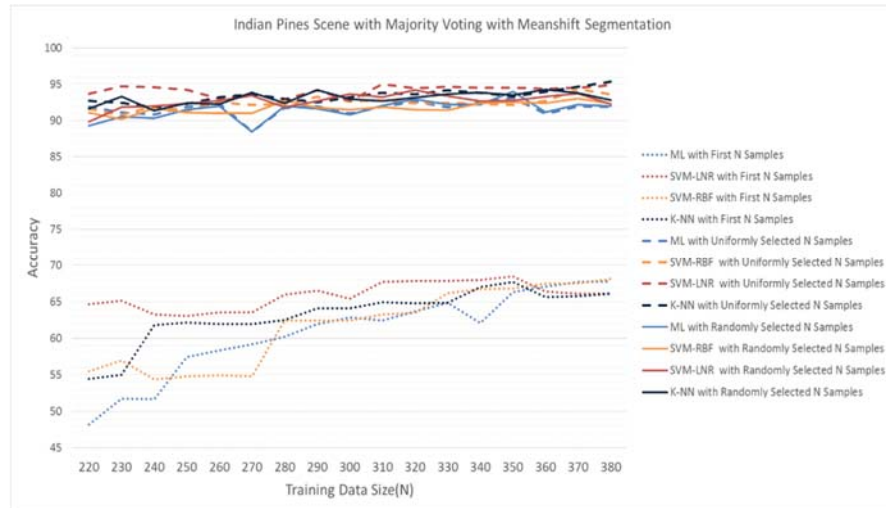


Figure 21 IPS Classification Results for Majority Voting with Meanshift Segmentation

4.2.4 Indian Pines Scene with Post-Processing and Pre-Processing

Last stage of IPS experiments is performing hyperspectral classification first pre-processing with PCA. After PCA, classification process is carried out. Lastly, filtering with 3x3 window and majority voting with meanshift segmentation as post-processing is performed. Previous experiments show that, whereas usage of the PCA reduces dimension and classification time, majority voting with meanshift segmentation is increasing classification accuracy. These experiments aim to investigate and explain the effects of pre-processing and post-processing usage for hyperspectral classification. Detailed information about this step is given in Section 3.6.

According to the experiments in Section 4.2.2., PCA usage is affecting classification algorithms differently. For SVM-LNR, SVM-RBF and ML algorithms pre-processing usage is not improving the classification accuracy. However, even with lower training data sizes, K-NN algorithm improves the classification accuracy when more than six principal components are used.

Pre-processing with PCA and post-processing as majority voting with meanshift segmentation (Figure 22) also improves the classification accuracy of classification with PCA and without post-processing method. Similarly with filtering with 3x3 window method, SVM-RBF obtained better results if PCA is not used. On the other hand, ML algorithm generally obtains better classification accuracies if PCA is not used. The usage of PCA is reducing the classification accuracy for SVM-RBF approximately by 4%. K-NN algorithm obtains similar classification accuracies with the accuracies obtained in Section 4.2.3. PCA usage affects K-NN algorithm better than other classification algorithms. The reason of that K-NN algorithm is classifying with nearest neighborhood rule so working in lower dimensional feature space provides an advantage.

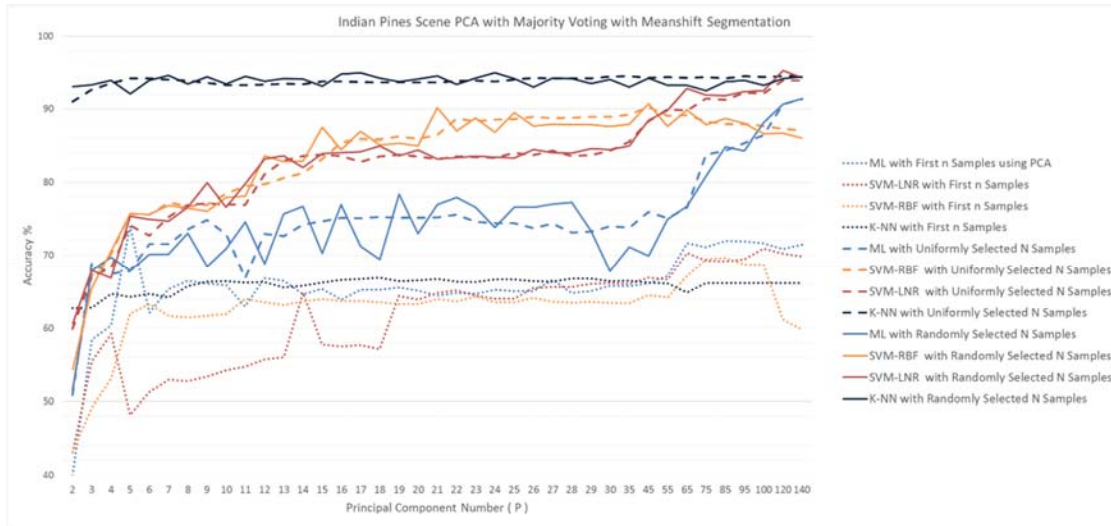


Figure 22 IPS with Majority Voting with Meanshift Segmentation

Pre-processing with PCA and post-processing as majority voting with meanshift segmentation (Figure 22) also improves the classification accuracy of classification with PCA and without post-processing method. Similarly with filtering with 3x3 window method, SVM-RBF obtained better results if PCA is not used. On the other hand, ML algorithm classification accuracies generally obtains better classification accuracies if PCA is not used. The usage of PCA is reducing the classification accuracy for SVM-RBF approximately 4%. K-NN algorithm obtains similar classification accuracies with the accuracies obtained in Section 0. PCA usage affects K-NN algorithm better than other classification algorithms. The reason of that K-NN algorithm is classifying with nearest neighborhood rule so working in lower dimensional feature space provides an advantage.

4.3 Salinas Scene Experiments

In this section we also analyze the effect of train data to classification accuracy on Salinas Scene. We also analyzed the effect of pre- and post-processing usage to classification accuracy on the same scene. Then as in Section 4.2 we implemented all the classification algorithms that we used (K-NN, ML, SVM-LNR and SVM-RBF). SSC experiments are conducted similarly with IPS experiments. SSC and IPS are acquired from same sensor but SSC is larger than IPS. We compared the results for both scenes. The confusion matrixes for SSC are given in Appendix B.

4.3.1 Salinas Scene without Pre-Processing and Post-Processing

In the same way that we conducted the experiments for IPS, the results of these experiments are analyzed by training data size and algorithm. Training data extraction methods are also same. The behaviors of each training data selection methods are examined for all algorithms.

The accuracies of the selection training data as first N samples method is around 75-80%. Training data selection method affects the Salinas data in the same way with IPS. Uniform and random selection methods acquire higher accuracies than first N sample selection method for all training data sizes. Nevertheless the accuracies obtained with first N sample training data selection method are higher than IPS. This may be due to spectral signatures of the materials in SSC being more differentiable than IPS. So, the classification results for SSC are expected to perform better than that of IPS. The Figure 23, shows that the classification accuracies can reach above 90% without pre-processing or post-processing.

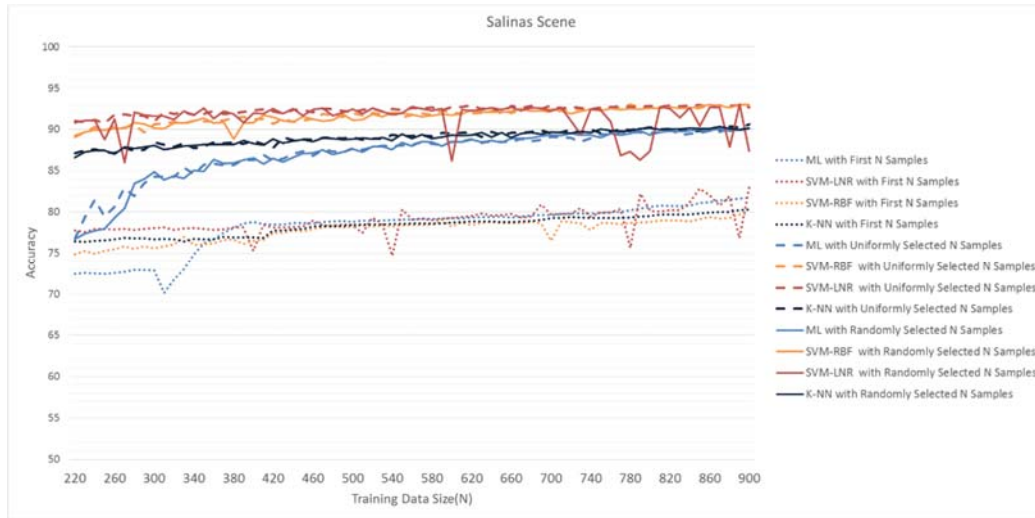


Figure 23 Salinas Scene Classification Results without Pre-Processing and Post-Processing

When we observe the results on the basis of algorithms, ML algorithm increases its classification accuracy as the training data size increases. SVM-LNR, SVM-RBF and K-NN are not affected as much from the increase of training data size. From these three algorithms, SVM-RBF obtains better classification results than others. It obtains 93.03% classification accuracy with 780 training data. On the other hand SVM-LNR obtains 92.96% classification accuracy with 890 training data. Lastly, K-NN obtains 90.62% classification accuracy with 900 training data. ML acquired lowest accuracy (90.06%) among all algorithms.

IPS and SSC have the same patterns for all classification algorithms that we applied. This situation is available for all training data selection models. Additionally the classification accuracies of SSC are better than classification accuracies of IPS. The training data selection methods also acquired similar results. First N sample selection method acquired worse results than uniformly N sample selection and randomly N sample selection methods.

Figure 24 shows the classification results for SSC uniformly selected N samples (top), first N samples from right side of the image (middle) and last N samples from left side of the image (bottom). Similar with IPS, the misclassified pixels with uniformly selected N sample selection method are dispersed through all the classes. However, right part of the 8th class denoted by green (grape-untrained) and left part of the 15th class denoted by red (vineyard-untrained) have very similar spectral characteristics. With first N sample selection method where we select training samples from left side of the classes, training samples for 15th class are very similar to samples from 8th class. As a result, the classification accuracy for 15th class decreases to 3%. Similarly, when we select training samples from right side of the classes, the accuracy of the 8th class decreases to 20%. This is the main reason for the sudden decreases of SVM-LNR with random N sample selection method in Figure 23.

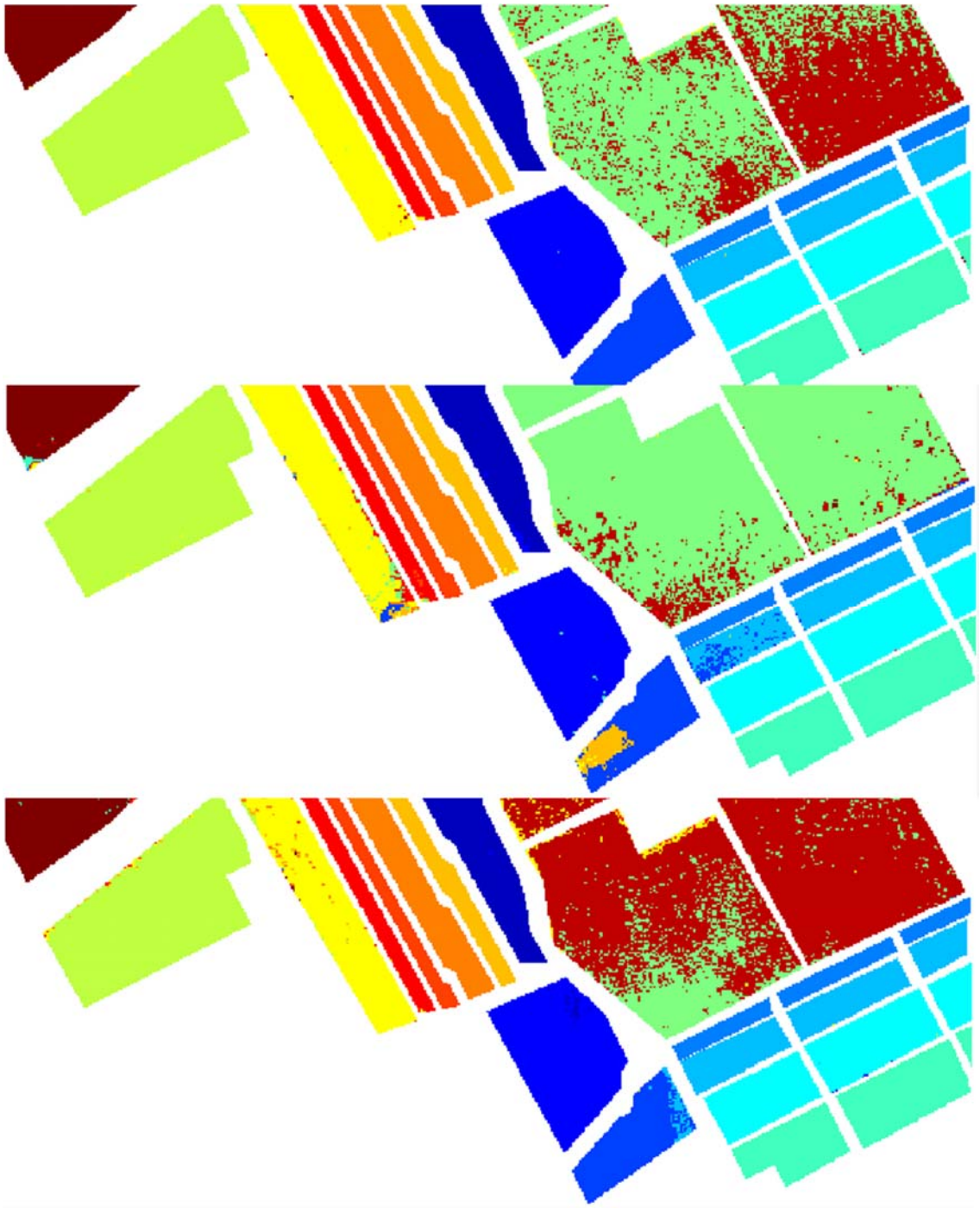


Figure 24 Salinas Scene Classification Results $N=900$ (top) Uniformly Selected N Samples (middle) First N Samples from right side of the image (bottom) Last N Samples from left side of the image

4.3.2 Salinas Scene with Pre-Processing

PCA usage as a pre-processing method does not affect IPS as classification time. However, SSC is affected by the usage of PCA as classification time when lower number of principal components is used (Figure 25). The complete classification time for ML algorithm is 44.53 seconds, however the classification time reduces to 4.68 seconds if two principal components are used for classification. Similarly, the classification time for SVM-LNR reduces from 272.9 seconds to 102.85 seconds, for SVM-RBF from 404.71 seconds to 180.15 seconds. Pre-processing usage mostly affects K-NN algorithm, the classification time of K-NN reduces from 356.6 seconds to 3.05 seconds. This may be due to the fact that K-NN algorithm directly measures the distance in feature space, so after several principal components adding new principal components does not affect the distance between neighborhoods.

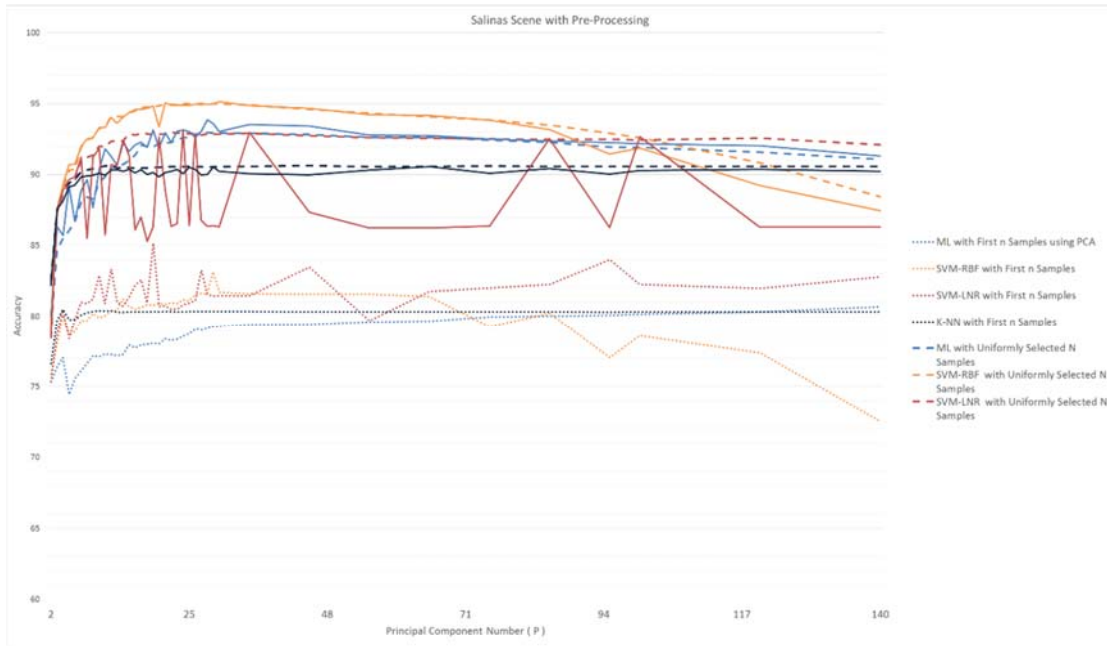


Figure 25 SSC with Pre-Processing

In IPS, the increase of the number of principal components affected classification accuracy positively. However in SSC the increase of the number of principal component decreases the classification accuracy. In Figure 26, the dashed lines are the results with 140 principal components and the solid lines are the results for 16 principal components. These results show that using 16 principal components improves the classification accuracy by 10% for SVM-RBF, 5% for ML and SVM-LNR. K-NN results are very similar for 16 and 140 principal components.

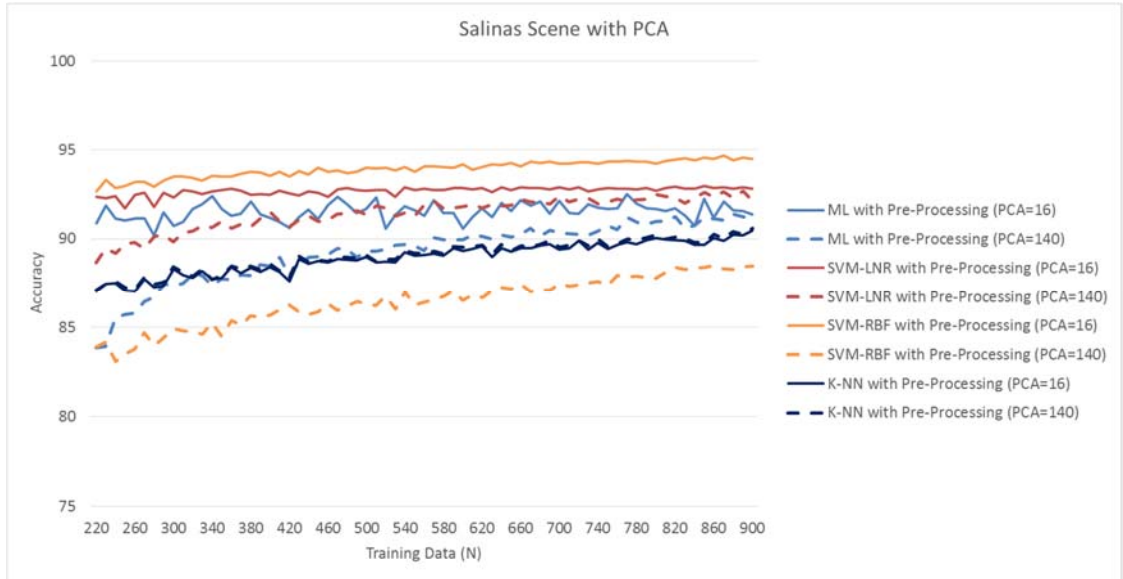


Figure 26 Salinas Scene classification with PCA results

Similarly with IPS, PCA usage also improves ML the classification accuracy. Classification accuracy for 16 principal components and 220 training data is 90.88%, whereas the classification accuracy without PCA and same number of training data is 76.78%. PCA usage provides an advantage for lower training data sizes. The difference for 900 training data size is 1.07% (PCA-91.06% and Without PCA - 89.99%). Similarly with previous experiment results, ML algorithm obtains better results with higher train data sizes for higher dimensions. When 140 principal components are used, the classification results are getting worse. Classification accuracy reduces by 10% for lower training data sizes, however, for 900 training data the classification accuracies are almost the same. The number of principal component mostly affects SVM-RBF algorithm. We obtained 92.68% classification accuracy with 16 principal components and 220 training data, whereas the classification accuracy for 140 principal components and 220 training data is 83.87%. The highest and the lowest accuracies are obtained with SVM-RBF. The classification rates of SVM-RBF increase until the 24-30 principal components, then it starts to decrease gradually. Random and uniform sample selection methods yielded similar accuracies.

Random selection method has more influence on SVM-LNR (Figure 27). Classification accuracies may change by 5%-10% for 10 training data change. SVM-LNR, SVM-RBF and K-NN acquire similar classification accuracies after 10 principal components. SVM-LNR acquires 92% with 10 principal components and 92.10% with 140 principal components. Likewise, K-NN acquires 90.51% with 10 principal components and 90.58% with 140 principal components.

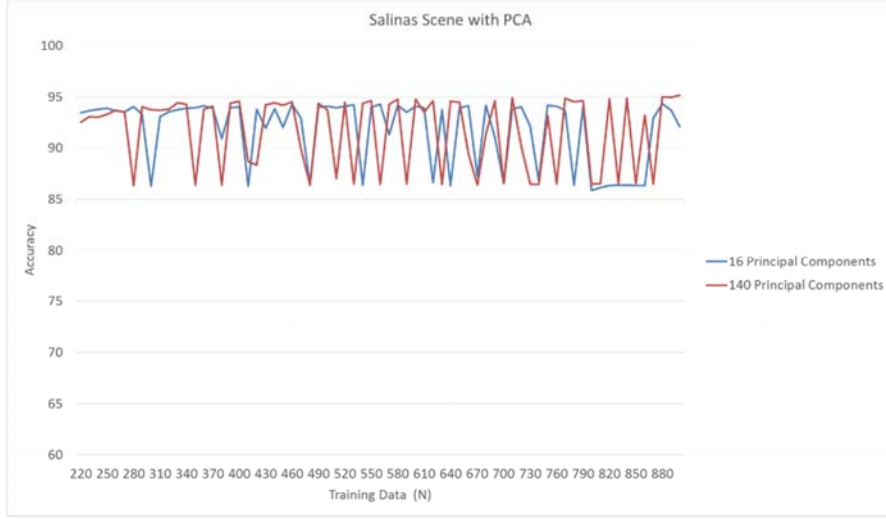


Figure 27 SVM-LNR Classification Results (Randomly Selected N Samples with PCA)

4.3.3 *Salinas Scene with Post-Processing*

Post-processing experiments aim to investigate and explain the impacts of spatial information usage to hyperspectral classification. We conducted the experiments as we stated in Section 3.5.1 for 3x3 window filtering and Section 3.5.2 for majority voting with meanshift segmentation. The results for filtering with 3x3 window results and majority voting with meanshift segmentation results can be seen in Figure 28 and Figure 29.

SSC and IPS have similar properties for classification by using post-processing as filtering with 3x3 window. This method improves the classification accuracies of the classification without post-processing for all algorithms. All the classification algorithms for SSC generally obtained better classification accuracies than IPS. Even for first N sample selection method the classification accuracies are around 75%. Similar to IPS experiments, filtering with 3x3 window increases the classification accuracy by 5-8%. When lower number of training data sizes are used, SVM-RBF and SVM-LNR algorithms obtain better classification accuracies than the other two algorithms. The accuracies are almost the same when higher training data sizes are used. The classification accuracies for 220 training data size are 93.95% for SVM-LNR, 92.60% for SVM-RBF, 91.50% for K-NN and 85.92% for ML algorithm. ML algorithm obtains the worst accuracy for lower training data sizes. However, if 900 training data used for classification, the accuracies are almost the same for SVM-LNR and SVM-RBF. The accuracies for 900 training data are 94.89% for SVM-LNR, 95.49% for SVM-RBF, 95.26% for K-NN and 94.84% for ML algorithm.

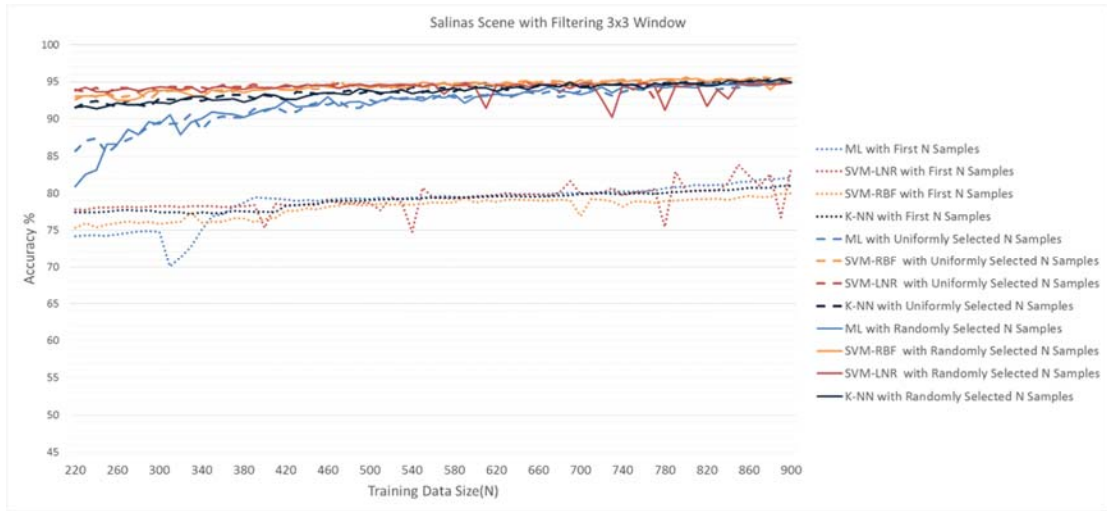


Figure 28 Salinas Scene Filtering with 3x3 Window

We obtained the best classification accuracies by using majority voting with meanshift segmentation for SSC. However, random training data selection mostly affects SVM-RBF algorithm. The main reason is that between two untrained crop fields, one dominates the other in majority voting via meanshift segmentation. Uniform training data selection ensures the classification accuracy above 90%. SVM-RBF obtained 99.29%, SVM-LNR obtained 99.33%, ML obtained 99.34% and K-NN obtained 99.38% classification accuracy.

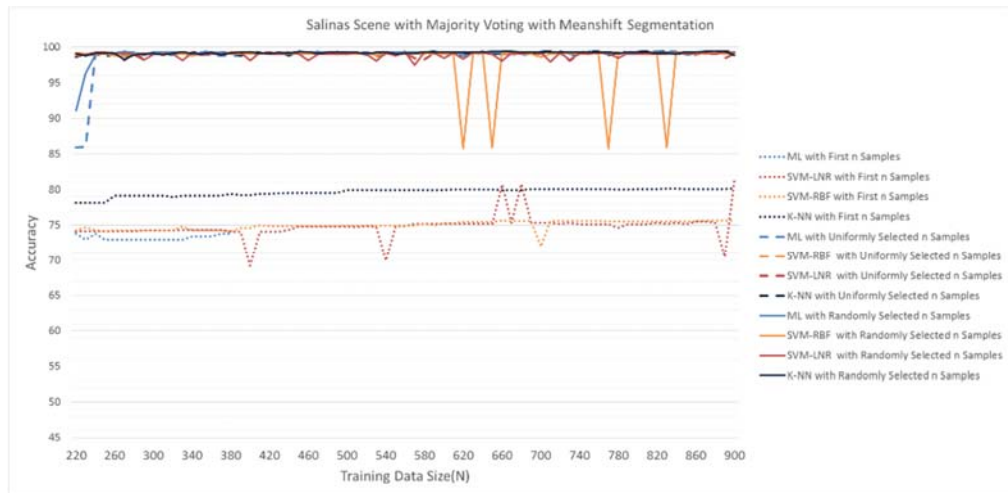


Figure 29 Salinas Scene Majority Voting with Meanshift Segmentation

4.3.4 Salinas Scene with Pre-Processing and Post-Processing

As we stated before, PCA usage reduces classification time for SSC. ML is the fastest algorithm that we used for classification. Normal classification time of ML with post-processing is 59.26 seconds, however it reduces to 18.56 seconds when joint pre-processing and post-processing is used. The classification times for all other algorithms also increased by 10-15 seconds when post-processing methods are used. However, the combination of post-processing and pre-processing usage improves classification time without reducing the classification accuracy. The classification times reduce for SVM-LNR from 293.14 to 114.98 seconds, for SVM-RBF from 419.20 to 194.65 seconds, for K-NN from 367.44 to 17.33 seconds.

Our experiments show that, in order to achieve higher accuracy rates with lower training data size, spatial information usage is necessary. For SSC experiments with post-processing, filtering with 3x3 window improves the classification accuracy by 3% to 5%, whereas majority voting with meanshift segmentation improves the classification accuracy by 5% to 20%.

In Figure 30 and Figure 31, the performance of classification algorithms can be seen. As is seen in the figures, pre-processing improves the classification accuracy, however they are still worse than the accuracies that we obtained by post-processing. The combination of pre-processing and post-processing methods improve both classification time and accuracy. Even if lower number of training data sizes used, classification accuracy that we obtained by using majority voting with meanshift segmentation is around 99%. With 12 principal components the classification accuracies with pre-processing and post-processing is very similar to the accuracies of classification with post-processing methods (Section 4.3.3).

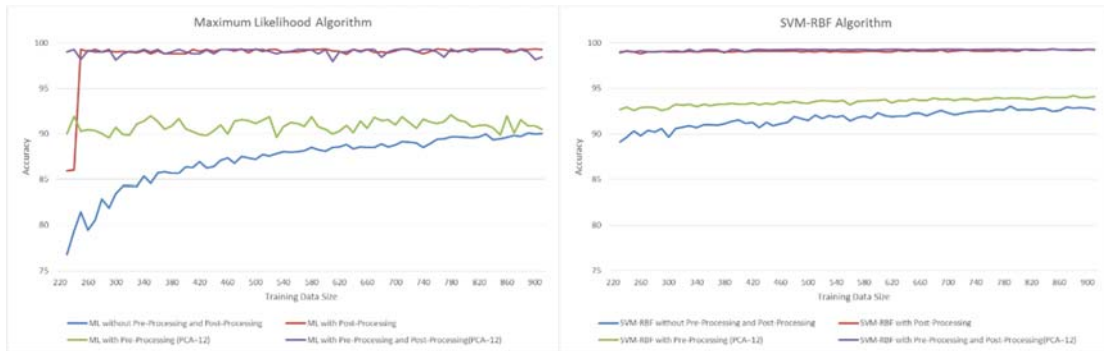


Figure 30 The Classification Accuracies for ML (left) and SVM-RBF (right)

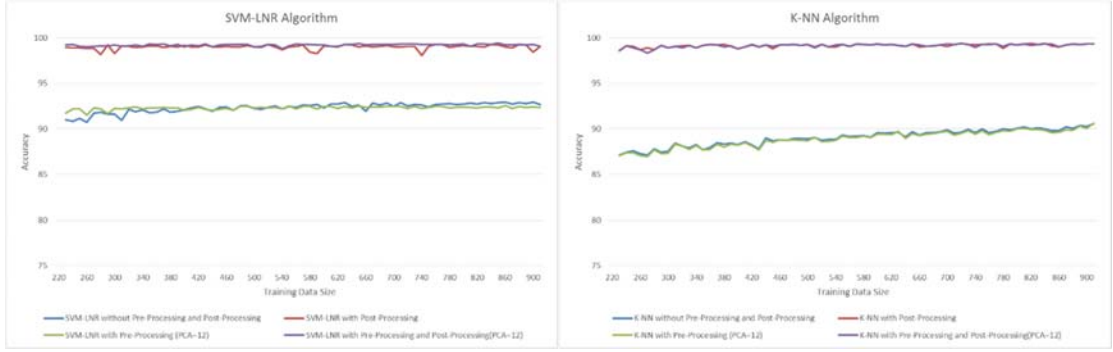


Figure 31 The Classification Accuracies for SVM-LNR (left) and K-NN (right)

Training data selection methods affect classification accuracy similar to previous experiments. First N sample selection method acquired lower results than uniformly selected N sample selection and randomly selected N sample selection methods. The accuracies that we obtain from random selection methods are also similar to the accuracies obtained from uniform selection methods. The highest accuracy that we obtained from first N sample with PCA and filtering with 3x3 window method is for ML 80.67% which is obtained with 103 principal components and 900 training data, for SVM-LNR 87.65%, for SVM-RBF 84.51 and for K-NN 81.53%. When we changed the post-processing method to meanshift segmentation most of the cases the classification accuracy is not improved for K-NN. However, SVM-LNR can reach the 96% accuracy with this selection method.

The improvements on the classification of SSC can be clearly seen in Figure 30 and Figure 31. For 12 principal components even with lowest training data sizes the accuracies are reaching 99%. Segmentation success is also an important factor for this method. Since IPS is a very small scene, majority voting after segmentation improves the accuracy up to a specific point. However, SSC is larger than IPS and majority voting with meanshift segmentation method can improve the accuracy to 99.5%. Although the classification accuracy with 10 principal components and 460 training data is 99.31%, the maximum accuracy for ML algorithm is 99.51% with 120 principal components and 280 training data. The classification accuracies for other algorithms are also similar. SVM-LNR achieves its maximum classification accuracy (99.54%) with 28 principal components and 610 training data. Average accuracy for more than 10 principal components is more than 99%. The classification accuracies of SVM-RBF and K-NN are very similar with SVM-LNR. For 12 principal components, on the average accuracy is 99.41% for SVM-RBF and 99.31% for K-NN.

4.4 Pavia University Scene Experiments

In literature, PUS is one of the most used scenes for hyperspectral classification. We conducted our experiments as we did in Section 4.2 and Section 4.3. The experiments of this chapter are aimed to investigate the differences of classification algorithms with different size of training data. Additionally, pre-processing and post processing usage with these algorithms is also investigated. The confusion matrixes for PUS are given in Appendix C.

4.4.1 Pavia University Scene without Pre-Processing and Post-Processing

As in other experiments in Section 4.2.1 for IPS and Section 4.3.1 for SSC, we conducted our experiments for PUS. The results of these experiments are analyzed for the effect of training data size to classification algorithms. This section specifically focuses on the training data size and training data selection type. We discussed our training data selection methods in Section 3.3.

Training data selection methods affect classification accuracy for all algorithms similarly with IPS and SSC. The lowest classification accuracies are acquired by the first N sample selection method for all algorithms. Besides SVM-RBF algorithm obtained better results than other classification algorithms with this method. Still the accuracies are not exceeding 71%. Figure 32 shows the classification accuracies for all algorithms with three different training data selection methods. As is seen in the figure, for uniformly selected N sample method obtained very similar accuracies with randomly selected N sample method.

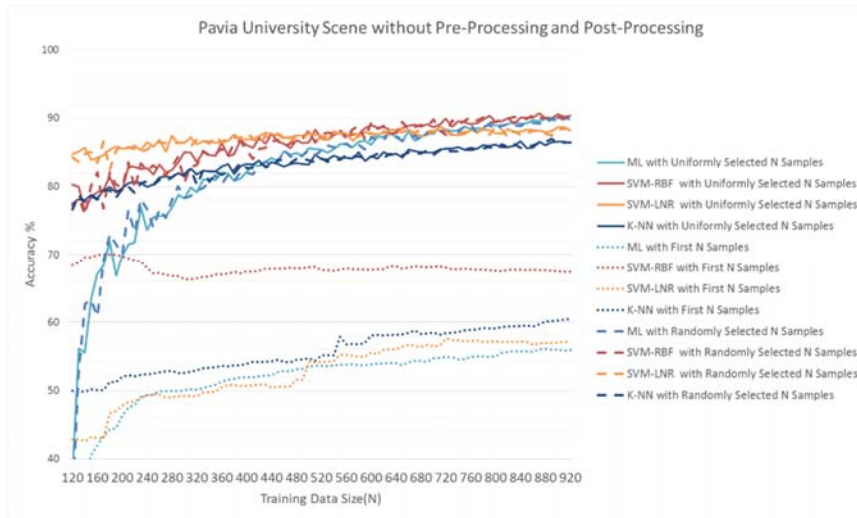


Figure 32 The Classification Results of Pavia University Scene without Pre-Processing and Post-Processing

SVM-LNR and K-NN algorithms are not much affected by training data size. However, the difference between the accuracies that procured with minimum and maximum training data size for ML is very high. On the other hand, SVM-RBF and ML procured the best accuracies with higher training data size. K-NN algorithm acquired lowest accuracies if more than 450 training data size used for learning. The best accuracies without using pre-processing or post-processing method are for ML algorithm 90.31%, for SVM-RBF algorithm 90.69%, for SVM-LNR algorithm 88.69% and for K-NN algorithm 86.66%. When we compared to the other two scenes, for PUS ML algorithm performs better. SVM-LNR and K-NN perform similar for all three scenes.

4.4.2 Pavia University Scene with Pre-Processing

The usage of pre-processing affects the classification time of PUS similar to SSC. SVM-RBF and ML algorithm both improve the accuracy and decrease the classification time. For ML algorithm classification time without pre-processing is 26.86 seconds, however the classification time with pre-processing reduces to 4.25 seconds. Other algorithms also improve both the classification time and accuracy. SVM-RBF reduces the classification time from 136.64 seconds to 112.35 seconds, SVM-LNR reduces the classification time from 113.66 seconds to 41.12 seconds and K-NN reduces the classification time from 171.27 seconds to 2.87 seconds.

In Figure 33, the improvements for ML and SVM-RBF with PCA are shown. The usage of more principal components does not give an advantage to any algorithm, although K-NN algorithm acquired almost the same accuracy for all training sizes for more than 11 principal components. On the average SVM-RBF and SVM-LNR decreased the classification accuracies if more than 45 principal components are used.

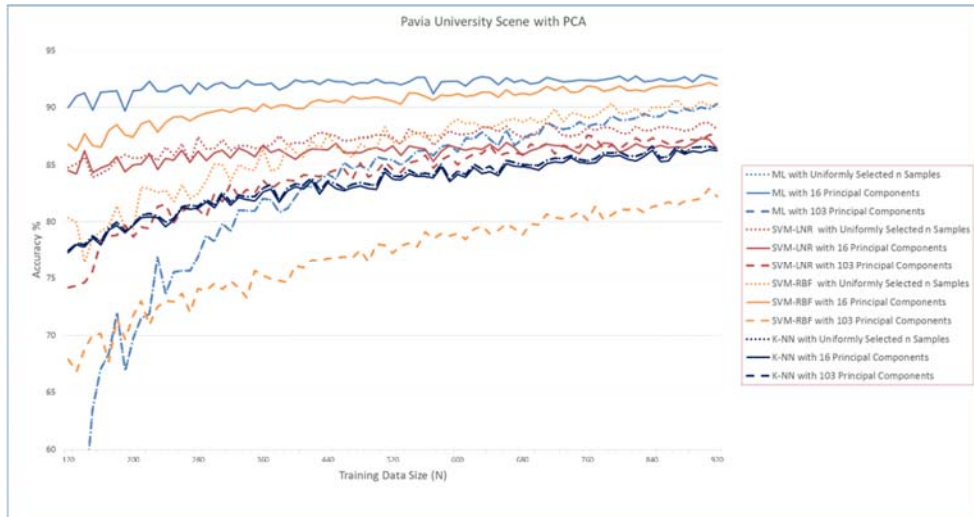


Figure 33 The Classification Accuracies of Pavia University Scene with Pre-Processing

Similar to the other two scenes, the usage of PCA improves the classification accuracy of the ML algorithms. By using 16 principal components, ML algorithm reached 92.05% on the average. Its maximum accuracy is 93.18% with 900 training data. The average classification accuracy for the ML algorithm without pre-processing is 82.18% and maximum accuracy is 90.31%. SVM-RBF algorithm also improved the classification accuracy by using 16 principal components. The average accuracy with 16 principal components is 90.35% and maximum accuracy is 92.77%. However, the average classification accuracy for SVM-RBF algorithm without pre-processing is 86.40% and maximum accuracy is 88.69%. Besides, K-NN algorithm does not improve its classification accuracy from pre-processing. Average and maximum accuracies are very similar for both cases. The average accuracy for K-NN algorithm with PCA is 83.47% while without using PCA is 83.19%. Maximum accuracy for K-NN algorithm with PCA is 86.65% whereas without using PCA is 86.34. While the usage of PCA is giving an advantage for classification of PUS, SVM-LNR decreases its classification accuracy almost 1% if pre-processing step is used. On the average its classification accuracy with PCA is 86.14% and maximum classification accuracy is 88.45%, whereas without using PCA average accuracy is 87.16% and maximum accuracy is 88.69%.

4.4.3 Pavia University Scene with Post-Processing

Similar with other two scenes, both post-processing methods that we mentioned in Section 3.5 improved the classification accuracy for all algorithms. The results for filtering with 3x3 windows can be seen in Figure 34 and the results for majority voting with meanshift segmentation results can be seen in Figure 35. The patterns for filtering with 3x3 windows method are very similar to patterns of the classification results without post-processing. The classification accuracies improved 3%-10% for all algorithms. Majority voting with meanshift segmentation also performed like other scenes. Best accuracies are obtained with the classification by using majority voting with meanshift segmentation as post-processing. Training data selection methods affected classification accuracy as it is in the without pre-processing and post-processing methods (Section 4.4.1). Randomly selected N samples method and uniformly selected N samples method achieved almost the same accuracy. Likewise, first N sample method performed worse than the other two methods.

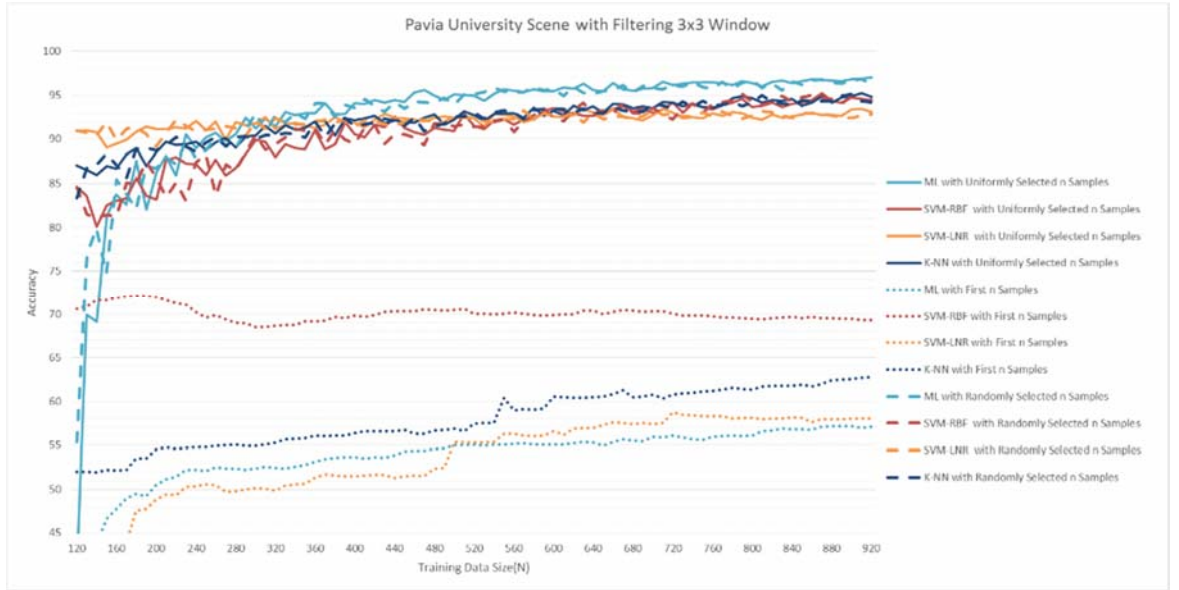


Figure 34 Pavia University Scene Results for Filtering with 3x3 Window

Filtering with 3x3 window method mostly improved the K-NN and ML algorithms. If more than 400 training data used, ML is achieving better accuracies than the other algorithms. The best accuracy that we obtained with ML algorithm is 97.04%. ML algorithm performs 92.48% on the average. On the other hand, SVM-LNR algorithm obtained at most 93.46% accuracy, however its average accuracy is 92.13. It is more robust than ML algorithm. SVM-RBF algorithm could reach 95.23% accuracy. Finally, K-NN algorithm obtained the maximum accuracy as 95.27%. Its average accuracy is 92.08%.

Majority voting with meanshift segmentation method also improves the classification accuracy as it is in other two scenes. In Figure 35, the classification results for all training data selection methods can be seen. As is seen in the figure, K-NN and ML algorithms perform better than other two algorithms for more than 360 training data. Their classification accuracies are almost 99% for K-NN and ML. SVM-LNR and SVM-RBF also perform very similarly. First N sample selection method's performance is very similar to IPS and SSC. Randomly selected N sample method and uniformly selected N sample methods obtained similar accuracies for the same training data. Selection method mostly changed SVM-LNR algorithm's accuracy.

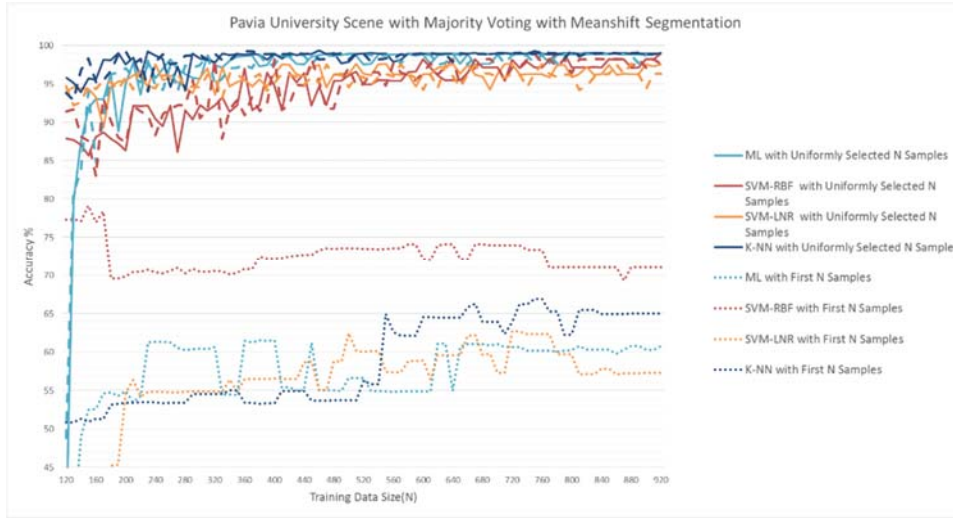


Figure 35 Pavia University Scene Results for Majority Voting with Meanshift Segmentation

Average and maximum accuracies are also increased when majority voting with meanshift segmentation method are used. For the SVM-LNR algorithm, highest accuracy achieved with majority voting with meanshift segmentation is 97.64% with 640 training samples. Average accuracy of SVM-LNR algorithm is 96.02%. SVM-RBF achieved 98.97% with 900 training data, however its average accuracy (94.42%) is lower than SVM-LNR. ML algorithm performs better than SVM-LNR and SVM-RBF algorithms. It achieves 98.99% accuracy with 750 training data. On the average ML algorithm achieves 96.74%. The average classification rate increases to 98.55% when 360 or more training data used. The best classification accuracies with this method are achieved with K-NN algorithm for PUS. K-NN algorithm achieves 99.34% with 460 training data and average accuracy for K-NN algorithm is 98.39%.

4.4.4 Pavia University Scene with Pre-Processing and Post-Processing

We mentioned about the classification time aspect of pre-processing in Section 4.4.2. Post-processing usage is slightly affecting the classification time of PCA. Post-processing step requires 10 (filtering with 3x3 window) and 15 (majority voting with meanshift segmentation) seconds to process. The classification process with filtering with 3x3 window takes 38.5 seconds. If PCA is used for filtering with 3x3 window the classification time reduces to 17.09 seconds. Required time for PCA and majority voting with meanshift segmentation is also reducing from 40.48 seconds to 18.4 seconds. The classification time for SVM-RBF reduces from 146.51 seconds to 123.56 seconds for filtering with 3x3 window and 147.38 seconds to 127.78 seconds for majority voting with meanshift segmentation, for SVM-LNR reduces from 125.53 seconds to 54.36 seconds for filtering with 3x3 window and 115.40 seconds to 55.80 seconds for majority voting with meanshift segmentation. Similar with the previous experiments, K-NN algorithm improves its classification time mostly with PCA. Its classification time reduces from 184.33 seconds to 15.88 seconds for filtering with 3x3 window method and 184.53 to 16.95 seconds for majority voting with meanshift segmentation.

Classification with PCA and filtering with 3x3 window method improves the classification accuracy similar to the other scenes. We mentioned about PCA reducing the duration of the classification process. The aim is getting higher classification rates with lower number of principal components. Figure 36 shows that with 16 principal components and filtering with 3x3 window method improves the classification accuracies of the classification without pre-processing and post-processing methods for all training data sizes. For SVM-RBF and ML algorithms, this method also performs better than classification with only post-processing method. The classification accuracy of PCA and without PCA methods are very similar for K-NN algorithm.

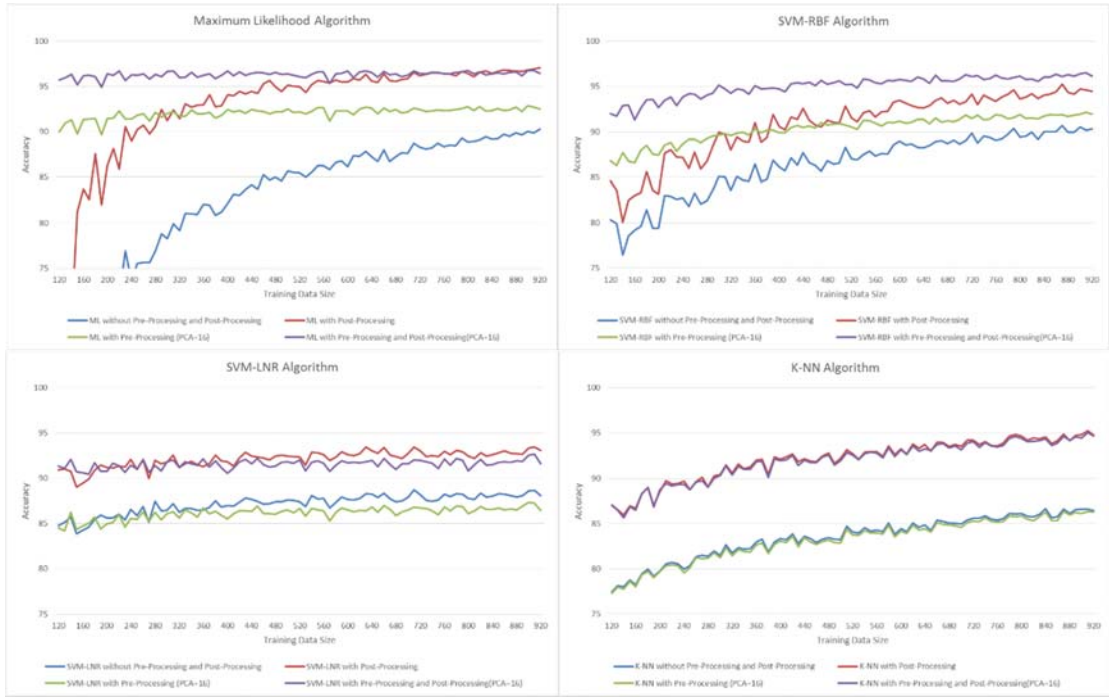


Figure 36 Pavia University Scene Classification Results for Filtering with 3x3 Window Method ML (top-left), SVM-RBF(top-right) SVM-LNR (bottom-left) and K-NN (bottom-right)

For filtering with 3x3 method, ML algorithm obtained best classification accuracy with ML with post-processing method with 97.04%. Although ML with pre-processing and post-processing method achieves 96.77%, its average accuracy (96.30%) is better than ML with post-processing method (92.48%). In Figure 36, for lower training data sizes the difference between the two methods can be seen clearly. SVM-RBF also performs better with pre-processing and post-processing method. Both its average accuracy (95.03%) and maximum accuracy (96.50%) are better than SVM-RBF with post-processing method (Average 90.81% and Maximum 95.23%). K-NN algorithm performs almost same for both methods. The average accuracies for K-NN with post-processing method are 92.08% and for K-NN with pre-processing and post-processing is 91.90%. The maximum classification accuracies are also similar. The maximum classification accuracy for K-NN with post-processing method is 95.27% and for K-NN with pre-processing and post-processing method is 95.06%.

Majority voting with meanshift segmentation results can be seen in Figure 37. When the classification process carried out without post-processing, ML and SVM-RBF performs better with pre-processing usage. On the other hand, K-NN acquires almost the same accuracies with or without using PCA. All algorithms obtained their best results with joint pre-processing and post-processing method. However, all of them obtained their best results with different number of principal components. For example, ML algorithm achieved its best result with 460 training data and 17 principal components, SVM-LNR achieves its best results with 500 training data and 75 principal components. SVM-RBF algorithm obtained its best accuracy (99.21%) with 260 training data and 27 principal components. Lastly, K-NN algorithm obtained the best results of these experiments with 99.41% with 670 training data and 13 principal components.

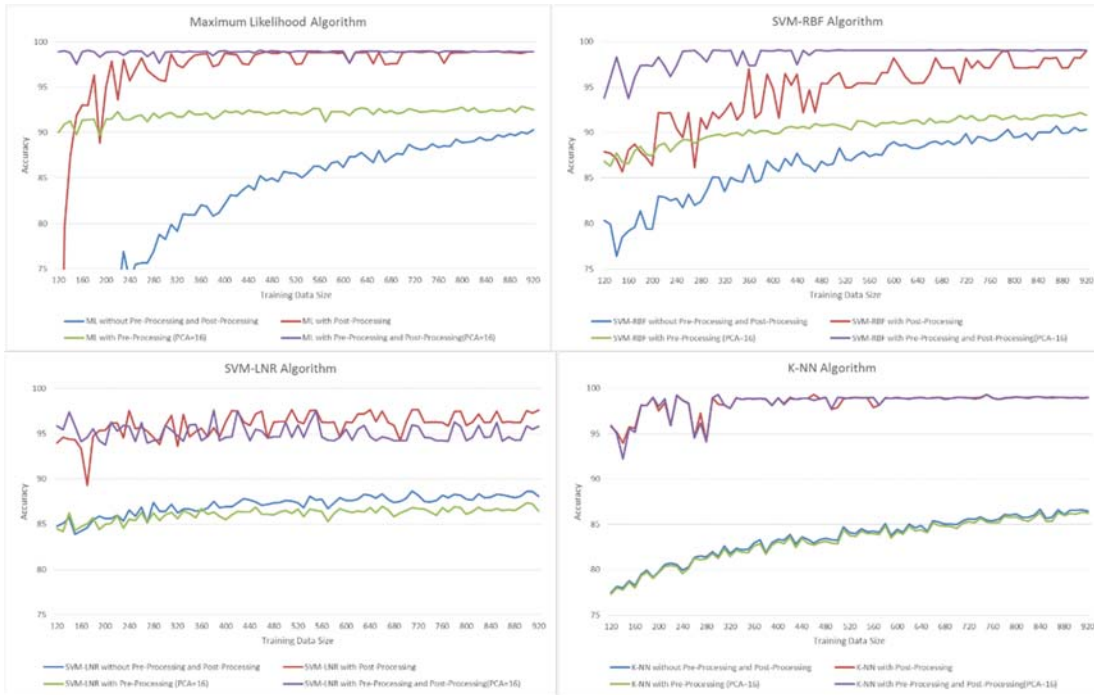


Figure 37 Pavia University Scene Classification Results for Filtering with 3x3 Window Method ML (top-left), SVM-RBF(top-right) SVM-LNR (bottom-left) and K-NN (bottom-right)

4.5 The Effects of Segregation of Training and Testing Data

In order to compare our classification results with those in the literature we extract training data from the image for training but tested on all available data. So, the same data is used both for training and testing. In this section, for SSC first we extract training data from the image. Then, we segregate the training data from testing data. Then we carried on the classification process. We observed IPS and PUS and obtained similar results with SSC. Sample groundtruth for SSC can be seen in Figure 38.

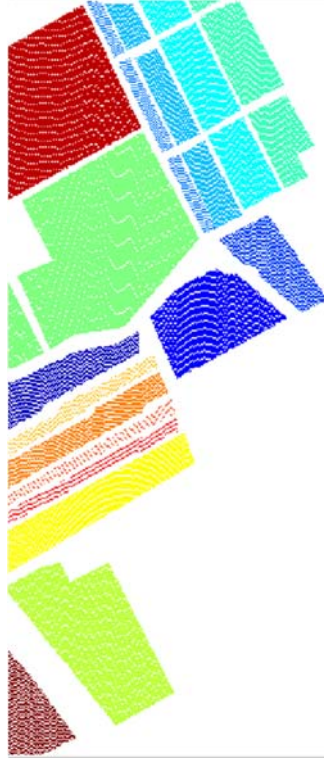


Figure 38 Sample Groundtruth for SSC (N=700)

Figure 39 shows the classification results with and without segregation training data from testing data. It is clearly seen that, the difference between two results are lower for lower number of training data and it gets higher when we increase the training data size for all classification algorithms. On the average the difference between two methods is 2.44% for ML, 1.32% for SVM-RBF, 1.13% for SVM-LNR and 2.61% for K-NN algorithm. However, the maximum classification accuracy reduces from 90.05% to 86.77% for ML algorithm, 93.02% to 91.37% for SVM-RBF algorithm, 92.96% to 91.70% for SVM-LNR algorithm and 90.59% to 87.42% for K-NN algorithm.

For all algorithms, classification accuracies do not change much after 500 training data. Especially for ML and K-NN algorithms, while the classification accuracy increase after 500 training data, when we segregate training data from testing data the accuracies do not change much.

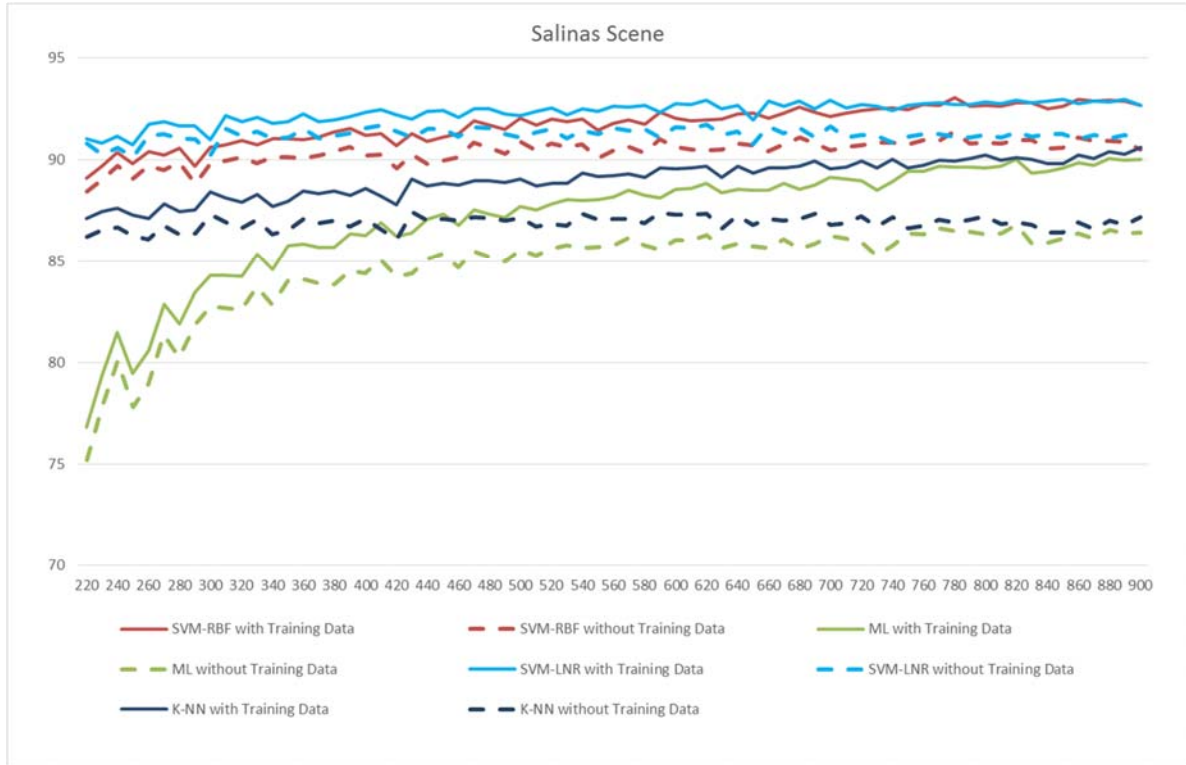


Figure 39 Classification Results for SSC with and without Segregation of Training Data

We used majority voting with meanshift segmentation method in order to use spatial information. Even though the classification accuracies for all training data sizes and algorithms reduce with segregation of train and test data, the classification accuracies are not affected when spatial information are used. After pixelwise classification, for every segment we perform majority voting to assign class labels to all pixels in that segment. The pixels that are used for training data are not counted in this process. Figure 40 shows the results for SSC with and without segregation of training data by using majority voting with meanshift segmentation method. The average and maximum accuracies for both methods are very similar. The difference for all training data sizes is not more than 0.5%. Since we need 8 neighborhood for filtering with 3x3 window method, we could not apply it to segregated groundtruth map.

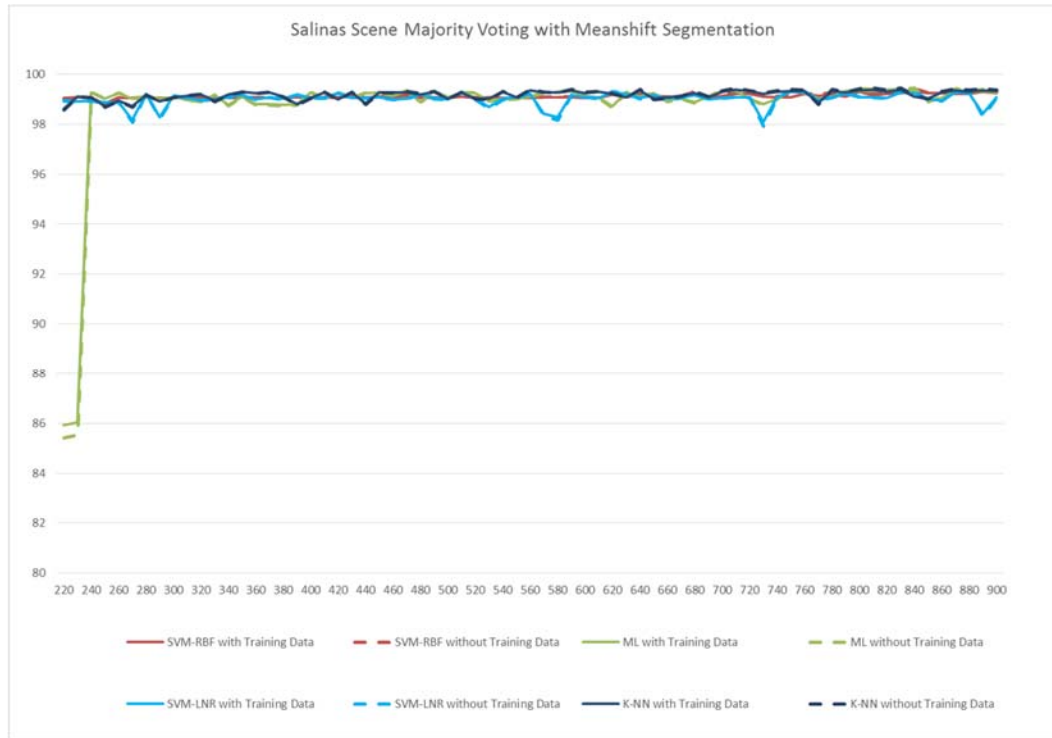


Figure 40 Classification Results for SSC with and without Segregation of Training Data by using Majority Voting with Meanshift Segmentation Method

CHAPTER 5

CONCLUSION

5.1 Summary

In this study, variations on hyperspectral image classification algorithms are analyzed. Indian Pines, Salinas and Pavia University scenes are used for classification experiments. Indian Pines and Salinas Scenes were acquired by AVIRIS sensor. Pavia University Scene was acquired by ROSIS sensor. For all scenes the bands which have no information or water absorption bands were removed. We employed ML, SVM and K-NN as supervised classification algorithms. We also implemented two different kernels for the SVM algorithm: A linear kernel and an RBF kernel. In order to show the effect of training data on hyperspectral classification, we employed three different training data selection methods: First N samples of all classes, randomly selecting N samples from groundtruth classes and uniformly selecting training data from groundtruth classes. We utilized different training data sizes for all training data selection methods. These training data sizes differ for three scenes by their band size. We also investigated the contribution of pre-processing with PCA and post-processing with spatial information from filtering with 3x3 window and majority voting with meanshift segmentation for each of the algorithms listed above.

In summary, we found out that the training data selection method and training data size are very important for hyperspectral classification. Especially ML is affected by the change of the training data size for all three scenes. PCA reduces the classification time for SSC and PUS. It also improves the classification accuracy for ML and SVM. PCA does not improve both the classification accuracy and classification time for IPS. Spatial information usage improves the classification accuracy for all IPS, SSC and PUS. For IPS, filtering with 3x3 window method improves the classification performance 5-10% for SVM-RBF, 7-11% for SVM-LNR, 10-13% for K-NN and 9-18% for ML. This method showed more improvement for smaller training data sizes. Majority voting with meanshift segmentation method obtained best results for all algorithms and scenes. All algorithms improved their classification accuracies by 5-15%. Pre-processing with PCA and post-processing usage do not affect the classification accuracies for IPS.

Almost all algorithms procured similar results for without pre-processing and post-processing method. ML algorithm increases classification accuracy more than SVM-LNR, SVM-RBF and K-NN algorithms as the training data size increases. PCA improves the classification accuracy and reduces classification time for PUS and SSC. For SSC, filtering with 3x3 window method improves the classification performance 2-5% for SVM-RBF, 2-4% for SVM-LNR, 4-5% for K-NN and 4-9% for ML. Pre-processing with PCA and post-processing usage jointly improve the classification accuracies for SSC and PUS. With this method, K-NN improves classification accuracy more than other algorithms for SSC and PUS. With lower numbers of principal components this method also lowers the classification time. Segregation of training data from

test data reduces classification accuracy unless majority voting with meanshift segmentation is employed for post-processing.

Without using pre-processing and post-processing, all algorithms confuse spectrally similar materials in the Pavia University Scene (PUS). We can separate this confusion into two groups. The first group consists of asphalt, gravel, bitumen, self-blocking bricks and the second group consists of bare soil, meadow and tree. Pre-processing (PCA) method usage does not change the classification performance for all materials. The confusion within the elements in the groups still exist. When post-processing (filtering with 3x3 window) method is used for classification all algorithms reduce confusion between asphalt-bitumen-self-blocking bricks and bare soil-meadow materials. However, SVM-LNR and K-NN do not improve as much as SVM-RBF and ML. When post-processing (majority voting with meanshift segmentation) method is used for classification all algorithms reduce confusion for both groups. With this method ML, K-NN and SVM-RBF mainly confuse meadow-tree and SVM-LNR confuses meadow-bare soil materials. The confusion highly depends on the meanshift segmentation performance. When pre-processing (PCA) and post-processing (filtering with 3x3 window) method is used for classification ML confuses asphalt-self-blocking bricks and meadow-tree-bare soil materials. This method especially reduces the confusion between gravel-self-blocking bricks and asphalt-gravel for ML algorithm. SVM-RBF algorithm improves the dissociation between tree-meadow with this method. SVM-LNR and K-NN performances are not improved. When pre-processing (PCA) and post-processing (majority voting with meanshift segmentation) method is used for classification ML and K-NN perform similar to post-processing (majority voting with meanshift segmentation) method. SVM-RBF confuses asphalt-tree materials. Again this confusion is caused by the meanshift segmentation performance.

In the Salinas Scene (SSC) all algorithms confuse two untrained fields (vinyards and grapes) with each other. These classes are both grapes produced for different purposes. Hence they are very much alike spectrally. They can mainly be separated with the help of spatial information. As post-processing (majority voting with meanshift segmentation) segments the two areas (grapes and vinyards) successfully, it becomes possible to identify the classes correctly using majority voting. The other pair of classes that are confused are grapes and corn. This observation is more prominent with K-NN and SVM-RBF. This might be due to the presence of mixed pixels in the groundtruth. Such mixed pixels tend to be used for classification of pixels that are spectrally close. K-NN and SVM-RBF are likely to be affected more by this phenomenon. Grapes and corn pair is also mixed when post-processing (majority voting with meanshift segmentation) method is used. We observed that SSC has many unclassified pixels in the ground truth. These pixels are usually roads next to agricultural areas. These road areas are also segmented by meanshift segmentation however, they are not labeled in the groundtruth. The classification algorithms usually classify these roads as corn fields. When these segments overlap with agricultural areas, as a result of majority voting some pixels in the agricultural areas are also labeled as corn. When we used pre-processing (PCA) and post-processing (majority voting with meanshift segmentation) these classification errors are reduced for all algorithms.

For the Indian Pines Scene (IPS) all algorithms confuse Corn (notill-mintill)-Soybean (notill-mintill) classes with each other without spatial information. However, SVM-RBF distinguishes Soybean-notill and Soybean-mintill, Soybean-notill-Corn (notill-mintill) classes better than other algorithms. The main reason for this confusion might be due to the presence of mixed pixels in the groundtruth and the lower resolution of IPS scene. Pre-processing (PCA) method usage does

not change the classification performance for all algorithms. When we use spatial information with post-processing (filtering with 3x3 window) the confusion between Corn (notill-mintill) - Soybean(notill-mintill) is distinctly reduced for SVM-RBF and K-NN. ML still confuses Soybean-mintill with Soybean-notill and Corn-notill classes. K-NN only confuses Soybean (Notill-mintill) classes with each other. As it is in the other scenes with all algorithms post-processing (majority voting with meanshift segmentation) method performs better than the other methods. All algorithms confuse Soybean-mintill with Corn-notill classes more than other classes. The main reason for that confusion is that some part of Soybean-mintill class is segmented by meanshift segmentation as Corn-notill. As a result of majority voting these regions are classified as Corn-notill. This confusion is also valid for pre-processing (PCA) and post-processing (majority voting with meanshift segmentation) methods.

5.2 Comparison with the Literature and Discussion

Training data selection methods gave similar results for all scenes. First N sample selection method obtained lowest accuracies for all training data sizes because the training data samples cannot represent the whole scene. Uniform N sample selection method and randomly N sample selection method obtained similar accuracies. ML algorithm is affected most by training data size for all scenes. It needs larger train data for satisfactory performance. On the other hand, SVM-RBF, SVM-LNR and K-NN algorithms are slightly affected by training data sizes. Without pre-processing and post-processing, the performances of the algorithms are very similar.

We compare the results with literature on hyperspectral classification with IPS. Since there is no pre-determined training data set for all maps, all of them used different training data set for classification. We acquired 95.38% by using spatial information as post-processing by using almost 30% of data. [58] used Gaussian maximum likelihood classifier and leave-one-out covariance estimation method for hyperspectral classification. They selected 20% of total samples as training samples for each class. They acquired 89.1% classification rate. [59], also used SVM and composite kernels for hyperspectral classification. They used 20% of the samples as training set. They obtained 96.53% classification accuracy with spectral and contextual kernels. [60] obtained 96.5% overall accuracy for IPS by using spatial information based SVM. They used randomly selected %10 of the samples as training samples. [61] used linear discriminant analysis to for dimensionality reduction. They also employed Markov Random Fields concept to incorporate the spatial information of the image. They also investigated the effect of training data size on accuracy by using three different training subsets and test it on randomly selected 100 samples per class. They acquired 90.78% classification rate with the best training subset.

For PUS, we obtained 99.41% by using PCA and majority voting with meanshift segmentation method and K-NN algorithm with almost 14% of data. [62] used PUS to compare the SVM-LNR, SVM-RBF, K-NN and RBF classifiers' performance. They do not use any pre-processing or post-processing methods. Their training data is not specified but they used 4757 training samples and 4588 testing samples from the image. Overall accuracy is given as 93.42% with SVM-RBF. The duration of total classification time is 2702 seconds. [63] used both supervised and unsupervised learning methods for hyperspectral classification. They used SVM for classification and fuzzy-c-means for providing segmentation maps. In order to the employ

segmentation output, they used weighted majority voting rule. They repeat the classification process five times with randomly selected 50 training samples for IPS. They performed the same experiments for PUS. They obtained 91.05% accuracy for IPS and 95.99% accuracy for PUS. [64] also used majority voting over segmented image. After pixelwise classification, they employed majority voting over three different segmentation methods (Watershed, Gaussian mixture resolving and hierarchical segmentation). The results of these three majority voting process are used to assign markers to a pixel. Final part of their classification process is to group these pixels into a minimum spanning forest (MSF) and they procured spectral-spatial classification map from MSF. They obtained 97.90% overall accuracy for PUS. They used SVM-RBF for pixelwise classification with 3921 pixels for training data but there is not detailed information about the specific numbers for classes. The overall accuracy for IPS with the same method is 92.32%. [65] also used spatial information to improve pixelwise classification of hyperspectral images. They used morphological operators following pixelwise classification to incorporate spatial information. They specify different training sample size for classes and used 718 training samples. They obtained 95.57% classification accuracy.

[65] has used a model with pre-processing and post-processing methods. As a pre-processing method they used PCA. They also used morphological profiles to improve classification accuracy. They randomly selected 2% of the pixels from all classes. They perform experiments on morphological operators' number effect on classification accuracy. They acquired 95.03% accuracy for SSC. [66] used PCA for dimensionality reduction. They used synergetic theory which is founded by [67]. They classify SSC and obtained 90% average accuracy for the whole scene.

We obtained the best results by using majority voting with meanshift segmentation method. The performances of our methods mainly depend on the performance of meanshift segmentation. Except SVM-LNR, all three algorithms perform above 99% classification accuracies. Specifically on PUS, K-NN performs 99.33% classification accuracy with 460 training sample for each class. The improvement of K-NN algorithm is 16.40%. It acquired 99.22% with 220 training samples and 99.01% with 190 training samples from each class. On the other hand SVM-RBF and ML algorithms perform better with PCA. When we used 16 principal components, we obtained 99% classification accuracy by using SVM-RBF with 260 training samples from each class and 99.05% classification accuracy by using ML with 170 training samples from each class. We also observed that K-NN algorithm is not affected by the increase of the number of principal components after $N=15$. With lower number of principal components K-NN algorithm performs better than the other three algorithms.

Segregation of training data from test data experiments are also conducted in this study. It is observed that the average classification accuracies decrease by 4-5% when training data is segregated from test data. ML and K-NN algorithms are mostly affected by segregation process. We have not provided detailed tables with segregated data because it is not the practice within the remote sensing community. Spatial information usage with segregation of training data from test data experiments showed that the average classification accuracies are not changed with majority voting with meanshift segmentation method. For all training data sizes, the classification accuracies are almost the same with or without segregating training data.

5.3 Future Work

We used supervised classification algorithms for three remote sensing images. Although the obtained results are very promising, experiments with other images should also be conducted to generalize these results. We also showed that the performance of supervised classification algorithms mainly depend on training data. The minimum training sample set consisted of 120 samples for each class. This number might be considered excessively high for some applications. One possible future work is to improve classification performance with even lower training data sizes.

We show that spatial information usage by filtering with 3x3 window and majority voting with meanshift segmentation improves classification accuracy. Further experiments with other segmentation methods should also be conducted.

Hyperspectral images have high spectral resolution and low spatial resolution whereas multispectral images may have high spatial resolution but low spectral resolution. In order to improve hyperspectral classification accuracy, fusion of a high resolution multispectral image and a hyperspectral image may be considered.

We also noted that the imperfections in the groundtruth has negative effects on the classification accuracy. To combat the degrading effect of these imperfections of the groundtruth, the samples that are typical of the given class may be employed for training. Another approach could be to assign weights on training samples depending on their impurity. These issues will be investigated in future studies.

REFERENCES

- [1] Jensen, J. R. (2007). Remote Sensing of the Environment. Pearson Prentice Hall.
- [2] Lefsky, M. A., Cohen, W. B., Parker, G. G., & Harding, D. J. (2002). "Lidar Remote Sensing for Ecosystem Studies", *BioScience*, 52(1), 19-30.
- [3] Eric M. and Vincent P. (2005). "A Superspectral Micro Satellite System For GMES Land Cover Applications". *56th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*.
- [4] Heidner III, R. F., Straus, J. M. (2005). "The Future of US Commercial Remote Sensing from Space", *Global Priorities in Land Remote Sensing*
- [5] Andrew D. Meigs, John O., Tatyana Y.C. (2008, Oct). "Ultraspectral Imaging: A New Contribution To Global Virtual Presence" *Aerospace and Electronic Systems Magazine*, pp. 11-18.
- [6] Smith, R. B. (2012). "Introduction to Hyperspectral Imaging". Retrieved from [www.microimages.com: http://www.microimages.com/documentation/Tutorials/hyprspec.pdf](http://www.microimages.com/documentation/Tutorials/hyprspec.pdf)
- [7] Keinosuke F., Raymond R.H., (1989), "Effects of sample size in classifier design", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11873885
- [8] G. Hughes, (1968), "On the mean accuracy of statistical pattern recognizers", *IEEE Transactions on Information Theory*
- [9] Tadjudin, S., & Landgrebe, D. (1998). "Classification of high dimensional data with limited training samples"
- [10] Masaeli, M., Fung, G., Dy, J. G. (2010). "From transformation-based dimensionality reduction to feature selection". *International Conference on Machine Learning*.
- [11] Hall M. A. and Smith L. A. (1997). "Feature subset selection: A correlation-based filter approach". *Proc. Int. Conf. Neural Inf. Process. Intelligence Information System* (pp. 855-858).
- [12] Peng H., Long F., and Ding C. (2005). "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and minredundancy." *IEEE Trans. Pattern Anal. Mach. Intel.* (pp. 1226-1238).

- [13] Pai-Hui H., Yi-Hsing T. and Peng G. (2002, June). "Dimension Reduction of Hyperspectral Images for Classification Applications". *Geographic Information Sciences*.
- [14] Anil K. J., Robert P.W. D., Jianchang M. (2000). "Statistical Pattern Recognition: A Review.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- [15] B. Schölkopf, A. J. Smola, K. R. Muller. (1998). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", *Neural Computation*.
- [16] Craig R., Jie S. (2002). "Principal Component Analysis for Hyperspectral Image Classification", *Surveying and Land Information Science*, (pp. 115-122).
- [17] Duda, R. O. And hart, P.E. (1973), "Pattern classification and scene analysis", *Wiley, New York*
- [18] Gualtieri J. A. and Chettri S. (2000), "Support vector machines for classification of hyperspectral data", *Proc. of IEEE-IGARSS*, Hawaii, pp. 813-815.
- [19] Melgani F. And Bruzzone L., (2002) "Support vector machines for classification of hyperspectral remote sensing images," in *IGARSS*
- [20] Lennon M., Mercier G., and Hubert-Moy L. , (2002)"Classification of hyperspectral images with nonlinear filtering and support vector machines," , *IGARSS*
- [21] Burges C. J., (1998), "A tutorial on support vector machines for pattern recognition," in *Data mining and knowledge discovery*, U. Fayyad, Ed. *Kluwer Academic*, , pp. 1–43.
- [22] Mountrakis, G., Im, J., & Ogole, C. (2011). "Support vector machines in remote sensing: A review." *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3), 247-259.
- [23] Watanachaturaporn, P., Arora, M. K., & Varshney, P. K. (2005). "Hyperspectral image classification using support vector machines: A comparison with decision tree and neural network classifiers." In *American Society for Photogrammetry & Remote Sensing (ASPRS)*, Reno, NV.
- [24] Keinosuke F., (1990) "Introduction to Statistical Pattern Recognition." *San Diego, California, Academic Press, Inc.*,
- [25] Cover, T., & Hart, P. (1967). "Nearest neighbor pattern classification. Information Theory", *IEEE Transactions on*, 13(1), 21-27.
- [26] Atiya A. F., (2005) "Estimating the posterior probabilities using the k-nearest neighbor rule", *Neural Computation* vol. 17, no. 3, pp. 731–740

- [27] Chang, C. I. (2003). "Hyperspectral imaging: techniques for spectral detection and classification", *Springer*.
- [28] Jia, X., & Richards, J. A. (1994). "Efficient maximum likelihood classification for imaging spectrometer data sets", *Geoscience and Remote Sensing, IEEE Transactions on*, 32(2), 274-281.
- [29] Paalanen, P. (2004). "Bayesian classification using Gaussian mixture model and EM estimation: Implementations and comparisons", *Information Technology Project*.
- [30] Cortes C. and Vapnik V. (1995), "Support vector networks." *Machine Learning*, 20:273–297
- [31] Yao, Y., Frasconi, P., & Pontil, M. (2001, January). "Fingerprint classification with combinations of support vector machines. In Audio-and Video-Based Biometric Person Authentication", *Springer Berlin Heidelberg* (pp. 253-258).
- [32] Smola A.J. and Schölkopf B. (1998), "On a kernel-based method for pattern recognition, regression, approximation and operator inversion". *Algorithmica*, 22 (pp. 211 – 231)
- [33] Schölkopf, B., Burges, C., & Vapnik, V. (1995, August). "Extracting support data for a given task.", *In Proceedings, First International Conference on Knowledge Discovery & Data Mining. AAAI Press, Menlo Park, CA* (pp. 252-257).
- [34] Bryant P. and Williamson J.A. (1978). "Asymptotic behaviour of classification maximum likelihood estimates." *Biometrika*, (pp. 273-281)
- [35] Marriott, F. (1975), "Separating mixtures of normal distributions.", *Biometrics*. (pp. 553-563).
- [36] Day, N. (1969). "Estimating the components of a mixture of normal distributions", *Biometrika*. (pp. 463-474).
- [37] Scott A.J. and Michael J.S. (1971), "Clustering methods based on likelihood ratio criteria" *Biometrics*. (pp. 387-397).
- [38] Strahler, A. H. (1980). "The Use of Prior Probabilities in Maximum Likelihood Classification of Remotely Sensed Data", *Remote Sensing Environment*, (pp. 135-163).
- [39] Xiuping J. and John A. R. (1994). "Efficient Maximum Likelihood Classification for Imaging Spectrometer Data Sets", *IEEE Transactions on Geoscience And Remote Sensing*, (pp. 274-281).

- [40] Mader S., Vohland M. , Jarmer T. and Casper M. (2006). "Crop Classification With Hyperspectral Data Of The Hymap Sensor Using Different Feature Extraction Techniques", *EARSeL SIG on Land Use and Land Cover*, (pp. 96-101)
- [41] Fix E. and Hodges J.L. (1951). "Discriminatory analysis: Non-parametric discrimination: Consistency properties", *USAF School of Aviation Medicine*.
- [42] Smola A.J. and B. Schölkopf. (1998) "On a kernel-based method for pattern recognition, regression, approximation and operator inversion", *Algorithmica*, 22: (pp. 211 – 231)
- [43] Schölkopf, B., Burges, C., & Vapnik, V. (1996). "Incorporating invariances in support vector learning machines", *In Artificial Neural Networks—ICANN* (pp. 47-52).
- [44] LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J. S. and Vapnik, V. (1995, October). "Comparison of learning algorithms for handwritten digit recognition", *In International conference on artificial neural networks* (Vol. 60).
- [45] Blanz, V., Schölkopf, B., Bühlhoff, H. C. B. V. V., Burges, C., Vapnik, V., & Vetter, T. (1996). "Comparison of view-based object recognition algorithms using realistic 3D models", *In Artificial Neural Networks—ICANN* (pp. 251-256)
- [46] Roobaert, D., & Van Hulle, M. M. (1999, August). "View-based 3d object recognition with support vector machines." *In Neural Networks for Signal Processing IX, Proceedings of the 1999 IEEE Signal Processing Society Workshop*. (pp. 77-84)
- [47] Vapnik, V. (1979), "Estimation of Dependences Based on Empirical Data [in Russian]. Nauka, Moscow, (English)
- [48] Keerthi, S. S., & Lin, C. J. (2003). "Asymptotic behaviors of support vector machines with Gaussian kernel", *Neural computation*
- [49] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). "A practical guide to support vector classification"
- [50] Yuliya T. Jón Atli B. (2009). "Spectral–Spatial Classification of Hyperspectral Imagery Based on Partitional Clustering Techniques", *IEEE Transactions On Geoscience And Remote Sensing*, (pp. 2973-2987)
- [51] Fauvel, M. (2007). "Spectral and spatial methods for the classification of urban remote sensing data". *Ph.D. dissertation", Grenoble Inst. Technol., Grenoble*.

- [52] Fauvel, M., Chanussot J. , Benediktsson J. A. , and Sveinsson J. R. (2007). "Spectral And Spatial Classification Of Hyperspectral Data Using SVMs And Morphological Profiles", *IEEE Transactions Geoscience on Remote Sensing*, (pp. 3804-3814)
- [53] Jonathon S. (2005). "A tutorial on principal component analysis", *Systems Neurobiology Laboratory, Salk Insitute for Biological Studies*.
- [54] Hooke, R.; Jeeves, T.A. (1961). "'Direct search" solution of numerical and statistical problems". *Journal of the Association for Computing Machinery (ACM)* 8 (pp. 212–229)
- [55] Audet, C., & Dennis Jr, J. E. (2002). "Analysis of generalized pattern searches", *SIAM Journal on Optimization*, (pp. 889-903)
- [56] Torczon, V.J. (1997). "On the convergence of pattern search algorithms". *SIAM Journal on Optimization* 7, (pp. 1–25)
- [57] Torczon, Virginia, (1995), "Pattern Search Methods for Nonlinear Optimization", *SIAG/OPT Views-and-News: A Forum for the SIAM Activity Group on Optimization*, (pp. 7 – 11)
- [58] Saldju T. and David A.L. (1999, July). "Covariance Estimation With Limited Training Samples", *IEEE Transactions on Geoscience and Remote Sensing*.
- [59] Gustavo C.V., Luis G.C., Jordi M.M., Joan V.F., Javier C.M. (2006). "Composite Kernels for Hyperspectral Image Classification." *IEEE Geoscience And Remote Sensing Letters*, (pp. 93-97)
- [60] Bor-Chen K., Chih-Sheng H., Chih-Cheng H., Yu-Lung L., I-Ling C. (2010). "Spatial Information Based Support Vector Machine for Hyperspectral Image Classification", *IGARSS*, (pp. 832-835)
- [61] Cheng-Hsuan L., Hui-Shan C., Bor-Chen K. and Chin-Teng L. (2011), "Hyperspectral Image Classification Using Spectral and Spatial Information Based Linear Disciriminant Analysis", *IGARSS*, (pp. 1716-1719)
- [62] Farid M. and Lorenzo B. (2002). "Support Vector Machines for Classification of Hyperspectral Remote-Sensing Images", *IGARSS*, (pp. 506-508)
- [63] Naif A., Yakoub B., Farid M., Ronald R.Y. (2012) "Fusion of supervised and unsupervised learning for improved classification of hyperspectral images", *Information Sciences*, (pp. 39-55)

- [64] Yuliya T., Jón Atli B., Jocelyn C. and James C. T. (2010, November). "Multiple Spectral-Spatial Classification Approach for Hyperspectral Data", *IEEE Transactions On Geoscience And Remote Sensing*, (pp. 4122-4132).
- [65] Antonio P., Pablo M., Javier P., and Rosa P. (2005, March), "Dimensionality Reduction and Classification of Hyperspectral Image Data Using Sequences of Extended Morphological Transformations", *IEEE Transactions on Geoscience And Remote Sensing*, (pp. 466-479)
- [66] Daniele C., Rupert M. and Peter R. (2013, May). "A Classification Algorithm for Hyperspectral Images Based on Synergetics Theory." *IEEE Transactions On Geoscience And Remote Sensing*, (pp. 2887-2898)
- [67] Haken, H. (2007). "Synergetics", Retrieved from;
<http://www.scholarpedia.org/article/Synergetics>

APPENDICES

APPENDIX-A INDIAN PINES SCENE RESULTS

First N Sample without Pre-Processing or Post Processing

Table 4 IPS - ML -First N Sample without Pre-Processing or Post Processing

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	981	123	0	0	1	0	0	0	210	19	88	0	0	6	0	981	1428	68,70%	54,71%	60,91%
Corn-mintill	0	83	539	0	0	0	0	0	0	91	82	33	0	0	2	0	539	830	64,94%	68,14%	66,50%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	4	2	0	418	0	0	5	0	1	2	7	0	0	44	0	418	483	86,54%	99,05%	92,38%
Grass-trees	0	0	0	0	2	600	0	0	0	1	0	4	0	2	121	0	600	730	82,19%	99,50%	90,02%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	98,96%	99,48%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	110	3	0	0	0	0	0	0	695	127	27	0	0	10	0	695	972	71,50%	40,43%	51,65%
Soybean-mintill	0	586	71	0	0	1	0	0	0	716	801	261	0	0	19	0	801	2455	32,63%	77,54%	45,93%
Soybean-clean	0	29	53	0	1	0	0	0	0	5	2	501	0	0	2	0	501	593	84,49%	54,40%	66,18%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	1	1	0	0	0	0	0	0	0	1038	225	0	1038	1265	82,06%	99,81%	90,07%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	386	100,00%	47,36%	64,28%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					66,91%

Table 11 IPS – K-NN - First N Sample with Pre-Processing (PCA)

Table 23 IPS – K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	819	364	0	8	0	0	1	0	43	16	161	0	0	16	0	819	1428	57,35%	50,52%	53,72%
Corn-mintill	0	36	558	0	0	0	0	0	0	75	145	10	0	1	5	0	558	830	67,23%	45,51%	54,28%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	0	0	0	423	0	0	16	0	0	6	0	0	0	38	0	423	483	87,58%	91,16%	89,33%
Grass-trees	0	0	0	0	5	700	0	0	0	0	0	0	0	0	25	0	700	730	95,89%	99,01%	97,43%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	96,57%	98,25%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	87	41	0	1	0	0	0	0	707	78	49	0	0	9	0	707	972	72,74%	62,73%	67,37%
Soybean-mintill	0	632	187	0	8	3	0	0	0	299	1083	226	0	5	12	0	1083	2455	44,11%	81,25%	57,18%
Soybean-clean	0	47	76	0	19	4	0	0	0	3	5	437	0	0	2	0	437	593	73,69%	49,49%	59,21%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	0	0	0	0	0	0	0	0	0	979	286	0	979	1265	77,39%	99,39%	87,02%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	386	100,00%	49,55%	66,27%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					68,30%

First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 23 IPS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 28 IPS – SVM-RBF -Randomly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 29 IPS – SVM-LNR -Randomly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 34 IPS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA)

Table 40 IPS – SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 41 IPS – SVM-LNR - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 42 IPS – K-NN - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 43 IPS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly-Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	1257	7	0	1	0	0	0	0	59	55	39	0	1	9	0	1257	1428	88,03%	90,37%	89,18%
Corn-mintill	0	12	770	0	1	0	0	0	0	2	14	24	0	3	4	0	770	830	92,77%	97,84%	95,24%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	0	0	0	472	4	0	0	0	0	0	7	0	0	0	0	472	483	97,72%	99,58%	98,64%
Grass-trees	0	0	0	0	0	726	0	0	0	0	0	0	0	1	3	0	726	730	99,45%	99,32%	99,38%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	100,00%	100,00%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	0	0	0	0	0	0	0	0	950	13	5	0	1	3	0	950	972	97,74%	84,15%	90,43%
Soybean-mintill	0	120	6	0	0	0	0	0	0	118	2100	89	0	6	16	0	2100	2455	85,54%	95,93%	90,44%
Soybean-clean	0	2	4	0	0	0	0	0	0	0	7	577	0	0	3	0	577	593	97,30%	77,87%	86,51%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	0	1	0	0	0	0	0	0	0	1239	25	0	1239	1265	97,94%	99,04%	98,49%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	386	100,00%	85,97%	92,46%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					93,09%

Table 46 IPS – K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	1217	3	0	3	0	0	0	0	68	71	50	0	2	14	0	1217	1428	85,22%	91,85%	88,41%
Corn-mintill	0	4	791	0	0	0	0	0	0	4	11	9	0	8	3	0	791	830	95,30%	94,05%	94,67%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	0	0	0	483	0	0	0	0	0	0	0	0	0	0	0	483	483	100,00%	97,58%	98,77%
Grass-trees	0	0	0	0	0	725	0	0	0	0	0	0	0	0	5	0	725	730	99,32%	99,59%	99,45%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	100,00%	100,00%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	0	0	0	2	1	0	0	0	956	3	6	0	0	4	0	956	972	98,35%	84,75%	91,05%
Soybean-mintill	0	101	47	0	4	2	0	0	0	100	2164	18	0	2	17	0	2164	2455	88,15%	96,22%	92,01%
Soybean-clean	0	3	0	0	0	0	0	0	0	0	0	589	0	0	1	0	589	593	99,33%	87,65%	93,12%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	3	0	0	0	0	0	0	0	0	1218	44	0	1218	1265	96,28%	99,02%	97,64%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	386	100,00%	81,43%	89,77%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					93,63%

Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 47 IPS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 54 IPS – K-NN - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Uniformly Selected N Sample with Pre-Processing (PCA)

Table 55 IPS – ML - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 56 IPS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 57 IPS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 58 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA)

Table 66 IPS – K-NN - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 67 IPS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	1188	6	0	0	0	0	0	0	78	73	69	0	4	10	0	1188	1428	83,19%	92,81%	87,74%
Corn-mintill	0	9	759	0	0	0	0	0	0	2	42	12	0	3	3	0	759	830	91,45%	98,06%	94,64%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	0	0	0	473	0	0	0	0	0	0	3	0	2	5	0	473	483	97,93%	100,00%	98,95%
Grass-trees	0	0	0	0	0	725	0	0	0	0	0	0	0	1	4	0	725	730	99,32%	99,86%	99,59%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	100,00%	100,00%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	1	0	0	0	0	0	0	0	945	12	7	0	3	4	0	945	972	97,22%	84,91%	90,65%
Soybean-mintill	0	82	9	0	0	1	0	0	0	88	2098	155	0	7	15	0	2098	2455	85,46%	94,21%	89,62%
Soybean-clean	0	0	0	0	0	0	0	0	0	0	2	590	0	0	1	0	590	593	99,49%	70,57%	82,58%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	0	0	0	0	0	0	0	0	0	1246	19	0	1246	1265	98,50%	98,03%	98,26%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	5	381	0	381	386	98,70%	86,20%	92,03%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					92,34%

Table 70 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Alfalfa	Corn-notill	Corn-mintill	Corn	Grass-pasture	Grass-trees	Grass-pasture-mowed	Hay-windrowed	Oats	Soybean-notill	Soybean-mintill	Soybean-clean	Wheat	Woods	Buildings-Grass-Trees-Drives	Stone-Steel-Towers	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Alfalfa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn-notill	0	1235	28	0	9	0	0	0	0	56	45	39	0	0	16	0	1235	1428	86,48%	95,44%	90,74%
Corn-mintill	0	6	804	0	1	0	0	0	0	4	8	0	0	5	2	0	804	830	96,87%	89,04%	92,79%
Corn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grass-pasture	0	0	0	0	482	0	0	0	0	0	0	0	0	1	0	0	482	483	99,79%	96,21%	97,97%
Grass-trees	0	0	0	0	0	726	0	0	0	0	0	0	0	1	3	0	726	730	99,45%	99,73%	99,59%
Grass-pasture-mowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hay-windrowed	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0	478	478	100,00%	100,00%	100,00%
Oats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Soybean-notill	0	0	1	0	4	1	0	0	0	953	3	4	0	0	6	0	953	972	98,05%	84,19%	90,59%
Soybean-mintill	0	53	67	0	4	1	0	0	0	118	2161	29	0	2	20	0	2161	2455	88,02%	97,43%	92,49%
Soybean-clean	0	0	3	0	0	0	0	0	0	1	1	588	0	0	0	0	588	593	99,16%	89,09%	93,85%
Wheat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Woods	0	0	0	0	1	0	0	0	0	0	0	0	0	1215	49	0	1215	1265	96,05%	99,26%	97,63%
Buildings-Grass-Trees-Drives	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	386	100,00%	80,08%	88,94%
Stone-Steel-Towers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																					93,85%

Table 71 IPS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 74 IPS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

APPENDIX-B: SALINAS SCENE RESULTS

First N Sample without Pre-Processing or Post Processing

Table 75 SSC - ML -First N Sample without Pre-Processing or Post Processing

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	1865	144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1865	2009	92.83%	100.00%	96.28%
Brocoli Green Weeds 2	0	3699	0	0	0	0	0	3	0	0	0	0	0	24	0	0	3699	3726	99.28%	96.25%	97.74%
Fallow	0	0	1976	0	0	0	0	0	0	0	0	0	0	0	0	0	1976	1976	100.00%	76.21%	86.50%
Fallow Rough Plow	0	0	2	1392	0	0	0	0	0	0	0	0	0	0	0	0	1392	1394	99.86%	98.58%	99.22%
Fallow Smooth	0	0	608	20	2044	4	0	1	0	1	0	0	0	0	0	0	2044	2678	76.33%	100.00%	86.57%
Stubble	0	0	0	0	0	3953	0	6	0	0	0	0	0	0	0	0	3953	3959	99.85%	99.90%	99.87%
Celery	0	0	0	0	0	0	3573	6	0	0	0	0	0	0	0	0	3573	3579	99.83%	100.00%	99.92%
Grapes Untrained	0	0	0	0	0	0	0	3990	0	4	0	0	0	0	7276	1	3990	11271	35.40%	75.35%	48.17%
Soil Vinyard Develop	0	0	0	0	0	0	0	14	5804	371	2	12	0	0	0	0	5804	6203	93.57%	99.95%	96.65%
Corn Senesced Green Weeds	0	0	0	0	0	0	0	163	3	3081	24	0	0	6	0	1	3081	3278	93.99%	87.65%	90.71%
Lettuce Romaine 4wk	0	0	0	0	0	0	0	0	0	1	1067	0	0	0	0	0	1067	1068	99.91%	97.62%	98.75%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	19	0	1902	6	0	0	0	1902	1927	98.70%	99.37%	99.04%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	914	2	0	0	914	916	99.78%	98.92%	99.35%
Lettuce Romaine 7wk	0	0	0	0	0	0	0	2	0	9	0	0	4	1055	0	0	1055	1070	98.60%	97.06%	97.82%
Vinyard Untrained	0	0	7	0	0	0	0	1108	0	0	0	0	0	0	6153	0	6153	7268	84.66%	45.82%	59.46%
Vinyard Vertical Trellis	0	0	0	0	0	0	0	2	0	29	0	0	0	0	0	1776	1776	1807	98.28%	99.89%	99.08%
																					81.74%

Table 76 SSC - SVM-RBF -First N Sample without Pre-Processing or Post Processing

[illegible]

Table 77 SSC - SVM-LNR -First N Sample without Pre-Processing or Post Processing

[illegible]

Table 84 SSC - SVM-RBF - First N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 85 SSC - SVM-LNR - First N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 94 SSC - K-NN - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	1892	117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1892	2009	94,18%	100,00%	97,00%
Brocoli Green Weeds 2	0	3682	0	0	0	0	11	0	0	0	0	0	0	2	0	31	3682	3726	98,82%	96,92%	97,86%
Fallow	0	0	1456	0	0	0	0	0	0	58	461	1	0	0	0	0	1456	1976	73,68%	66,21%	69,75%
Fallow Rough Plow	0	0	0	1362	32	0	0	0	0	0	0	0	0	0	0	0	1362	1394	97,70%	99,63%	98,66%
Fallow Smooth	0	0	679	5	1987	1	0	0	1	1	3	1	0	0	0	0	1987	2678	74,20%	97,83%	84,39%
Stubble	0	0	0	0	0	3948	0	5	0	0	0	1	0	5	0	0	3948	3959	99,72%	99,97%	99,85%
Celery	0	0	0	0	0	0	3572	0	0	0	0	1	1	0	0	5	3572	3579	99,80%	99,64%	99,72%
Grapes Untrained	0	0	0	0	0	0	0	2634	38	113	37	5	4	1	8435	4	2634	11271	23,37%	87,22%	36,86%
Soil Vinyard Develop	0	0	0	0	0	0	0	0	6055	34	114	0	0	0	0	0	6055	6203	97,61%	98,55%	98,08%
Corn Senesced Green Weeds	0	0	53	0	0	0	0	132	43	2634	109	19	4	6	278	0	2634	3278	80,35%	92,29%	85,91%
Lettuce Romaine 4wk	0	0	0	0	2	0	0	0	0	0	1063	3	0	0	0	0	1063	1068	99,53%	59,49%	74,47%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	0	1927	0	0	0	0	0	1927	1927	100,00%	97,67%	98,82%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	900	16	0	0	0	900	916	98,25%	94,94%	96,57%
Lettuce Romaine 7wk	0	0	0	0	0	0	2	5	7	3	0	0	39	1008	6	0	1008	1070	94,21%	97,11%	95,64%
Vinyard Untrained	0	0	11	0	0	0	0	244	0	8	0	4	0	0	7001	0	7001	7268	96,33%	43,90%	60,31%
Vinyard Vertical Trellis	0	0	0	0	10	0	0	0	0	3	0	11	0	0	228	1555	1555	1807	86,05%	97,49%	91,42%
																					78,84%

First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 95 SSC - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 100 SSC - SVM-RBF -Randomly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 101 SSC - SVM-LNR -Randomly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 106 SSC – K-NN - Randomly Selected N Sample with Pre-Processing (PCA)

Table 108 SSC - SVM-RBF - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 109 SSC - SVM-LNR - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 112 SSC - SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 113 SSC - SVM-LNR - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 115 SSC - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	2009	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2009	2009	100.00%	100.00%	100.00%
Brocoli Green Weeds 2	0	3726	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3726	3726	100.00%	100.00%	100.00%
Fallow	0	0	1976	0	0	0	0	0	0	0	0	0	0	0	0	0	1976	1976	100.00%	100.00%	100.00%
Fallow Rough Plow	0	0	0	1391	3	0	0	0	0	0	0	0	0	0	0	0	1391	1394	99.78%	99.36%	99.57%
Fallow Smooth	0	0	0	9	2666	2	0	0	0	0	0	0	0	0	1	0	2666	2678	99.55%	99.89%	99.72%
Stubble	0	0	0	0	0	3959	0	0	0	0	0	0	0	0	0	0	3959	3959	100.00%	99.95%	99.97%
Celery	0	0	0	0	0	0	3576	1	0	1	0	0	0	0	1	0	3576	3579	99.92%	100.00%	99.96%
Grapes Untrained	0	0	0	0	0	0	0	9846	0	34	0	0	0	0	1391	0	9846	11271	87.36%	88.94%	88.14%
Soil Vinyard Develop	0	0	0	0	0	0	0	0	6196	7	0	0	0	0	0	0	6196	6203	99.89%	99.65%	99.77%
Corn Senesced Green Weeds	0	0	0	0	0	0	0	2	22	3254	0	0	0	0	0	0	3254	3278	99.27%	98.67%	98.97%
Lettuce Romaine 4wk	0	0	0	0	0	0	0	0	0	0	1068	0	0	0	0	0	1068	1068	100.00%	100.00%	100.00%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	0	0	1927	0	0	0	0	1927	1927	100.00%	100.00%	100.00%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	915	1	0	0	915	916	99.89%	100.00%	99.95%
Lettuce Romaine 7wk	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1069	0	1069	1070	99.91%	99.91%	99.91%
Vinyard Untrained	0	0	0	0	0	0	0	1222	0	0	0	0	0	0	6046	0	6046	7268	83.19%	81.27%	82.22%
Vinyard Vertical Trellis	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1806	1806	1807	99.94%	100.00%	99.97%
																					95.01%

Table 118 SSC - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	2007	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2007	2010	99,90%	100,00%	99,95%
Brocoli Green Weeds 2	0	3726	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3726	3727	100,00%	99,89%	99,95%
Fallow	0	0	1976	0	0	0	0	0	0	0	0	0	0	0	0	0	1976	1977	100,00%	97,97%	98,97%
Fallow Rough Plow	0	0	0	1390	4	0	0	0	0	0	0	0	0	0	0	0	1390	1395	99,71%	99,14%	99,43%
Fallow Smooth	0	0	11	12	2654	1	0	0	0	0	0	0	0	0	0	0	2654	2679	99,10%	99,85%	99,48%
Stubble	0	0	0	0	0	3959	0	0	0	0	0	0	0	0	0	0	3959	3960	100,00%	99,92%	99,96%
Celery	0	0	0	0	0	1	3577	0	0	0	0	0	0	1	0	0	3577	3580	99,94%	100,00%	99,97%
Grapes Untrained	0	0	0	0	0	1	0	9224	4	241	26	0	0	14	1753	8	9224	11272	81,84%	87,73%	84,68%
Soil Vinyard Develop	0	0	0	0	0	0	0	0	6168	9	26	0	0	0	0	0	6168	6204	99,44%	99,55%	99,49%
Corn Senesced Green Weeds	0	0	30	0	0	0	0	0	24	3172	40	3	0	1	2	6	3172	3279	96,77%	92,26%	94,46%
Lettuce Romaine 4wk	0	0	0	0	0	0	0	0	0	0	1068	0	0	0	0	0	1068	1069	100,00%	92,07%	95,87%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	0	0	1927	0	0	0	0	1927	1928	100,00%	99,84%	99,92%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	916	0	0	0	916	917	100,00%	100,00%	100,00%
Lettuce Romaine 7wk	0	0	0	0	0	0	0	0	0	3	0	0	0	1067	0	0	1067	1071	99,72%	98,52%	99,12%
Vinyard Untrained	0	0	0	0	0	0	0	1290	0	13	0	0	0	0	5962	3	5962	7269	82,03%	77,26%	79,57%
Vinyard Vertical Trellis	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1805	1805	1808	99,89%	99,07%	99,48%
																					93,45%

Table 119 SSC - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 120 SSC - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 121 SSC - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 124 SSC - SVM-RBF - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 125 SSC - SVM-LNR - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 126 SSC - K-NN - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Uniformly Selected N Sample with Pre-Processing (PCA)

Table 127 SSC – ML - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 130 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Sail Vinayd Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure	
Brocoli Green Weeds 1	2000	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2000	2010	99.55%	100.00%	99.78%	
Brocoli Green Weeds 2	0	3717	0	0	0	0	0	0	0	0	0	0	0	9	0	0	3717	3727	99.76%	99.70%	99.73%	
Fallow	0	0	1969	0	6	0	0	0	0	0	1	0	0	0	0	0	1969	1977	99.65%	96.52%	98.06%	
Fallow Rough Plow	0	0	0	1388	6	0	0	0	0	0	0	0	0	0	0	0	1388	1395	99.57%	98.86%	99.21%	
Fallow Smooth	0	0	30	16	2626	0	0	0	2	0	3	0	0	1	0	0	2626	2679	98.06%	99.09%	98.57%	
Stubble	0	0	0	0	2	3952	0	0	0	0	0	0	0	5	0	0	3952	3960	99.82%	99.87%	99.85%	
Celery	0	0	0	0	1	0	3552	1	0	0	0	0	2	8	4	11	3552	3580	99.25%	99.36%	99.30%	
Grapes Untrained	0	0	0	0	0	2	19	7760	5	219	14	1	2	16	3230	3	7760	11272	68.85%	78.10%	73.18%	
Sail Vinyard Develop	0	0	0	0	0	0	0	12	6082	28	59	0	0	22	0	0	6082	6204	98.05%	99.38%	98.71%	
Corn Senesced Green Weeds	0	1	40	0	5	3	0	27	28	3091	41	15	0	12	8	7	3091	3279	94.30%	90.41%	92.31%	
Lettuce Romaine 4wk	0	0	0	0	1	0	0	0	0	2	1064	1	0	0	0	0	1064	1069	99.63%	90.02%	94.58%	
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	2	0	1925	0	0	0	0	1925	1928	99.90%	99.02%	99.46%	
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	913	3	0	0	913	917	99.67%	98.70%	99.19%	
Lettuce Romaine 7wk	0	0	0	0	0	0	0	0	2	2	2	0	0	8	1054	2	0	1054	1071	98.50%	92.95%	95.64%
Vinyard Untrained	0	0	1	0	1	0	2	2134	1	68	0	2	0	4	5029	26	5029	7269	69.19%	60.77%	64.71%	
Vinyard Vertical Trellis	0	1	0	0	2	0	2	0	0	7	0	0	0	0	2	1793	1793	1808	99.23%	97.45%	88.49%	

Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

Table 131 SSC - ML - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 132 SSC - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 133 SSC - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 136 SSC - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 137 SSC - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 138 SSC - K-NN - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 139 SSC - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	2009	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2009	2010	100,00%	100,00%	100,00%
Brocoli Green Weeds 2	0	3726	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3726	3727	100,00%	100,00%	100,00%
Fallow	0	0	1976	0	0	0	0	0	0	0	0	0	0	0	0	0	1976	1977	100,00%	100,00%	100,00%
Fallow Rough Plow	0	0	0	1390	4	0	0	0	0	0	0	0	0	0	0	0	1390	1395	99,71%	99,29%	99,50%
Fallow Smooth	0	0	0	10	2664	1	0	0	0	2	0	0	0	0	1	0	2664	2679	99,48%	99,85%	99,66%
Stubble	0	0	0	0	0	3957	1	1	0	0	0	0	0	0	0	0	3957	3960	99,95%	99,97%	99,96%
Celery	0	0	0	0	0	0	3577	0	0	2	0	0	0	0	0	0	3577	3580	99,94%	99,97%	99,96%
Grapes Untrained	0	0	0	0	0	0	0	10150	0	96	0	0	0	0	1025	0	10150	11272	90,05%	93,68%	91,83%
Soil Vinyard Develop	0	0	0	0	0	0	0	0	6156	47	0	0	0	0	0	0	6156	6204	99,24%	99,66%	99,45%
Corn Senesced Green Weeds	0	0	0	0	0	0	0	0	21	3257	0	0	0	0	0	0	3257	3279	99,36%	95,07%	97,17%
Lettuce Romaine 4wk	0	0	0	0	0	0	0	0	0	0	1068	0	0	0	0	0	1068	1069	100,00%	100,00%	100,00%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	0	0	1927	0	0	0	0	1927	1928	100,00%	100,00%	100,00%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	908	8	0	0	908	917	99,13%	100,00%	99,56%
Lettuce Romaine 7wk	0	0	0	0	0	0	0	0	0	7	0	0	0	1063	0	0	1063	1071	99,35%	99,25%	99,30%
Vinyard Untrained	0	0	0	0	0	0	0	684	0	0	0	0	0	0	6584	0	6584	7269	90,59%	86,52%	88,51%
Vinyard Vertical Trellis	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	1792	1792	1808	99,17%	100,00%	99,58%
																					96,42%

Table 142 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Brocoli Green Weeds 1	Brocoli Green Weeds 2	Fallow	Fallow Rough Plow	Fallow Smooth	Stubble	Celery	Grapes Untrained	Soil Vinyard Develop	Corn Senesced Green Weeds	Lettuce Romaine 4wk	Lettuce Romaine 5wk	Lettuce Romaine 6wk	Lettuce Romaine 7wk	Vinyard Untrained	Vinyard Vertical Trellis	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Brocoli Green Weeds 1	2008	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2008	2010	99,95%	100,00%	99,98%
Brocoli Green Weeds 2	0	3724	0	0	0	0	0	0	0	0	0	0	0	2	0	0	3724	3727	99,95%	99,97%	99,96%
Fallow	0	0	1976	0	0	0	0	0	0	0	0	0	0	0	0	0	1976	1977	100,00%	97,34%	98,65%
Fallow Rough Plow	0	0	0	1393	1	0	0	0	0	0	0	0	0	0	0	0	1393	1395	99,93%	99,08%	99,50%
Fallow Smooth	0	0	17	13	2643	1	0	0	2	2	0	0	0	0	0	0	2643	2679	98,69%	99,92%	99,30%
Stubble	0	0	0	0	1	3956	0	0	0	0	0	0	0	2	0	0	3956	3960	99,92%	99,95%	99,94%
Celery	0	0	0	0	0	0	3576	0	0	0	0	0	1	1	1	0	3576	3580	99,92%	99,94%	99,93%
Grapes Untrained	0	0	0	0	0	1	0	9208	2	216	9	2	0	6	1824	3	9208	11272	81,70%	87,32%	84,42%
Soil Vinyard Develop	0	0	0	0	0	0	0	2	6172	21	7	0	0	1	0	0	6172	6204	99,50%	99,48%	99,49%
Corn Senesced Green Weeds	0	0	37	0	0	0	0	0	28	3196	4	10	0	0	0	3	3196	3279	97,50%	92,50%	94,94%
Lettuce Romaine 4wk	0	0	0	0	0	0	0	0	0	0	1066	2	0	0	0	0	1066	1069	99,81%	98,16%	98,98%
Lettuce Romaine 5wk	0	0	0	0	0	0	0	0	0	0	0	1927	0	0	0	0	1927	1928	100,00%	99,28%	99,64%
Lettuce Romaine 6wk	0	0	0	0	0	0	0	0	0	0	0	0	912	4	0	0	912	917	99,56%	99,89%	99,73%
Lettuce Romaine 7wk	0	0	0	0	0	0	0	0	0	3	0	0	0	1067	0	0	1067	1071	99,72%	98,52%	99,12%
Vinyard Untrained	0	0	0	0	0	0	0	1335	0	17	0	0	0	0	5867	49	5867	7269	80,72%	76,27%	78,44%
Vinyard Vertical Trellis	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1805	1805	1808	99,89%	97,04%	98,45%
																					93,26%

Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 143 SSC - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 146 SSC – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

APPENDIX-C: PAVIA UNIVERSITY SCENE RESULTS

First N Sample without Pre-Processing or Post Processing

Table 147 PUS - ML -First N Sample without Pre-Processing or Post Processing

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	4487	1	562	2	22	12	788	751	6	4487	6631	67,67%	95,55%	79,23%
Meadows	10	5281	1	8193	0	4186	0	978	0	5281	18649	28,32%	95,76%	43,71%
Gravel	2	0	2070	1	1	0	1	24	0	2070	2099	98,62%	46,70%	63,38%
Trees	0	0	0	3046	0	8	0	0	10	3046	3064	99,41%	26,73%	42,13%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	98,10%	99,04%
Bare Soil	172	233	1	154	0	3698	0	766	5	3698	5029	73,53%	46,76%	57,17%
Bitumen	14	0	27	0	0	0	1219	70	0	1219	1330	91,65%	60,65%	72,99%
Self-Blocking Bricks	7	0	1772	0	1	4	2	1896	0	1896	3682	51,49%	42,27%	46,43%
Shadows	4	0	0	0	2	0	0	0	941	941	947	99,37%	97,82%	98,59%
														56,07%

Table 148 PUS - SVM-RBF -First N Sample without Pre-Processing or Post Processing

[illegible]

Table 149 PUS - SVM-LNR -First N Sample without Pre-Processing or Post Processing

[illegible]

Table 150 PUS - K-NN -First N Sample without Pre-Processing or Post Processing

[illegible]

First N Sample with Pre-Processing (PCA)

Table 151 PUS – ML - First N Sample with Pre-Processing (PCA)

[illegible]

Table 154 PUS – K-NN - First N Sample with Pre-Processing (PCA)

[illegible]

First N Sample with Post-Processing (Filtering with 3x3 window)

Table 155 PUS - ML - First N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 158 PUS - K-NN - First N Sample with Post-Processing (Filtering with 3x3 window)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	5720	0	48	0	2	10	462	387	2	5720	6631	86,26%	79,72%	82,86%
Meadows	328	7047	0	2926	0	8096	7	244	1	7047	18649	37,79%	88,07%	52,88%
Gravel	229	0	1770	0	0	3	30	67	0	1770	2099	84,33%	63,69%	72,57%
Trees	1	2	0	3030	4	26	0	1	0	3030	3064	98,89%	50,80%	67,12%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	99,34%	99,67%
Bare Soil	766	950	0	9	2	3089	0	213	0	3089	5029	61,42%	27,52%	38,01%
Bitumen	45	0	5	0	0	0	1278	2	0	1278	1330	96,09%	71,92%	82,27%
Self-Blocking Bricks	86	3	956	0	1	1	0	2635	0	2635	3682	71,56%	74,25%	72,88%
Shadows	0	0	0	0	0	0	0	0	947	947	947	100,00%	99,68%	99,84%
														62,79%

First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

Table 159 PUS - ML - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 160 PUS - SVM-RBF - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 161 PUS - SVM-LNR - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 162 PUS - K-NN - First N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 163 PUS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

[illegible]

Table 164 PUS - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

[illegible]

First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 167 PUS - ML - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 168 PUS - SVM-RBF - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 169 PUS - SVM-LNR - First N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Randomly Selected N Sample with Pre-Processing (PCA)

Table 175 PUS – ML - Randomly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 176 PUS – SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA)

[illegible]

Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)

Table 179 PUS - ML - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 180 PUS - SVM-RBF - Randomly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

Table 183 PUS - ML - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 184 PUS - SVM-RBF - Randomly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 187 PUS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	6411	4	59	3	0	17	14	123	0	6411	6631	96,68%	98,66%	97,66%
Meadows	0	18191	0	263	0	195	0	0	0	18191	18649	97,54%	99,77%	98,64%
Gravel	5	0	1919	3	0	0	0	172	0	1919	2099	91,42%	90,82%	91,12%
Trees	0	3	0	3051	0	10	0	0	0	3051	3064	99,58%	91,76%	95,51%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	99,70%	99,85%
Bare Soil	0	27	0	3	0	4999	0	0	0	4999	5029	99,40%	95,62%	97,47%
Bitumen	41	0	0	0	0	1	1288	0	0	1288	1330	96,84%	98,92%	97,87%
Self-Blocking Bricks	35	8	135	2	0	6	0	3496	0	3496	3682	94,95%	92,22%	93,56%
Shadows	6	0	0	0	4	0	0	0	937	937	947	98,94%	100,00%	99,47%
														97,34%

Table 190 PUS - K-NN - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	6088	1	59	0	0	21	303	159	0	6088	6631	91,81%	99,67%	95,58%
Meadows	1	17596	0	107	0	943	0	2	0	17596	18649	94,35%	98,87%	96,56%
Gravel	0	0	2052	0	0	2	0	45	0	2052	2099	97,76%	93,49%	95,58%
Trees	0	38	0	3005	0	21	0	0	0	3005	3064	98,07%	96,47%	97,26%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	100,00%	100,00%
Bare Soil	0	162	0	3	0	4862	0	2	0	4862	5029	96,68%	82,94%	89,28%
Bitumen	2	0	0	0	0	0	1328	0	0	1328	1330	99,85%	81,32%	89,64%
Self-Blocking Bricks	17	0	84	0	0	13	2	3566	0	3566	3682	96,85%	94,49%	95,65%
Shadows	0	0	0	0	0	0	0	0	947	947	947	100,00%	100,00%	100,00%
														95,35%

Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 191 PUS - ML - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 192 PUS - SVM-RBF - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 193 PUS - SVM-LNR - Randomly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 196 PUS - SVM-RBF - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 197 PUS - SVM-LNR - Uniformly Selected N Sample without Pre-Processing or Post Processing

[illegible]

Table 200 PUS – SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 201 PUS – SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA)

[illegible]

Table 202 PUS – K-NN - Uniformly Selected N Sample with Pre-Processing (PCA)

Table 204 PUS - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 205 PUS - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Filtering with 3x3 window)

[illegible]

Table 208 PUS - SVM-RBF - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 209 PUS - SVM-LNR - Uniformly Selected N Sample with Post-Processing (Majority Voting with Meanshift Segmentation)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	6532	13	0	0	0	0	0	85	1	6532	6631	98,51%	99,65%	99,07%
Meadows	0	17942	0	0	0	706	0	1	0	17942	18649	96,21%	99,36%	97,76%
Gravel	0	2	2096	0	0	1	0	0	0	2096	2099	99,86%	99,57%	99,71%
Trees	0	92	0	2924	0	39	0	6	3	2924	3064	95,43%	99,97%	97,65%
Painted Metal Sheets	0	0	0	0	1329	12	0	4	0	1329	1345	98,81%	99,92%	99,36%
Bare Soil	0	2	0	0	0	5027	0	0	0	5027	5029	99,96%	86,90%	92,97%
Bitumen	0	0	0	0	0	0	1330	0	0	1330	1330	100,00%	100,00%	100,00%
Self-Blocking Bricks	23	7	9	1	0	0	0	3640	2	3640	3682	98,86%	97,43%	98,14%
Shadows	0	0	0	0	1	0	0	0	946	946	947	99,89%	99,37%	97,64%

Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

Table 211 PUS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	6395	4	92	0	0	14	20	106	0	6395	6631	96,44%	98,93%	97,67%
Meadows	0	18093	0	305	0	251	0	0	0	18093	18649	97,02%	99,62%	98,30%
Gravel	9	2	1995	0	0	0	0	93	0	1995	2099	95,05%	92,23%	93,62%
Trees	0	4	0	3046	0	14	0	0	0	3046	3064	99,41%	90,82%	94,92%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	99,78%	99,89%
Bare Soil	0	53	0	2	0	4974	0	0	0	4974	5029	98,91%	94,60%	96,70%
Bitumen	22	0	0	0	0	0	1308	0	0	1308	1330	98,35%	98,49%	98,42%
Self-Blocking Bricks	33	6	76	1	0	5	0	3561	0	3561	3682	96,71%	94,71%	95,70%
Shadows	5	0	0	0	3	0	0	0	939	939	947	99,16%	100,00%	99,58%
														97,38%

Table 214 PUS - K-NN - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Filtering with 3x3 window)

	Asphalt	Meadows	Gravel	Trees	Painted Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadows	Correctly Classified	Ground Truth	Recall	Precision	F-Measure
Asphalt	6141	2	53	0	0	16	287	132	0	6141	6631	92,61%	99,69%	96,02%
Meadows	1	17554	0	142	0	951	0	1	0	17554	18649	94,13%	98,78%	96,40%
Gravel	5	0	2054	0	0	0	0	40	0	2054	2099	97,86%	92,52%	95,11%
Trees	0	34	0	3009	4	17	0	0	0	3009	3064	98,20%	95,46%	96,81%
Painted Metal Sheets	0	0	0	0	1345	0	0	0	0	1345	1345	100,00%	99,70%	99,85%
Bare Soil	0	180	0	0	0	4845	0	4	0	4845	5029	96,34%	83,03%	89,19%
Bitumen	1	0	0	0	0	0	1329	0	0	1329	1330	99,92%	82,04%	90,10%
Self-Blocking Bricks	12	0	113	1	0	6	4	3546	0	3546	3682	96,31%	95,25%	95,77%
Shadows	0	0	0	0	0	0	0	0	947	947	947	100,00%	100,00%	100,00%
														95,31%

Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

Table 215 PUS - ML - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 216 PUS - SVM-RBF - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

Table 217 PUS - SVM-LNR - Uniformly Selected N Sample with Pre-Processing (PCA) and Post-Processing (Majority Voting with Meanshift Segmentation)

[illegible]

TEZ FOTOKOPİSİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü	<input type="checkbox"/>
Sosyal Bilimler Enstitüsü	<input type="checkbox"/>
Uygulamalı Matematik Enstitüsü	<input type="checkbox"/>
Enformatik Enstitüsü	<input checked="" type="checkbox"/>
Deniz Bilimleri Enstitüsü	<input type="checkbox"/>

YAZARIN

Soyadı : Özdemir
Adı : Okan Bilge
Bölümü : Bilişim Sistemleri

TEZİN ADI (İngilizce) : An Investigation On Hyperspectral Image
Classifiers For Remote Sensing

TEZİN TÜRÜ : Yüksek Lisans ☒ Doktora ☐

1. Tezimin tamamından kaynak gösterilmek şartıyla fotokopi alınabilir. X
2. Tezimin içindekiler sayfası, özet, indeks sayfalarından ve/veya bir bölümünden X
kaynak gösterilmek şartıyla fotokopi alınabilir.
3. Tezimden bir (1) yıl süreyle fotokopi alınamaz. ☐

TEZİN KÜTÜPHANEYE TESLİM TARİHİ: