

Old Dominion University

ODU Digital Commons

Computational Modeling & Simulation
Engineering Theses & Dissertations

Computational Modeling & Simulation
Engineering

Summer 8-2020

Deep Learning for Remote Sensing Image Processing

Yan Lu

Old Dominion University, vcu.yanlu@gmail.com

Follow this and additional works at: https://digitalcommons.odu.edu/msve_etds



Part of the [Aerospace Engineering Commons](#), [Artificial Intelligence and Robotics Commons](#), and the [Remote Sensing Commons](#)

Recommended Citation

Lu, Yan. "Deep Learning for Remote Sensing Image Processing" (2020). Doctor of Philosophy (PhD), Dissertation, Computational Modeling & Simulation Engineering, Old Dominion University, DOI: 10.25777/9nwb-h685
https://digitalcommons.odu.edu/msve_etds/57

This Dissertation is brought to you for free and open access by the Computational Modeling & Simulation Engineering at ODU Digital Commons. It has been accepted for inclusion in Computational Modeling & Simulation Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

DEEP LEARNING FOR REMOTE SENSING IMAGE PROCESSING

by

Yan Lu

B.S. July 2004, Beijing Jiaotong University

M.S. July 2007, Chinese Academy of Sciences

M.S. May 2009, Virginia Commonwealth University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY

August 2020

Approved by:

Jiang Li (Director)

Rick McKenzie (Member)

Yuzhong Shen (Member)

Duc T. Nguyen (Member)

Hong Yang (Member)

ABSTRACT

DEEP LEARNING FOR REMOTE SENSING IMAGE PROCESSING

Yan Lu
Old Dominion University, 2020
Director: Dr. Jiang Li

Remote sensing images have many applications such as ground object detection, environmental change monitoring, urban growth monitoring and natural disaster damage assessment. As of 2019, there were roughly 700 satellites listing “earth observation” as their primary application. Both spatial and temporal resolutions of satellite images have improved consistently in recent years and provided opportunities in resolving fine details on the Earth's surface. In the past decade, deep learning techniques have revolutionized many applications in the field of computer vision but have not fully been explored in remote sensing image processing. In this dissertation, several state-of-the-art deep learning models have been investigated and customized for satellite image processing in the applications of landcover classification and ground object detection.

First, a simple and effective Convolutional Neural Network (CNN) model is developed to detect fresh soil from tunnel digging activities near the U.S. and Mexico border by using pan-sharpened synthetic hyperspectral images. These tunnels' exits are usually hidden under warehouses and are used for illegal activities, for example, by drug dealers. Detecting fresh soil nearby is an indirect way to search for these tunnels. While multispectral images have been used widely and regularly in remote sensing since the 1970s, with the fast advances in hyperspectral sensors, hyperspectral imagery is becoming popular. A combination of 80 synthetic hyperspectral channels with the original eight multispectral channels collected by the WorldView-2 satellite are used by CNN to detect fresh soil. Experimental results show that detection performance can be

significantly improved by the combination of synthetic hyperspectral images with those original multispectral channels.

Second, an end-to-end, pixel-level Fully Convolutional Network (FCN) model is implemented to estimate the number of refugee tents in the Rukban area near the Syrian-Jordan border using high-resolution multispectral satellite images collected by WorldView-2. Rukban is a desert area crossing the border between Syria and Jordan, and thousands of Syrian refugees have fled into this area since the Syrian civil war in 2014. In the past few years, the number of refugee shelters for the forcibly displaced Syrian refugees in this area has increased rapidly. Estimating the location and number of refugee tents has become a key factor in maintaining the sustainability of the refugee shelter camps. Manually counting the shelters is labor-intensive and sometimes prohibitive given the large quantities. In addition, these shelters/tents are usually small in size, irregular in shape, and sparsely distributed in a very large area and could be easily missed by the traditional image-analysis techniques, making the image-based approaches also challenging. The FCN model is also boosted by transfer learning with the knowledge in the pre-trained VGG-16 model. Experimental results show that the FCN model is very accurate and has less than 2% of error.

Last, we investigate the Generative Adversarial Networks (GAN) to augment training data to improve the training of FCN model for refugee tent detection. Segmentation based methods like FCN require a large amount of finely labeled images for training. In practice, this is labor-intensive, time consuming, and tedious. The data-hungry problem is currently a big hurdle for this application. Experimental results show that the GAN model is a better tool as compared to traditional methods for data augmentation. Overall, our research made a significant contribution to remote sensing image processing.

Copyright, 2020, by Yan Lu, All Rights Reserved.

This dissertation is dedicated to my parents, Peijin Mu and Jinshe Lu.

ACKNOWLEDGMENTS

Many people have contributed to the successful completion of this dissertation. I extend many, many thanks to my advisors and committee members for their patience and hours of guidance on my research and editing of this manuscript. I would like to offer my sincerest appreciation to my advisor, Dr. Jiang Li, who is the most hard-working person I have ever met. Without his guidance, I could not imagine how I started my research. His advice and support pushed me moving forward, surviving, and thriving in this challenging but fun journey. I would like to thank my committee members: Dr. Yuzhong Shen, Dr. Duc Nguyen, and Dr. Hong Yang. I am extremely grateful for their time and effort in helping me finish my dissertation. I would also like to express my heartfelt appreciation to our late department chair, Dr. Rick Mackenzie; many students benefited from his kind help. I am one of them. He will be forever missed by us. I would like to thank Dr. Chiman Kwan from Signal Processing Inc., Dr. Jonathan Graham from Norfolk State University and my lab mates Danielle, Kazi, Reshad, Shahab and Adam, thanks for all the support and help you gave to me. I would also like to express my deepest gratitude to my parents and family, without their unconditional love, support and sacrifice, I could never have gotten this done.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	ix
LIST OF FIGURES	x
 Chapter	
1. INTRODUCTION.....	1
1.1 PROBLEM STATEMENT	2
1.2 PROPOSED WORK.....	3
1.3 CONTRIBUTIONS OF THIS DISSERTATION	3
1.4 ORGANIZATION OF THE DISSERTATION.....	5
 2. BACKGROUND OF THE STUDY: METHODS AND MODELS.....	 7
2.1 REMOTE SENSING IMAGERY	7
2.2 PIXEL-BASED METHODS.....	10
2.3 OBJECT-BASED METHODS	11
2.4 DEEP LEARNING.....	12
2.5 CONVOLUTIONAL NEURAL NETWORK	24
2.6 TRANSFER LEARNING.....	32
 3. HYPERSPECTRAL IMAGE CLASSIFICATION BY CNN.....	 34
3.1 INTRODUCTION	34
3.2 MOTIVATION	35
3.3 METHODOLOGY	36
3.4 MODEL TRAINING AND EXPERIMENTAL RESULTS	42
3.5 CONCLUSION	52
 4. EFFECTIVE REFUGEE TENT EXTRACTION BY FCN.....	 53
4.1 INTRODUCTION	54
4.2 MOTIVATION	56
4.3 METHODOLOGY	58
4.4 MODEL TRAINING AND EXPERIMENTAL RESULTS	68
4.5 CONCLUSION	71

5. DATA AUGMENTATION BY GAN	73
5.1 INTRODUCTION	73
5.2 METHODOLOGY	75
5.3 MODEL TRAINING AND EXPERIMENTAL RESULTS	83
5.4 CONCLUSION	87
6. CONCLUSIONS AND FUTURE WORK.....	89
6.1 CONCLUSIONS	89
6.2 FUTURE WORK	90
REFERENCES	92
VITA	106
LIST OF PUBLICATIONS.....	107

LIST OF TABLES

Table	Page
1. Summary of Contributions.....	6
2. Summary of Traditional RS Image Classification Techniques.....	12
3. Description of 14 Classes.....	39
4. AUC Scores of CNN Models on Testing Areas.....	51
5. Extracted patched for the 4 classes.....	65
6. FCN Performances with Different Patch Size.....	68
7. Model Comparison on Validation Data.....	69
8. Tent Number Estimation by Different Models	71
9. Datasets-FR by Horizontal Flipping (first row), Right Rotating 90 Degrees (second row) and Vertical Flipping (second row) of the Original Training Images.....	80
10. Image Dataset-SC by Scaling and Cropping the Original Images by Factors of 1.5, 2.0 and 2.5.....	81
11. Images Dataset-N by Adding Gaussian, Poisson, and Salt and Pepper Noises.	82
12. Validation Performance Matrices by Synthetic Datasets with Input Size of 128x128	85
13. Validation Performance Matrices by Synthetic Datasets with Input Size of 96x96	86
14. Validation Performance Matrices by Synthetic Datasets with Input Size of 64x64	86
15. Validation Performance Matrices by Synthetic Datasets with Input Size of 32x32	87

LIST OF FIGURES

Figure	Page
1. Electromagnetic Spectrum	9
2. Landsat-7 satellite (left), AVIRIS system (middle) and MODIS system (right)	9
3. “Explaining away.”	13
4. Multi-layer Neural Network.....	14
5. Local Minima and Saddle Point	16
6. An Infinite Logistic Belief Net with Tied Weights.	18
7. Restricted Boltzmann Machine with 12 Visible Units and 3 Hidden Units.	19
8. Contrastive Divergence.....	21
9. Pre-training Procedure and Stacked RBM	24
10. Local Receptive Field.	25
11. Feature Maps	26
12. Max-Pooling.....	27
13. Input Layers, Convolutional Layers and Pooling Layer.....	27
14. CNN Model Structure.....	28
15. AlexNet Overall Structure	28
16. ReLU, Sigmoid and Hyperbolic Tangent Activation Functions	30
17. Vanishing Derivatives of Hyperbolic Tangent and Sigmoid Activation Functions.....	31
18. Dropout	32
19. Illegal Tunnels near U.S. and Mexico Border.....	35
20. EMAP Synthetic Hyperspectral Bands Generation and Soil Detection Framework	37

21. The CNN Model Structure in Soil Detection.....	37
22. Soil Samples in WV-2.....	38
23. All 14 Classes of Training Samples in Soil Detection.....	39
24. Testing Images and Ground Truth Masks in Soil Detection.....	40
25. Soil Detection Results for the Testing Image Dated 3/19/2010.....	43
26. Soil Detection Results for the Testing Image Dated 10/11/2010.....	44
27. Soil Detection Results for the Testing Image Dated 12/2/2010.....	46
28. Post-processing Results for Testing Image Dated 3/19/2010.....	47
29. Post-processing Results for Testing Image Dated 10/11/2010.....	49
30. Post-Processing Results for Testing Image Dated 10/11/2010.....	50
31. A Typical Refugee Camp in the Rukban Area.....	55
32. System Architecture of Our FCN Model.....	59
33. CNN Structures. Top: CNN-7. Middle: CNN-5. Bottom: CNN-3.....	61
34. Summary of Models of R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN.....	63
35. Mask R-CNN with ResNet-50 Backbone for Tent Detection.....	64
36. Masks of the 4 Classes for Training CNN.....	65
37. Training Data for FCN.....	66
38. Validation Data and Ground Truth from Time-2.....	66
39. FCN Tent Detection Map in the Common Area from the Cropped Time-2 Data.....	68
40. Tent Extraction Maps in Time-1 (1st row), Time-2 (2nd row) and Validation Images (3rd row). From the left to right in each row: The Original RGB Images, Results by SAM, CNN- 3, CNN-5, CNN-7, Mask-RCNN and FCN-32.....	70
41. GAN Network Model.....	74

42. SinGAN's Multi-scale Pipeline.....	76
43. Single Scale/Level Generation.....	77
44. GAN Generated Samples: (a) Original Training Samples O1, (b)-(d): SinGAN-generated Data G1, G2 and G3	78

CHAPTER 1

INTRODUCTION

Remote sensing (RS) plays a critical role in many aspects of earth observation tasks such as ground object detection, climate and environmental change monitoring, urban growth monitoring and natural disaster damage assessment. RS images provide detailed global observation and insights of ecological health and sustainability for both natural and anthropogenic activities. As of 2019, there were roughly 700 satellites listing “earth observation” as their primary application. With the increasing commercial players in the sector in recent years, the consistent improvements of both spatial and temporal resolutions of satellite images provide more opportunities in resolving fine details on the Earth's surface. For instance, the Worldview-2 satellite has an average revisit time of 1.1 days on the Earth surface, and it is capable of collecting up to 1 million square kilometers of 8-band imagery per day and provides 0.46-meter panchromatic resolution and 1.84-meter multispectral resolution [1]. The Sentinel-2 satellite acquires 6 TB of data every day, a full image of the Earth is acquired every five days [2].

Multispectral images (MSI) usually refer to satellite images with 3 to 10 bands. MSIs are widely and regularly used since 1970s. Meanwhile, hyperspectral imaging (HSI) known as imaging spectrometry is becoming popular recently. HSIs usually contain hundreds or thousands of bands with much narrower spectral bandwidth (10-20 nm) than multispectral images. Each pixel in a hyperspectral image can be regarded as a high-dimensional vector corresponding to the spectral reflectance from hundreds of continuous narrow spectral channels within a specific wavelength range. The current HSI acquisition technologies can offer not only high spectral resolution but also high spatial resolution. MSIs and HSIs data are able to convey very complex

characteristics and richer spectral and spatial information. HSIs are expected to be more effective and accurate for advanced image analysis in a variety of quality demanding earth observation tasks such as target identification [3-5] and anomalous materials and object detection [6, 7]. Significant efforts have been made in the past few years to develop a variety of methods for object detection by RS images.

1.1 Problem Statement

Machine learning has been widely used in remote sensing image analysis for many years. The applications for multispectral or hyperspectral satellite image classification tasks often used random forests [8, 9], support vector machines [10-16], or decision trees [10, 17] in the earlier years. These automatic or semi-automated machine learning methods are able to improve the efficiency of analytic workflows, which have been conducted in pixel-based or object-based classification [18, 19], rule-based object classification [20, 21], and mathematical morphology-based classification [22, 23]. However, these machine learning approaches usually include hand-crafted features, and their performances highly rely on quality of the hand-crafted features. MSIs and HSIs data have very high dimensionalities and optimally “hand-crafting” the best feature representations are usually impossible.

The rich information in MSIs and HSIs data provides opportunities for many applications but also renders major challenges: 1) Processing a large amount of inherently non-linearly related high dimensional spatial and spectral data is computationally expensive, 2) labeling remote sensing images is label intensive and there are not enough training data to train machine learning models, and 3) detecting small, cluttered ground objects in MSI and HSI is a non-trivial task since those

objects are usually small in size, irregular in shape, and sometimes partially overlapped. We will explore these challenges in this dissertation.

1.2 Proposed Work

Deep learning networks have been widely applied for computer vision and achieved remarkable performances in a wide range of computer vision tasks [24-27]. Compared with traditional machine learning methods, deep learning models can automatically learn to extract hierarchy image features in an end-to-end manner - hand-craft feature extraction and post-processing procedures are not needed. In order to address these aforementioned challenges, we will investigate state-of-the-art deep learning models including CNNs, FCNs and Mask R-CNNs for small object detection in MSI and HSI data. In addition, we will explore GAN models for data augmentation to tackle the data-hungry issue in training large FCN models for MSI data.

1.3 Contributions of This Dissertation

First, we proposed a CNN model for soil detection near the U.S. and Mexico border by using 88 channels of synthetic HSIs data obtained by the extended morphological attributes profile (EMAP) [28] method. Our results show that with the synthetic hyperspectral bands, the proposed CNN model achieved a significant better performance. The area under the curve (AUC) scores of the receiver operating characteristic (ROC) curve of the CNN model have been improved both in high spatial resolution and low spatial resolution images. The largest improvement of AUC is 28.74% while the average AUC improvements are 9.02% in high-resolution images and 7.42% in low resolution images as compared to the results by using the eight original multispectral data

alone. AUCs are further improved by 3.31% in high resolution images and by 2.85% in low resolution images, respectively, with post-processing steps.

Second, we proposed an end-to-end pixel-level FCN model to extract refugee tents near the Syrian and Jordan border in Worldview-2 satellite imagery. By implementing bilinear interpolation deconvolution to up-sample the feature maps in the convolutional layers, the FCN model can utilize both the spectral and spatial information in the remote sensing imagery to generate outputs. The FCN model is scale-free and is able to analyze images with any sizes that are larger than the input patch size. We also applied transfer learning to initialize the FCN model with the pre-trained VGG-16 model [29] and then fine-tuned the FCN model with a small training dataset that was manually labeled for tent extraction. The transfer learning [30] strategy mitigated the lack of training data issues and significantly improved the performances of the FCN model. Our experimental results show that the FCN model improved overall accuracy by 4.49%, 3.54%, and 0.88% as compared to the spectrum angle mapper (SAM), CNNs and Mask R-CNN models, and improved precision by 34.61%, 41.99% and 11.87%, respectively.

In the last part of this dissertation, we proposed the use of GAN to generate synthetic training samples to furtherly improve the training of the FCN model. We compared the GAN augmentation method with other traditional methods, including flipping, rotating, scaling, and adding Gaussian, Poisson or salt and pepper noises. Our experimental results show that with GAN generated data samples, the overall performance of the FCN model is improved by 0.2-1.5% over the classical augmentation methods.

A summary of the dissertation contributions is listed in Table 1. We have several peer-reviewed publications: the research related to topic 1 is published in the 2018 IEEE Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) [31]. Research

related to topic 2 is published in IEEE Geoscience and Remote Sensing Letters [32]. Topic 3 research is published in 2019 Winter Simulation Conference [33] and 2019 UEMCON [34].

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. Chapter 2 provides the overall background information of my dissertation. It includes a literature review for the research of remote sensing image classification, object detection, and deep learning techniques used in this dissertation. Chapter 3 presents data, models, and experimental results for soil detection in the proposed research topic 1. Chapter 4 discusses data, models, and experimental results for the refugee tent detection near the Syria-Jordan border, as proposed in research topic 2. Chapter 5 presents the proposed GAN-based data augmentation model for improving the refugee tent detection by the FCN model. Chapter 6 concludes this dissertation with a summary and suggestions for future work.

TABLE 1. Summary of Contributions

Topic	Contributions
1.Deep Learning with Synthetic Hyperspectral Images for Improved Soil Detection in Multispectral Imagery	In this research, we presented a four layers deep convolutional neural network (CNN) model for soil detection by using the combination of 80 synthetic hyperspectral bands and its original eight multispectral bands, which are collected by the WorldView-2 satellite. We applied the CNN model onto a set of high-resolution data created by pan-sharpening the original multispectral bands and its synthetic hyperspectral bands. Our results indicate that by using the pan-sharpened synthetic hyperspectral bands, the performance of the CNN model for soil detection has been significantly improved [31].
2.Deep Learning for Effective Refugee Tent Extraction near Syrian-Jordan Border	In this research, we presented an FCN model to tackle the small ground objects detection problem in Worldview-2 (WV-2) satellite images and applied it to the refugee tent extraction problem. We transferred knowledge in the pre-trained VGG-16 model to improve the detection accuracy and network training convergence. We compared the proposed approach with the traditional spectral angle mapper (SAM) method, CNNs models, and the Mask R-CNN model. Experimental results show that the FCN model significantly improved the overall performance as compared other competing models [32].
3.Generative Adversarial Network for improving deep learning-based image classification	In this research, we proposed a data augmentation method by using GAN to generate synthetic image samples to improve the performance of the deep learning-based image classification models [33, 34]. We applied the GAN generated training data samples to the FCN model in research topic 2, the overall performance was improved. The intersection over union (IoU) scores were improved ranging from 0.2-1.5% as compared to without the augmented data.

CHAPTER 2

BACKGROUND OF THE STUDY: METHODS AND MODELS

This Chapter includes the literature review in remote sensing imagery, remote sensing image classification, deep learning, convolutional neural network models and transfer learning.

2.1 Remote Sensing Imagery

Remote Sensing (RS) image processing has had a very long tradition since the 1840s. It began with the invention of the camera more than 150 years ago. As noted by Avery and Berlin [35] as well as Baumann [36], “the RS imagery can be defined as any process whereby the information is gathered by the reflectance of light energy from an external source such as sun without being contact with it by any devices.” It is like how our human eyes work. Remote sensing has become associated more specifically with the Earth's surface monitoring with electromagnetic spectrum by satellites in nowadays [37], and the electromagnetic spectrum is shown in Fig 1. Through the fast development and wide utilization of RS satellites, RS image has become a major tool for data acquisition on the entire Earth surface. The revisit time ranges from a couple of days to a matter of hours [38, 39]. Many GIS applications integrate RS images for various analyses, particularly for those involved in natural resources [40-42].

The RS sensors can be divided into two types - passive and active. The passive sensors do not supply energy to objects being detected, and they are mostly used for measuring and recording the reflection of light off Earth objects' features. The aerial photography is a major form of remote sensing by passive sensors, and it can collect from visible to near-infrared wavelength, or even longer wavelength from the solar radiation. In contrast, the active sensors supply their own source

of energy, the flash photography and Radio Detection and Ranging (RADAR) system are examples. The RADAR system emits energy with wavelengths in the microwave section in the electromagnetic spectrum, as shown in Fig. 1, the reflection of this energy from the earth's surface produces RADAR images.

The electromagnetic spectrum is very broad. However, not all wavelengths are equally effective for RS imaging. Besides, most of the shorter wavelengths such as ultraviolet will be absorbed by the atmosphere and the glass lenses of sensors. The most commonly used in RS research are the visible, near-infrared mid-infrared, and thermal infrared bands. From the spectrum as shown in Fig.1, the visible wavelengths reside in the first section of the spectrum chart. The red, green, blue, and near-infrared wavelengths can all provide substantial good opportunities for observing the earth's surface without significant interference by atmosphere or the sensor itself. The middle infrared wavelengths and the thermal regions can be beneficial in many geological applications by monitoring heat distributions from industrial, animals, or the soil moisture conditions. After the thermal infrared, the area in the microwave region has significant importance in environmental RS imagery especially for the use of active radar imaging. This is not only because it responds significantly to the texture of the earth surface but it also supplements the information gained in other wavelengths such as offering night vision for the regions that are consistently covered by cloud. The radar imaging will not be affected significantly by clouds.

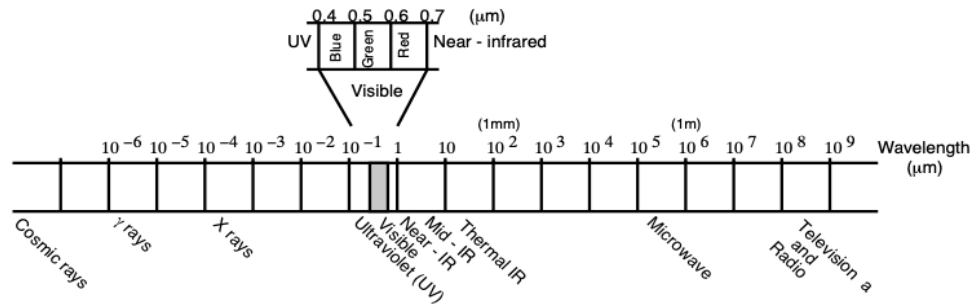


Fig. 1. Electromagnetic Spectrum from Lillesand, Kiefer et al. 1987 [43]

By collecting reflected spectral responses over a range of wavelengths, the sensor forms spectral response patterns, and these patterns form spectrum signatures. The spectrum signatures could be from the multi-spectrum bands such as the RS images that the LANDSAT Thematic Mapper system collects [44], which provide multi-spectral imagery in seven spectral bands at 30 meters resolution. The spectrum signatures could also have more bands such as AVIRIS [45] and MODIS[46] system cover similar wavelength ranges but with much narrow band width. Fig. 2 shows the Landsat-7 satellite, AVIRIS and MODIS systems.

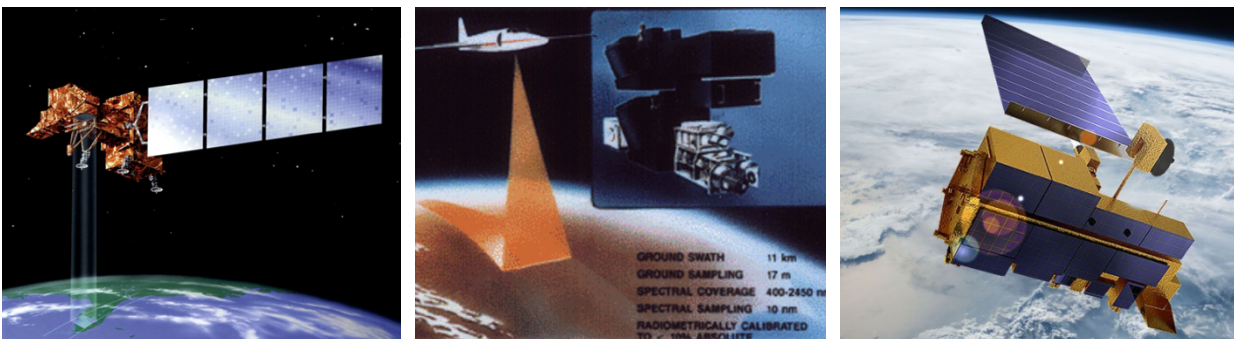


Fig. 2. Landsat-7 satellite (left), AVIRIS system (middle) and MODIS system (right)

To distinguish spectral patterns from the spectrum is the key to most of the procedures for computer-assisted interpretation in RS image processing. In early days it was believed that different materials would have distinctive spectrum signatures. However, in reality it is not often the case. Similar objects may have very different spectrum signatures while different objects may have very similar spectrum signatures. Finding the most effective bands or features to reflect the characteristics for different classes of objects is never a trivial task. Most of the time, it is laborious and time-consuming. Most importantly, in traditional spectrum analysis, it only utilized the spectrum response patterns and sometimes it was often simply the color features. The texture, size, shape, or context information were ignored even though RS image classification techniques had been developed since 1980s.

2.2 Pixel-Based Methods

Pixel-based methods employed pixel as the basic unit of analysis. Based on spectral reflectance of pixel, a series of classification techniques had been developed such as unsupervised methods: k-means [47], principle component analysis (PCA) [48-50] and ISODATA [47, 51], and supervised methods including maximum likelihood [10, 52], artificial neural networks [53-55], decision trees [10, 56, 57], support vector machine [10, 13, 15, 16], random forest [8, 9] and hybrid classification [12]. Pixel-based methods are easy to implement. However, the pixel-based methods only take consideration of individual pixels without their neighboring pixels and it purely relies on spectral characteristics. To resolve this limitation, fuzzy classification [20, 58], spectral mixture analysis [59, 60] and some post-classification approaches [28, 61] were introduced.

2.3 Object-Based Methods

Object-based classification methods [12, 18, 19, 58, 62-64] have been developed since the early 2000s with the launch of the very high-resolution remote sensing satellites such as QuickBird [65] and WorldView-2 [1]. The object-based methods are built up on the homogeneous properties within a group of pixels instead of individual pixels. Object-based image classification involved the identification of image objects or segments which are spatially contiguous with similar texture, color and tone. Object-based methods are more effective than the pixel-based methods, since the approaches allow for consideration of shape, size, and context as well as the spectral features. The grouped object pixels enhance the complexity of the high-resolution scene which might be involving shadows, changes, and delineating the corresponding physical features such as shapes of the objects [66, 67].

Though a large number of pixel-based and object-based RS image classification methods have been developed, these methods are still limited because they only utilize spectral characteristics and spatial features are more or less ignored. Many RS image classification tasks remain challenging due to high intra-class variations and low inter-class disparities.

Later, new models were developed to incorporate spatial context information such as shape, connectivity, contiguity, distance, or direction amongst adjacent objects. However, these methods achieved less satisfactory results since they required pre-defined “spatio-contextual” information about shape, size, or structure information of the objects. A group of studies had been originated to address this “spatio-contextual” issues by incorporating geographic models [68] or geostatistics [69] into RS image processing. These methods were accepted in geography, geology and economics but rarely used in RS image classification. Table 2 summarizes traditional RS image classification methods.

TABLE 2. Summary of Traditional RS Image Classification Techniques

Category	Method	Example
Pixel-Based	Rely on spectral characteristic solely on individual pixels	K-means, ISODATA, K-nearest Neighbors, Random forest, SVM
Object-Based	Incorporating characteristic and spatial information such as objects shape, connectivity, direction contiguity	Neural networks, Regression model, Fuzzy-spectral mixture analysis, OBIA
Object-based w/ Spatio-Contextual	Incorporating geographic models and geostatistical information	ArcGIS feature analyst

2.4 Deep Learning

Before 2006, most machine learning algorithms had shallow-structured architectures such as Hidden Markov Models, Support Vectors Machines, Multi-perceptron Neural Networks with one hidden layer. The shallow structure is effective in solving well-constrained problems but limited in modeling complex problems which require more layers of nonlinear processing such as image processing and human speech processing.

The concept of deep learning originates from artificial neural networks. However, neural network learning algorithms tend to be trapped in poor local optimums due to vanishing gradient problem if a neural network model goes deeper. To overcome the limitation of back-propagation in training deep neural networks, Hinton et al. proposed a greedy layer-wise learning algorithm [70] to pre-train the deep structure. This layer-wise greedy learning algorithm fundamentally changed the way how neural network weight updating mechanism which made it possible to learn a larger and deeper neural network.

2.4.1 Explaining Away

To explain how this layer-wise greedy algorithm works, it is very important to understand the phenomenon of “explaining away” which is the key reason that makes the inference difficult. Fig. 3 shows a simple logistic belief net that contains two independent highly un-correlated hidden causes and one observed fact. The bias of the node indicates that the observed fact is very unlikely to happen unless one of the causes is true. If one of the causes happens, the input is present and the observed fact “house jump” node is on.

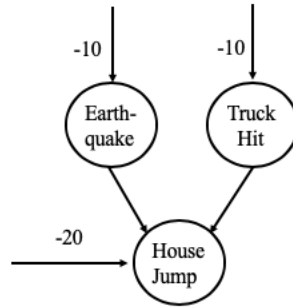


Fig. 3. “Explaining away.”

Assume that the posterior over the hidden variables are:

$$p(1,1) = 0.0001, p(1,0) = 0.4999, p(0,1) = 0.4999, p(0,0) = 0.0001 \quad (1)$$

Though there are four different combinations that caused the house jump, two of them are extremely unlikely to happen at the same time. The other two are equally probable and exclusive to each other. The causes “earthquake” and “truck hits” are two marginally independent causes. When the two causes compete to explain the observed data, the two independent causes become conditionally dependent given the observed data. The probability of both causes happened is so small and the confirmation of one cause reduces the need to invoke another. This is called

explaining away. In this case, the earthquake node “explaining away” the evidence for the truck node. The phenomenon of explaining away makes the posterior distribution of the hidden variables in the networks is hard to infer, which also makes it difficult to learn in a multi-layer network. Fig. 4 shows a simple multilayer neural network, which is an acyclic directed graph and the nodes are random variables.

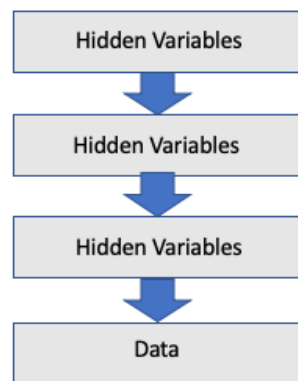


Fig. 4. Multi-layer Neural Network

To learn the weights in this network, for example, W between the first hidden layer and the data, it needs to sample from the posterior distribution in the first hidden layer. However, because of the “explaining away,” the posterior distribution of the first hidden layer is not factorial and not independent. Since there are higher-level hidden variables, those hidden variables in the layer above create a prior on the first hidden layer. These variables in the higher layer are not independent with the prior, which causes correlations of the hidden variable in the first hidden layer. Thus, to learn W , even if they are only used to approximate the posterior, this requires all the weights in the higher layers to be learned. All possible configurations in the higher layers need to be learned to get the prior for the first hidden layer.

2.4.2 Vanishing and Exploding Gradients

Other problems with training deep neural networks are the vanishing and exploding gradients. When training a very deep network, the derivatives of the loss function can sometimes get to very big or very small. It could grow even exponentially, which makes training difficult. We use a simple example to explain this problem. Suppose a deep neural network has l layer with two hidden units per layer. Each layer has weight parameters $w^0, w^1, w^2, \dots, w^l$.

For simplicity, suppose the activation function $g(z)$ is linear: z equals z . And suppose the bias of each layer b^l equals to 0, thus $g(z)$ can be defined as,

$$g(z) = z, b^l = 0, \quad (2)$$

In this case, the output of the network \hat{y} is,

$$\hat{y} = g(w^{[l]}g(w^{[l-1]} \dots g(w^{[2]}g(w^{[1]}g(w^{[0]}x + b^0) + b^1) + b^2) \dots + b^{l-1}) + b^l) \quad (3)$$

$$= w^{[l]}w^{[l-1]} \dots w^{[2]}w^{[1]}w^{[0]}x \quad (4)$$

Suppose the weight matrix of each layer $w^{[l]}$ is equal to identity matrices as,

$$w^{[l]} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} \quad (5)$$

So, the output of the network \hat{y} will be a to the power of $l-1$ times x as,

$$\hat{y} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}^{[l]} \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}^{[l-1]} \dots \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}^{[0]} x = a^{[l-1]}x \quad (6)$$

If the a is larger than 1 and l is very large, the activation function grows exponentially. Conversely, if a is smaller than 1 and when the network goes deeper, the activation of the network will decrease exponentially to 1. The same argument could be applied to the derivatives of the gradients, which will increase or decrease exponentially as a function of the number of layers. This makes the training take a long time for the gradient descent algorithm to learn anything.

2.4.3 Local Minima and Gradients Diffusion

The direct results caused by exploding or vanishing gradient are that training deep networks are extremely difficult. Two major problems are: local minima [71-73] and gradients diffusion/vanishing [74, 75]:

- **Local Minima:** The training of a deep neural network model is a highly non-convex optimization problem. If weights in the model are initially large, training with gradient descent usually lead to poor local minima. The solution points are likely to be saddle points [76, 77]. Such saddle points are surrounded by high error plateaus that can dramatically slow down the learning.

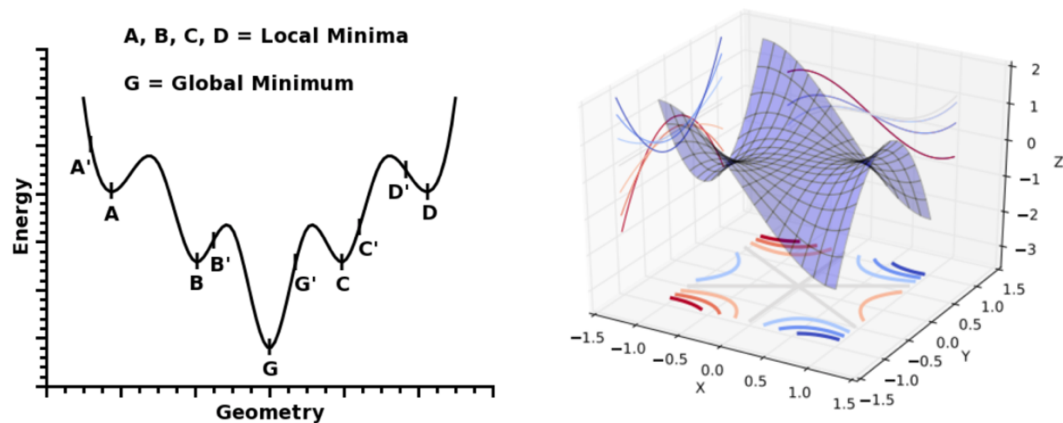


Fig. 5. Local Minima (Left) and Saddle Point (Right) [77]

- **Diffusion/vanishing of gradients:** If weights are initially small and gradients in the early layers are tiny, gradients will rapidly diminish in magnitude as the network layers increases. Therefore, weight changes at the earlier layer are very small and the earlier layers fail to learn. It is infeasible to train networks with many hidden layers.

2.4.4 Complementary Priors

To eliminate the “explaining away” effect in the multilayer network, Hinton et al. [85] proposed a method by adding an extra hidden layer as a “complementary prior” to the first hidden layer. This “complementary prior” will create exactly the opposite correlations of the likelihood between layers. When the likelihood term is multiplied by the priors, it will get a posterior which is factorial with respect to the opposite likelihood.

Specifically, for a joint distribution over observation x and hidden variables y given a likelihood function $p(x|y)$, it defines a complementary prior to those distributions $p(y)$ for this joint distribution $p(x, y)$:

$$p(x, y) = p(x|y)p(y) \quad (7)$$

It leads to the posteriors $p(y|x)$:

$$p(y|x) = \prod_j p(y_j|x) \quad (8)$$

As shown in Fig. 6, it is a multilayer logistic belief network with the complementary of the priors. The units composed the logistic belief nets are stochastic binary units. These units follow the logistic function to generate data. The hidden variables are binary and independent. The non-independence of the posterior distributions is created by the likelihood term coming from the data.

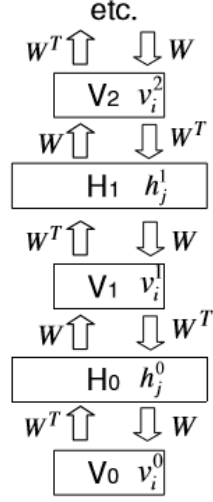


Fig. 6. An Infinite Logistic Belief Net with Tied Weights.

With the complementary priors, it can sample from the true posterior distribution over hidden layers. It starts with a data vector on the visible units, and then uses the transposed weight matrices to infer the factorial distributions over the hidden layer. Because the true posterior can be sampled, derivatives of the log probability of the data can be computed. For the generative weight w_{ij}^{00} from unit j in layer H_0 to unit i in layer V_0 , the maximum likelihood learning rule in logistic net for a single data vector, v^0 is,

$$\frac{\partial \log p(v^0)}{\partial w_{ij}^{00}} = \langle h_j^0 (v_i^0 - \hat{v}_i^0) \rangle \quad (9)$$

In Eq. 9, $\langle \cdot \rangle$ denotes the average over the sampled states and \hat{v}_i^0 is the probability that unit i would be turned on, when the visible vector was re-calculated from the sampled hidden states. To compute the posterior distribution of the second hidden layer V_1 , since we already sampled the first hidden layer, v_i^1 is a sample from a Bernoulli random variable with probability \hat{v}_i^0 ,

$$\frac{\partial \log p(v^0)}{\partial w_{ij}^{00}} = \langle h_j^0 (v_i^0 - v_i^1) \rangle \quad (10)$$

To sum the derivatives of the generative weights between all pairs of layers, we get the full derivative for a generative weight as,

$$\frac{\partial \log p(v^0)}{\partial w_{ij}^{00}} = \langle h_j^0 (v_i^0 - v_i^1) \rangle + \langle v_i^1 (h_j^0 - h_j^1) \rangle + \langle h_j^1 (v_i^1 - v_i^2) \rangle + \dots \quad (11)$$

2.4.5 Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) is a generative stochastic neural network model proposed by Smolensky et al. [78] in 1986, improved by Freund and Haussler in 1992 [79] and Hinton in 2002 [80]. A single RBM is a two-layer bipartite undirected network consists of two layers: visible layer and hidden layer. It uses symmetrically weighted connections between the visible layer and the hidden layer. There are no connections between the nodes in the same layer (Fig. 7).

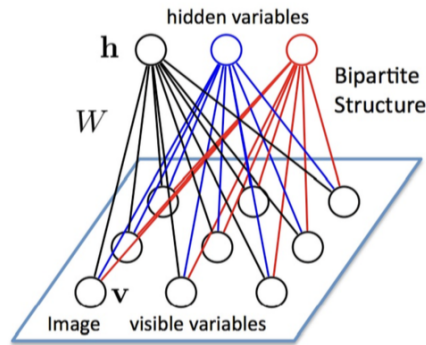


Fig. 7. Restricted Boltzmann Machine with 12 Visible Units and 3 Hidden Units.

Each visible node takes a low-level feature. For example, if the input is an image, each visible node will receive one pixel-value for each pixel in an image whereas the hidden units represent feature detectors. RBMs are undirected and each RBM has a single weight matrix W . If the bias units take a for the visible layer v and b for the hidden layer h , the energy function of a joint configuration v, h is defined as,

$$E(v, h) = - \sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j \quad (12)$$

The probability distributions over the joint configuration are defined in terms of the energy function as,

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (13)$$

Where Z is a normalizer, which is the sum over all possible configuration $Z = \sum_{v, h} e^{-E(v, h)}$. The probability assigned by the network to a visible vector is $p(v) = \frac{1}{Z} e^{-E(v, h)}$. The network assigns a probability to every possible input image/signal via the $p(v, h)$ function. The probability of a training image can be raised by adjusting the weights and biases to lower the energy and increase the energy of similar reconstructed images. We would like maximize the log-likelihood function of the observed data/input $p(v)$ as,

$$p(v) = \frac{1}{Z} \sum_h \exp [v^T W h + a^T h + b^T v] \quad (14)$$

$$L(\theta) = \frac{1}{N} \sum_{n=1}^N \log P_{\theta}(v^n) \quad (15)$$

To use stochastic gradient descent [81, 82] to maximize $L(\theta)$, we first compute the derivative of $L(\theta)$ to W as,

$$\frac{\partial L(\theta)}{\partial W_{ij}} = E_{P_{\text{data}}} [v_i h_j] - E_{P_{\theta}} [v_i h_j] \quad (16)$$

$E_{P_{\text{data}}} [v_i h_j]$ is easy to compute (average of all $v_i h_j$), but the second term $E_{P_{\theta}} [v_i h_j]$ has $2^{|v|+|h|}$ combinations, it is usually unsolvable. To solve it, Hinton proposed the Contrastive Divergence algorithm [80, 83]. This algorithm uses the input data vector v to get h , and then reconstructs input vector v_1 . Then v_1 is used to generate the new h_1 as shown in Fig 10. The reconstruction of v_1 and h_1 is one-time sample of $p(v, h)$. By sampling v and h for multiple times, the result set could be considered as a good estimation of $p(v, h)$, then the second term is estimated.

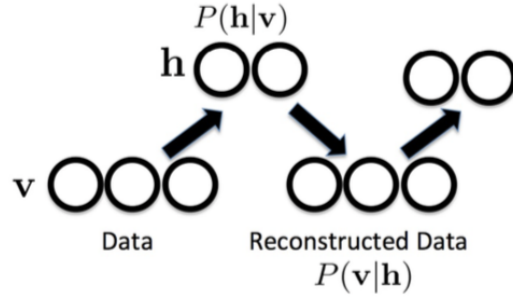


Fig. 8. Contrastive Divergence

Because of the specific structure of RBMs, the visible and hidden units are conditionally independent given one-another,

$$p(h|v) = \prod_j p(h_j|v) \quad (17)$$

$$p(v|h) = \prod_i p(v_i|h) \quad (18)$$

For the binary data input, given by the partite function, the probability of a visible unit set to 1 for a hidden vector h is

$$p(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}) \quad (19)$$

A binary hidden unit is set to 1 with the following probability for an input vector v as,

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}) \quad (20)$$

σ is the sigmoid function. Once the binary states have been chosen for the hidden units, v_i is set to 1 with probability $p(v_i = 1|h)$, the states of hidden units are then updated once more. The change in weights is given by

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{\text{data}} - < v_i h_j >_{\text{reconstructed}}) \quad (21)$$

To summarize, the weight training algorithm can be summarized as:

1. Get a sample data vector, random initialize W , set the visible layer as the sample data vector.
2. Update h_j using Eq. 14, then to every connection $v_i h_j$ compute $P_{\text{data}}(v_i h_j) = v_i h_j$.
3. According to the result of h and use Eq. 13 to reconstruct v_1 , then use v_1 and Eq. 14 to reconstruct h_1 , compute $P_{\text{model}}(v_1 h_1) = v_1 h_1$.
4. Update w_{ij} using $\Delta w_{ij} = \varepsilon(< v_i h_j >_{\text{data}} - < v_i h_j >_{\text{model}})$.
5. Get next sample vector, repeat steps 1-4.
6. Iterate steps 1-5 for K times.

After learning the first layer of binary features, the first layer of feature detectors now become the visible units for the learning of the next RBM. This layer by layer learning can be repeated multiple times as needed. For continuous data, the first-level RBM remains binary, but

the visible units are replaced by a continuous stochastic unit by adding a zero-mean Gaussian noise to the input. The energy function becomes as,

$$E(v, h) = \sum_{i \in \text{vis}} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j \in \text{hid}} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \quad (22)$$

where σ_i is the standard deviation of the Gaussian noise for visible unit i .

Under the modified energy function, the conditional probability and the update rule for each visible and hidden neuron become

$$p(v_i = v | h) = \mathcal{N}(v | b_i + \sum_j h_j w_{ij}, \sigma_i^2) \quad (23)$$

$$p(h_j = 1 | v) = \text{sigmoid}(c_j + \sum_i w_{ij} \frac{v_i}{\sigma_i}) \quad (24)$$

This procedure will learn a stack of RBMs with one layer at a time. The learned feature activations of one RBM are used as the data for training the next RBM in stack. As shown in Fig. 9, the network was divided into four stacked of RBMs and the output of the lower level RBM is the input of the next level RBM.

In summary, the training procedure can be summarized as:

1. Train one layer at a time, from first to last, with unsupervised criterion.
2. Fix the parameters of previous hidden layers.
3. Previous layers viewed as feature extraction.

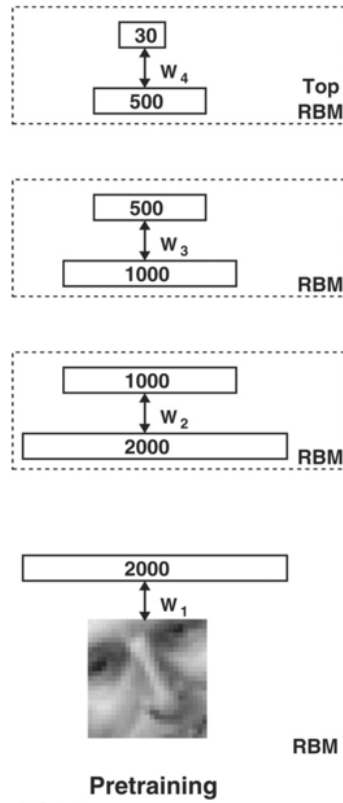


Fig. 9. Pre-training Procedure and Stacked RBM [84]

2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) is inspired from Hubel and Wiesel's early work on the cat's visual cortex [85]. It is a multi-layer feed forward artificial neural network. The origin of CNN dates back to the 1970s, and in 1998 Yann LeCun et. al [86] established the modern model of CNN. A typical convolutional neural network includes input layer, convolutional layer, pooling layers, and output layer. It introduces three basics but very important ideas into artificial neural networks:

- 1) Local Receptive Fields/ Sparse Connectivity (Input Layer).
- 2) Feature Map, Shared Weights and Bias (Convolutional Layer).

3) Pooling (Sub-sampling Layer).

2.5.1 Local Receptive Fields (Sparse Connectivity)

In CNN, the input layer is an $n \times n$ square of neurons, values of the neurons correspond to the input image with $n \times n$ pixels. Instead of connecting every input pixel to every hidden neuron as in the traditional fully connected neural network, in CNN network, a small region of input neurons is connected to one hidden neuron in the first hidden layer. In Fig. 10, a 5×5 region is corresponding to 25 input pixels and the 25 input neurons are only connected to 1 neuron in the first hidden layer. The local receptive field concept greatly reduced the dimension of input and the number of the hidden neurons in the first hidden layer.

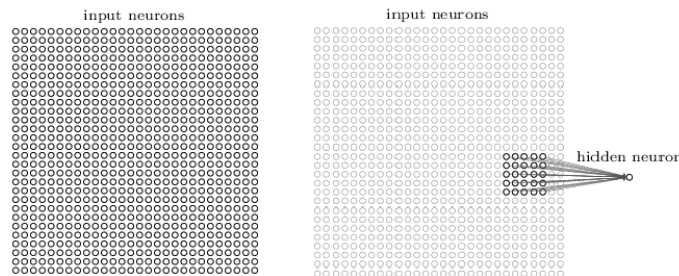


Fig. 10. Local Receptive Field.

2.5.2 Feature Maps, Shared Weight and Biases

Each hidden neuron has a bias and $o \times o$ weights connected to the local receptive field (suppose the local receptive field is in size $o \times o$). All hidden neurons in hidden layer share the same weights and bias. Use sigmoid function as the activation function, suppose b is the shared

bias, $w_{l,m}$ is the $o \times o$ array of shared weights, $a_{j+l,k+m}$ is the input, the output and activation function of the j, k_{th} hidden layer neuron are

$$\text{Sigmoid}(b + \sum_{l=0}^{o-1} \sum_{m=0}^{o-1} w_{l,m} a_{j+l,k+m}) \quad (25)$$

Suppose the hidden unit is connected to some particular area in the image. And this particular shared weight $w_{l,m}$ and the bias b in this connection will cause the sigmoid function activated. In other words, the hidden neuron is activated. The local receptive fields which are associated with this hidden neuron might follow some particular patterns, say it is a vertical edge. By keep moving the same local receptive field over the whole input image, all hidden neurons which got activated indicate the same pattern in the input images.

The mapping from the input layer to the hidden layer is called a feature map. The shared weights and bias define a convolutional kernel. By moving one kernel over the image results in a feature map. And each kernel will generate a feature map. As Fig. 11 shows, three feature maps are generated through convolution.

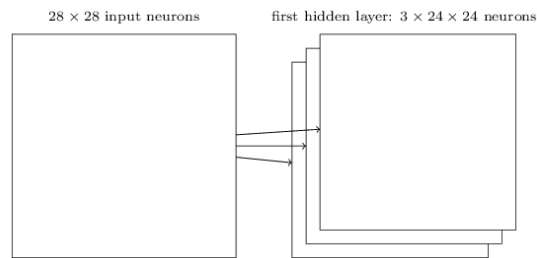


Fig. 11. Feature Maps

2.5.3 Pooling

Convolutional layers are usually followed by a pooling layer to reduce the dimension of the feature maps. The most common pooling is max-pooling, in which only the maximum value in a sub-region of a feature map is kept as shown in Fig. 12. If a CNN has more than one feature maps, the pooling process is usually applied to each of the feature maps as shown in Fig 13.

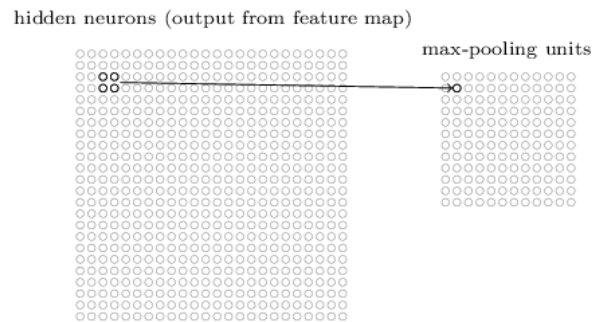


Fig. 12. Max-Pooling

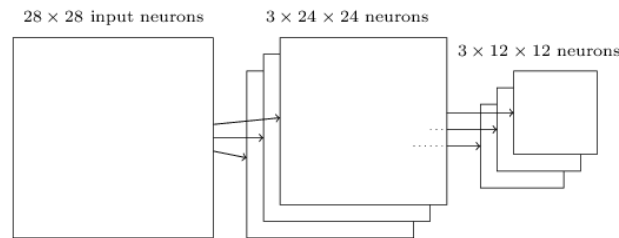


Fig. 13. Input Layers, Convolutional Layers and Pooling Layer.

Use the example in Fig. 13, suppose there are 10 possible outputs, the output layer will be fully connected to the pooling layer and all neurons of the pooling layer will be connected to all neurons in the output layer as shown in Fig. 14.

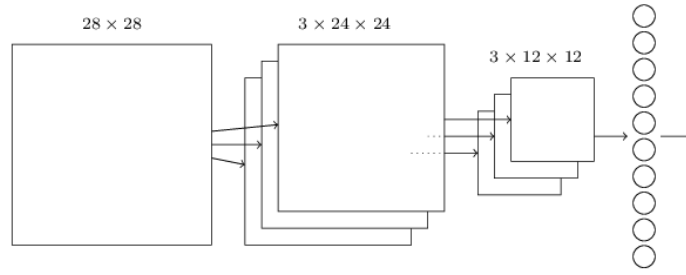


Fig. 14. CNN Model Structure.

2.5.4 AlexNet

Krizhevsky et al. trained a very large and deep CNN, named AlexNet, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [87] and won the competition. It achieved top-1 and top-5 error rate of 37.5% and 16.4%, respectively. These results outperformed the second-best results by a substantial margin of 10%. AlexNet has become a milestone and backbone of many later deep CNNs models such as deep residual network [88].

The architecture of AlexNet is shown in Fig. 15. The network has eight layers: five convolutional layers followed by max pooling layers and three fully connected layers with a final 1000-way SoftMax layer for output.

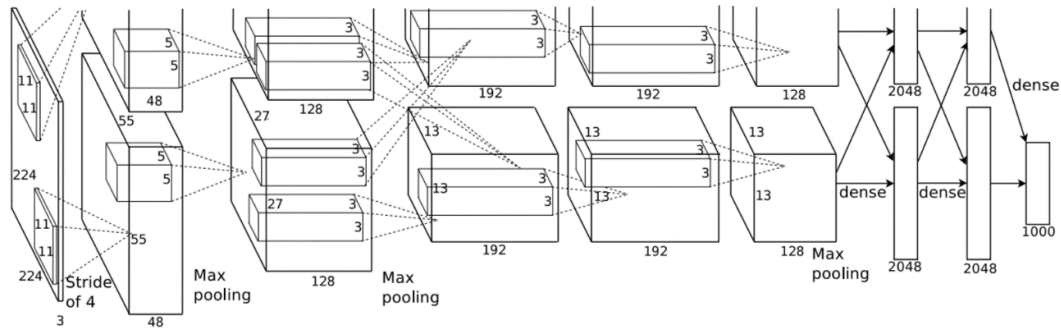


Fig. 15. AlexNet Overall Structure

The input layer contains $3 \times 224 \times 224$ neurons, representing the RGB value for a 224×224 image. The original image is firstly scaled to make the short side has a length of 256, and then is cropped out at the center of a 256×256 area, which is subsequently randomly cut as a 224×224 sub-image as input. The first convolutional layer has a local receptive field of 11×11 with a stride of 4. There are 96 feature maps generated. These feature maps are split into two groups and each GPU holds one group. Max pooling in a 3×3 region is applied to each feature map. The second convolutional layer uses a 5×5 local receptive field and there are 256 feature maps in total. Followed by the second convolutional layer is the second max-pooling layer. The third, fourth, and fifth hidden layers are all convolutional layers with a local receptive field of 3×3 . The third and fourth convolutional layers have 384 feature maps, and the last one has 256 feature maps. The sixth and seventh hidden layers are fully-connected layers with 4096 neurons in each layer. The output layer is a 1000-unit SoftMax layer. Overall, AlexNet has about 660K units, 61M parameters and over 600M connections. With the huge size of parameters, it is easy for the network to remember the data that results in overfitting. AlexNet uses some optimizing techniques to avoid overfitting as described below.

2.5.5 Activation Function - ReLU

AlexNet uses the rectified linear unit (ReLU) as activation function [89]. ReLU outputs input directly if the input is positive. Otherwise, it will output zero. Prior to the ReLU activation function, the hyperbolic tangent function as Eq. 27 and sigmoid function as Eq. 26 were generally accepted. Unlike sigmoid or tangent activation functions, ReLU does not saturate to -1, 0 or 1.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (26)$$

$$f(x) = \tanh(x) \quad (27)$$

$$f(x) = \sum_{i=1}^{\infty} \sigma(x - i + 0.5) \approx \log(1 + e^x) \quad (28)$$

The $\log(1 + e^x)$ can be approximated by max function $\text{Max}(0, x)$ as shown in Fig.16.

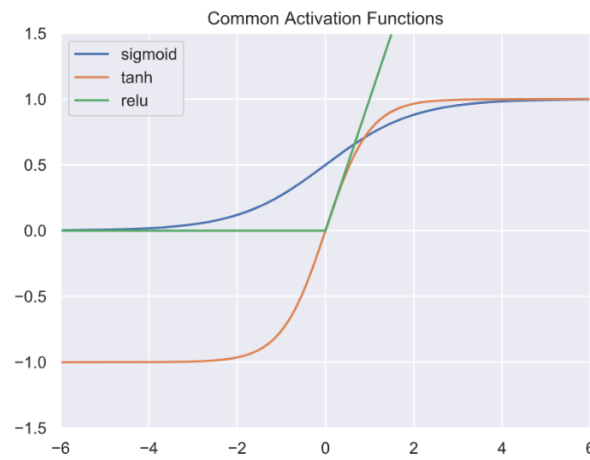


Fig. 16. ReLU, Sigmoid and Hyperbolic Tangent Activation Functions

One reason that ReLU performs better than sigmoid or tangent activation functions is that derivative of ReLU does not vanish. As shown in Fig. 17, hyperbolic tangent and sigmoid functions have the derivative vanishing problem.

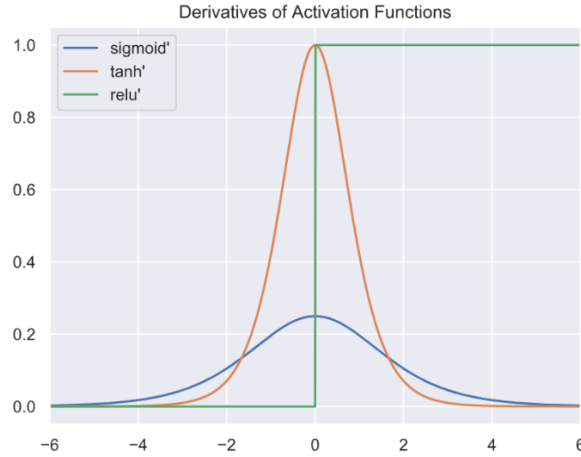


Fig. 17. Vanishing Derivatives of Hyperbolic Tangent and Sigmoid Activation Functions

2.5.6 Local Response Normalization

AlexNet proposed the local normalization scheme to improve generalization. Local response normalization was applied after ReLU for the first and second layers. The local response normalization is shown in Eq. 29, and it improved the result by 1%.

$$b_{x,y}^i = a_{x,y}^i / (k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^j)^2)^\beta \quad (29)$$

2.5.7 Dropout

Dropout is a regularization technique that holds many benefits for deep learning. Dropout works by removing certain randomly selected neurons in each layer during training. In Fig. 18, from top to bottom, the neural network contains an input layer, a dropout layer and an output layer. The dropout layers have removed several of the neurons. The dropout neurons do not contribute to the feed forward and back propagation computations during training. By applying dropout:

- It reduces the complexity of co-adaptation of neurons.

- It forced the network to learn more robust features.

Dropout was used in the first two layers in AlexNet and it substantially reduced the over-fitting problem. One shortcoming of Dropout is that it roughly doubles the number of training iterations required to converge.

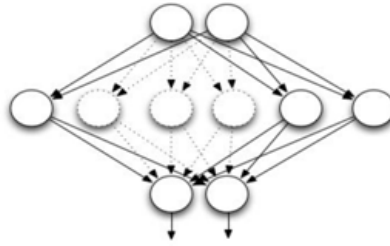


Fig. 18. Dropout

2.6 Transfer Learning

In the computer vision domain, certain low-level features such as edges, shapes, or curves can be shared across different tasks. If we have insufficient training data in one task domain, we may utilize knowledge learned in a related task to mitigate the challenge [30, 90]. The definition of transfer learning is described as follows.

A domain D consists of two components: feature space χ and marginal probability $P(X)$, where X is a sample data point, x_i represents a specific vector. Thus, the domain could be mathematically represented as

$$D = \{\chi, P(X)\} \quad \text{where } X = \{x_1, \dots, x_n\}, x_i \in \chi \quad (30)$$

A task T is defined as a two-element tuple of label space, with a label space ψ and an objective function η . The objective function is denoted as conditional probability distribution of

$P(Y|X)$, which is learned from training data pairs (x_i, y_i) . Thus, for a given domain D , task T could be defined as

$$T = \{\psi, P(Y|X)\} = \{\psi, \eta\} \text{ where } Y = \{y_1, \dots, y_n\}, y_i \in \psi \quad (31)$$

Given a source domain D_s and a corresponding source task T_s , the objective of transfer learning is to learn the target conditional probability distribution $P(Y_T|X_T)$ in target domain D_s with information learned from D_s and T_s .

CHAPTER 3

HYPERSPPECTRAL IMAGE CLASSIFICATION BY CNN

In this section, we present a 4-layer deep convolutional neural network (CNN) model for soil detection by using the combination of 80 synthetic hyperspectral bands and its original 8 multispectral bands which are collected by the WorldView-2 satellite. Our experimental result shows that by using the combined 80 synthetic hyperspectral bands and the original 8 multispectral bands, the area under the curve (AUC) scores of our CNN model for soil detection on the three testing images has been improved by 7.42% on average from 76.26% to 83.48%, as compared to the result by using the 8 multispectral bands alone. We also applied the CNN model onto a set of high-resolution data which is created by pan-sharpening the original multispectral bands and its synthetic hyperspectral bands, which quadrupled the spatial resolution of the combined synthetic hyperspectral bands. With the increased spatial resolution of the combined synthetic hyperspectral bands, the average AUC scores of our CNN model was furtherly improved by 10.02% from 81.44% to 91.47%. This significant improvement indicates that by using the pan-sharpened synthetic hyperspectral bands, the performance of CNN model for soil detection has been greatly improved.

3.1 Introduction

The U.S. has about 10,000 kilometers of international borders with Mexico and Canada. Typical border surveillance tasks include trail detection, illegal tunnel detection, and humanitarian missions - rescuing people lost in remote locations and exposed to harsh environmental conditions [91-93]. According to [94, 95] in 2014, a total of 101 cross-border tunnels were discovered. All tunnel digging activities started and ended inside warehouses as shown in Fig. 19, prohibiting

direct observation. However, since excavated soil needs to be disposed from the tunnel entrance and exit, it is possible to use exposed soil as an indirect indicator of tunnel activities. Conventional border monitoring approaches use airborne sensors [96-98] that have close to 1m spatial resolution. With recent advances in satellite images such as Worldview-2 (WV2) and Worldview-3 (WV-3) that have 0.31m resolution in pan band and 1.24m in visible and near infrared (VNIR) bands, it becomes possible to use satellite images for border monitoring and surveillance.



Fig. 19. Illegal Tunnels near U.S. and Mexico Border

3.2 Motivation

Remote sensing images play a critical role in earth's surface monitoring. Multispectral images, which usually refers to images with 3 to 10 bands, have been widely and regularly used in the remote sensing area since the 1970s. With the fast advances of hyperspectral sensors, the hyperspectral imagery, also known as imaging spectrometry, which contains many more bands with much narrower bandwidth (10-20 nm) than multispectral bands are expected to be more effective in target identification [3-5, 99], land cover classification [100, 101], anomalous

materials, and objects detection [102, 103]. Nonetheless, hyperspectral imagery data might not always be available. As an alternative, spectral dimensionality expanding methods [28, 104-106] can be used to create synthetic hyperspectral bands by applying it to the original multispectral bands. The synthetic hyperspectral bands might not have the same physical meanings as the real hyperspectral bands do; however, it is generally expected and accepted that the synthetic hyperspectral bands, which hold the correlations with the spatial characteristics of the objects along with the spectral information contained in the original images, are highly likely to achieve better object detection and classification results than the regular multispectral bands.

3.3 Methodology

3.3.1 Generating Hyperspectral Bands (EMAP)

The satellite images used in this project were captured by the Worldview-2 (WV-2) satellite, which contains eight bands of multispectral channels. The first step is to use both spectral and spatial information of the eight channels to generate a newly expanded image with high dimensional synthetic bands. We used the Extended Multi-Attribute Profile (EMAP) [104] to generate eighty synthetic hyperspectral bands as shown in Fig. 20. EMAP is an extended idea of attribute profile (AP), a method that has recently been presented as an efficient tool for spectral-spatial analysis of remote sensing images. APs provide a multi-level characterization of an image obtained by applying a sequence of morphological attribute filters to model different kinds of structural information on a single-band (or grayscale) image. These attribute filters can be morphological operators (so-called features) such as thinning or thickening operators that process an image by merging its connected pixels. APs using different types of attribute features on different threshold levels can be stacked together, generating EMAPs.

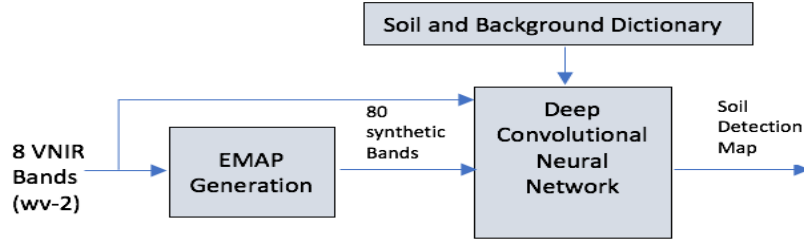


Fig. 20. EMAP Synthetic Hyperspectral Bands Generation and Soil Detection Framework

3.3.2 CNN structure

The structure of the CNN model we used in this study is shown in Fig. 21; the CNN Model has four convolutional layers with various filter sizes and one fully connected layer with 100 hidden units. After each convolutional layer, the Rectified Linear Unit (ReLU) is used as the activation function, the last fully connected layer uses the SoftMax function for classification. We add dropout layer for each convolutional layer with a dropout rate of 0.1 to mitigate overfitting.

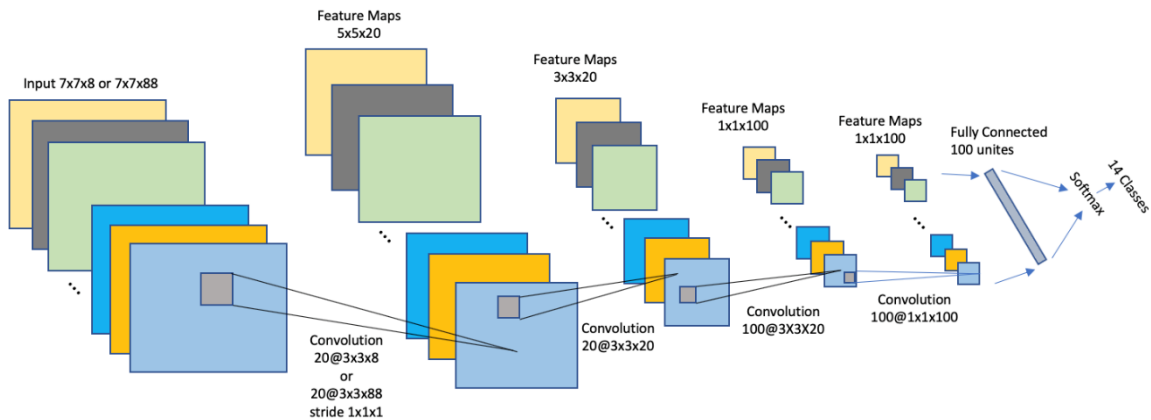


Fig. 21. The CNN Model Structure in Soil Detection.

3.3.3 Data Preparation

Soil class samples are collected in the red oval areas as shown in Fig. 22. Other than the soil class, there are 13 other classes of land cover types including cars, white trucks, black trucks, gray buildings, strip-shape-buildings, roads, runways, checkerboard shaped land, land near runways, land near soil, general trees, trees near soil and parking lots. All classes of samples are collected in the areas as shown in Fig. 23. The masks of these 14 classes are created manually using Matlab by visual inspection.



Fig. 22. Soil Samples in WV-2.

Besides the original 8-channel multispectral bands and the 80-channel synthetic hyperspectral bands in the original resolution, we applied a pan-sharpening technique to the 8-channel multispectral data and the 80-channel synthetic hyperspectral data in the original resolution (low-resolution). We obtained another training dataset in quadrupled resolution (high-resolution) as well. The pan-sharpening of multispectral images was carried out by Dao et al. [107] by using the Gram-Schmidt Adaptive (GSA) technique. To train the CNN model, we extracted patches for different classes using the masks. Patches are extracted with patch sizes of $7 \times 7 \times 8$ and $7 \times 7 \times 88$ from the 8-channel original multispectral data and the combined synthetic 88-channel hyperspectral data

in both low-resolution and high-resolution cases. Details of the extracted training patches for different classes are shown in Table 3.

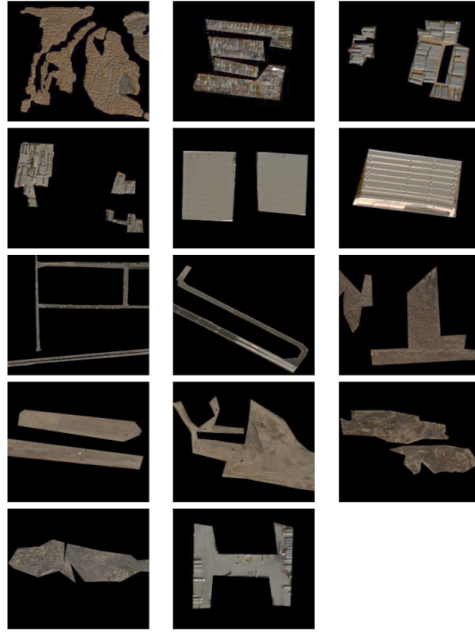


Fig. 23. All 14 Classes of Training Samples in Soil Detection.

TABLE 3. Description of 14 Classes in Soil Detection

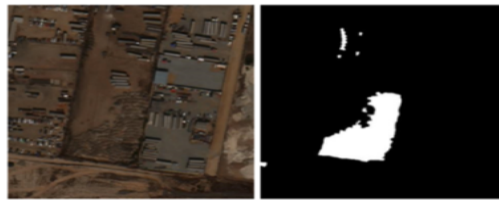
Class Name	Class ID	Number of Patches 7x7x8 & Patches 7x7x88 in high resolution	Number of Patches 7x7x8 & Patches 7x7x88 in low resolution
Soil	1	89,520	5,537
Car	2	68,118	4,286
Truck-White	3	70,599	6,341
Truck-Black	4	78,003	4,911
Building-Gray	5	461,257	28,769
Building-Strip	6	71,051	4,474
Road	7	635,539	39,404
Runway	8	588,135	36,826
Land-Checkerboard	9	850,743	53,226

Land-Runway	10	1,393,342	87,137
Land-Near-Soil	11	213,489	18,040
Tree-Near-Soil	12	492,081	30,038
Tree-General	13	187,812	17,169
Parking-Lot	14	141,556	13,343

By manually screening through 12 sets of WV-2 Pan-VNIR images, three images dated 03/19/2010, 10/11/2010 and 12/02/2010 were selected for testing. The contour of the masks of ground truth of soil area in the three images were developed manually as shown in Fig. 24.



(a) Testing Image Captured on 03/19/2010



(b) Testing Image Captured on 10/11/2010



(c) Testing Image Captured on 12/02/2010

Fig. 24. Testing Images and Ground Truth Masks in Soil Detection.

3.3.4 Imbalance Learning

We collected 14 classes in total for training, however, the goal of this project is to detect the soil class. Accurately identifying the other class types is not our major concern. Furthermore, the training dataset is highly unbalanced as shown in Table 3, of which some classes have significant more samples than others. Traditional imbalance learning methods include down-sampling majority classes or up-sampling minority classes to balance the data. In our study, to solve this imbalance learning problem, we randomly sample all classes to the smallest patch number among all of the classes (68,118 in high resolution and 4,286 in low resolution) before training. In the following training phase, we train the CNN model to classify all the 14 classes. In testing, we convert the 14-classes problem to 2-classes problem by using the following conversion as,

$$P'_{\text{Soil}} = \frac{P_{\text{Soil}}}{P_{\text{Soil}} + \max(P_{\text{Non-Soil}})} \quad (32)$$

$$P'_{\text{Non-Soil}} = \frac{\max(P_{\text{Non-Soil}})}{P_{\text{Soil}} + \max(P_{\text{Non-Soil}})} \quad (33)$$

where P_{Soil} and $P_{\text{Non-Soil}}$ are the probabilities of the soil and all non-soil classes predicted by the CNN model, while P'_{Soil} and $P'_{\text{Non-Soil}}$ are the converted 2-classes probabilities of soil class and non-soil class.

3.3.5 Post-processing

After we obtained soil probability map for a testing image, we applied the morphological closing operation filter to it to enforce group sparsity across neighbor pixels. This process can

reduce false positive and connect isolated soil regions. A low-pass filter was then applied to smooth out the soil map. We computed ROC curves and AUC scores as the performance metrics for comparing different methods.

3.4 Model Training and Experimental results

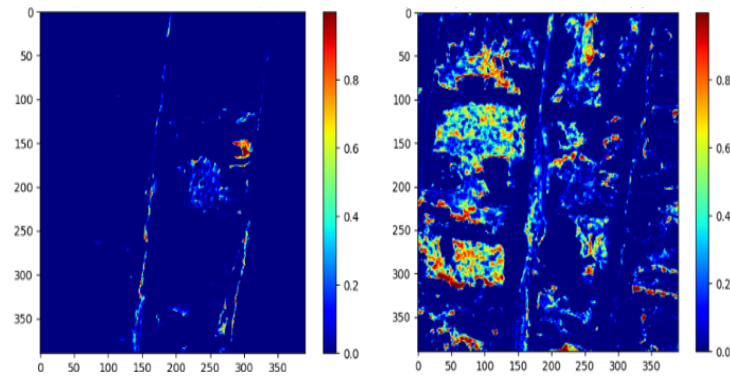
In this stage, we train the corresponding CNN models in accordance with the combined 88-channel hyperspectral and 8-channel multispectral data in both high-resolution and low-resolution scenario. We train the CNNs with a batch size of 64, the training is stopped when the CNNs are starting get converged at around 155 epochs of training. Then we save the corresponding models for the different testing scenario.

After we got the trained CNN models for soil detection task for both the high-resolution and low-resolution data, we apply the trained model on the three testing images to detect soil. Our results show that:

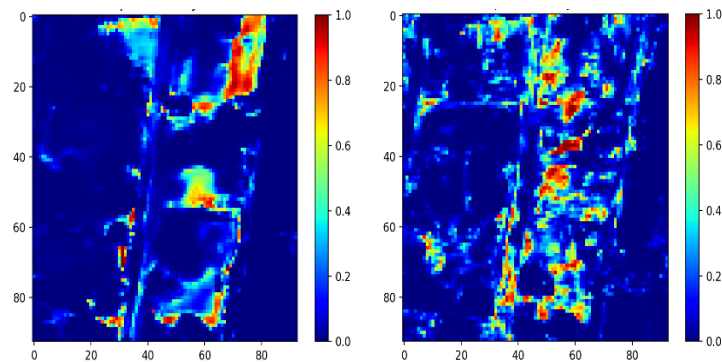
- 1) In all the three testing images, by using the combined synthetic 88-channel hyperspectral data, the AUC scores for both high-resolution and low-resolution are all improved. The improvement is ranged from 0.61% to 28.74% in different testing scenarios. For the three different images, the overall AUC has been improved, on average, 9.02% for high-resolution data, and 7.42% for low-resolution data.
- 2) From our testing results, the combined 88-channel hyperspectral data achieved the largest boost for soil detection on the testing image dated March 9th, 2010 as compared to the 8-channel multispectral data. The AUC is significantly improved by 28.74% for high-resolution data and 9.72% for low-resolution data in this testing image. The detailed results

by using the synthetic 88-channel hyperspectral data and the 8-channel multispectral data in the three testing images are shown in the subsequent subsections.

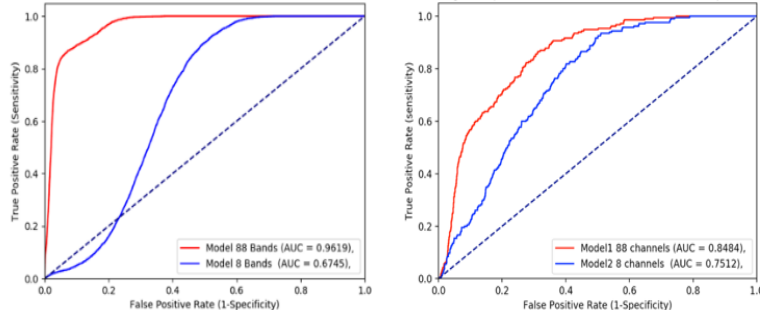
Fig. 25 shows the testing results obtained on the testing image taken on 03/19/2010. For this testing area, the AUC scores of our CNN soil detection model using 88-channel hyperspectral data and 8-channel multispectral data are 96.19% and 67.45% in high resolution, and 84.84% and 75.12% in low resolution, respectively. Using the synthetic hyperspectral data, it significantly improved the soil detection performances.



(a) High-resolution Results for Testing Image Dated 3/19/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.



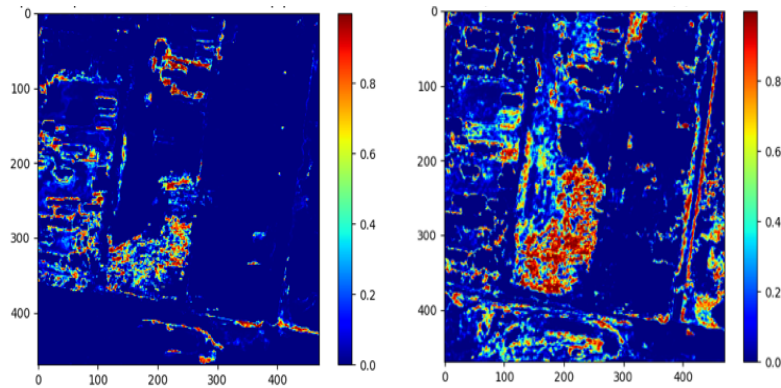
(b) Low-resolution Results for Testing Image Dated 3/19/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.



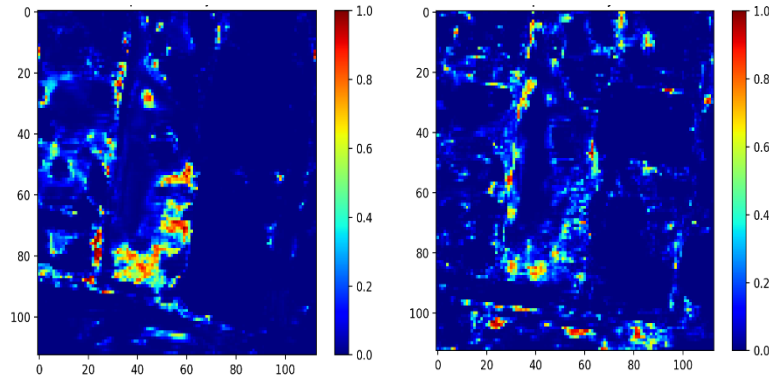
(c) Left: ROC Curves for the Results in (a). Right: ROC Curves for the Results in (b).

Fig. 25. Soil Detection Results for the Testing Image Dated 3/19/2010.

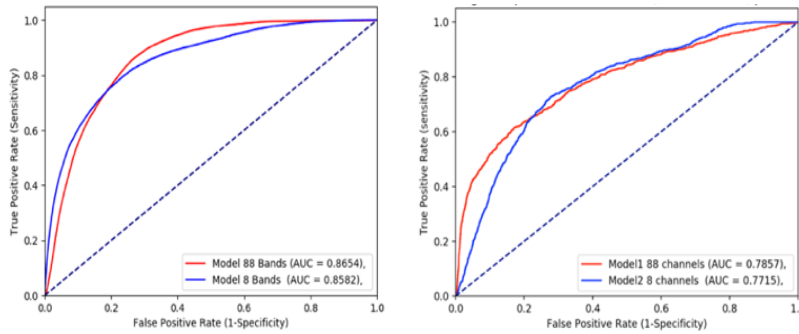
Fig. 26 shows the testing results obtained on the testing image taken on 10/11/2010. For this testing area, the AUC scores of our CNN soil detection model using 88-channel hyperspectral data and 8-channel multispectral data are 86.54% and 85.82% in high resolution, and 78.57% and 77.15% in low resolution, respectively. All results are comparable for this testing image.



(a) High-resolution Results for Testing Image Dated 10/11/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.



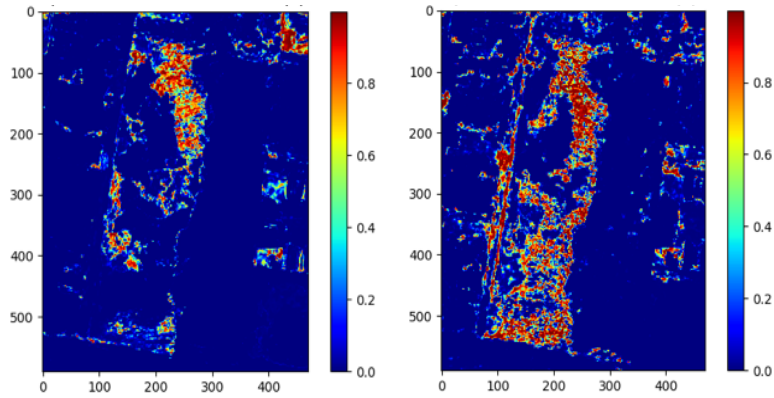
(b) Low-resolution Results for Testing Image Dated 10/11/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.



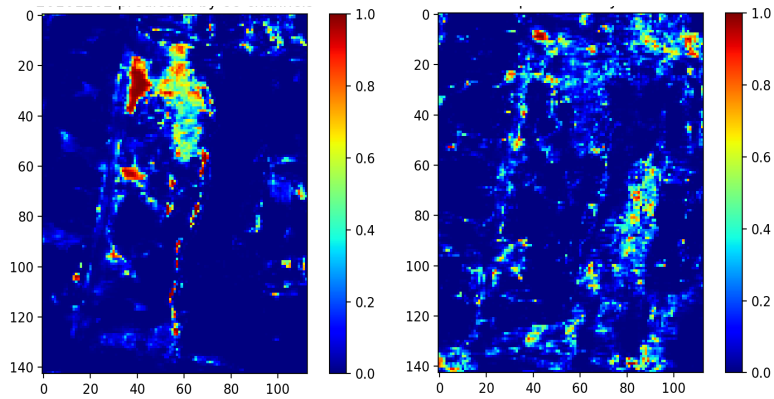
(c) Left: ROC Curves for the Results in (a). Right: ROC Curves for the Results in (b)

Fig. 26. Soil Detection Results for the Testing Image Dated 10/11/2010.

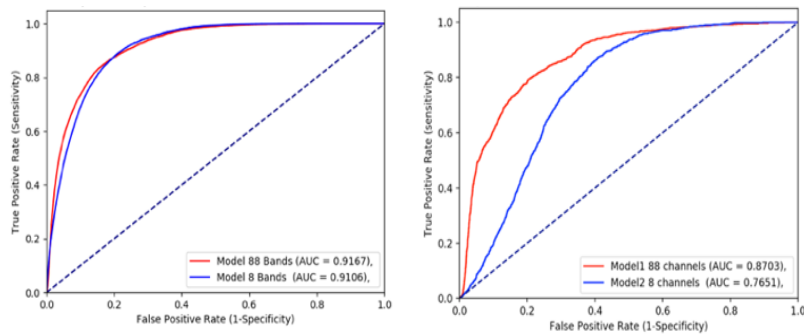
Fig. 27 shows the results obtained on the testing image taken on 12/02/2010. For this testing area, the AUC scores of our CNN soil detection model using 88-channel hyperspectral data and 8-channel multispectral data are 91.67% and 91.06% in high resolution, and 87.03% and 75.51% in low resolution, respectively. The results for high-resolution data are comparable. The soil detection result is significantly improved by using the combined 88-channel hyperspectral data over the 8-channel multispectral data for low-resolution data.



(a) High-resolution Results for Testing Image Dated 12/2/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.



(b) Low-resolution Results for Testing Image Dated 12/2/2010. Left: Soil Prediction Map by 88-band Data. Right: Soil Prediction Map by 8-band Data.

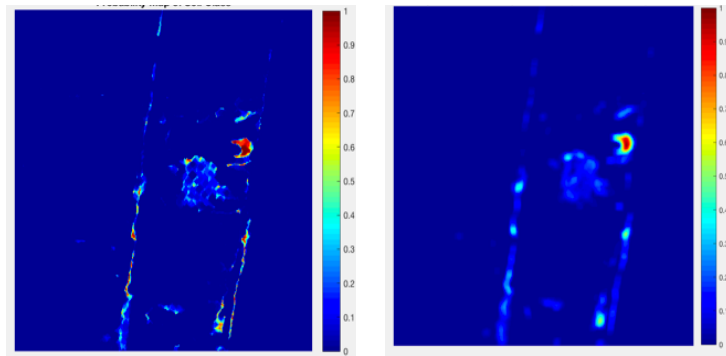


(c) Left: ROC Curves for the Results in (a). Right: ROC Curves for the Results in (b)

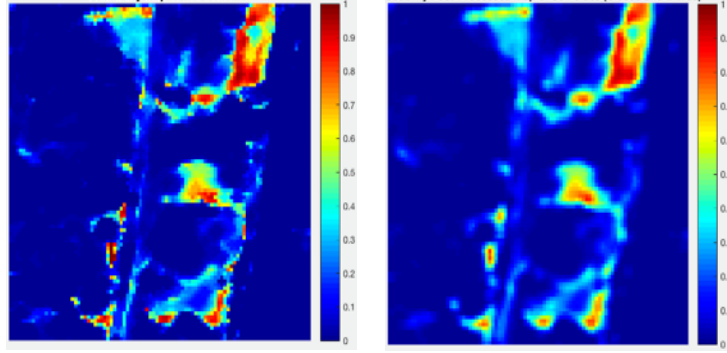
Fig. 27. Soil Detection Results for the Testing Image Dated 12/2/2010.

The soil detection results in the previous subsection demonstrated the benefits of using the 88-channel hyperspectral data, the soil detection performance of our CNN model has been significantly improved. In this subsection, we further applied post-processing with morphological filters to the results of 88-channels hyperspectral data. Our experiment results show that by post-processing, the results can be improved further.

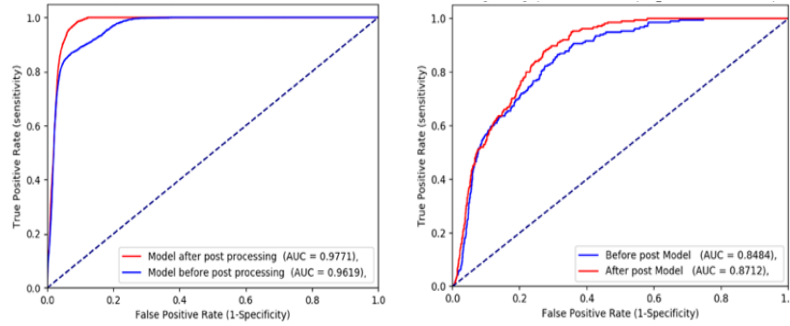
Fig. 28(a) shows that the detection results of soil prediction map before and after the post-processing using 88-channel hyperspectral data in high resolution, and Fig.28(b) shows the detection results of soil prediction map before and after the post-processing using 88-channel hyperspectral in low-resolution. Fig. 28(c) shows the ROC curves before and after the application of the post-processing in both the high resolution and low-resolution cases. The AUC scores were improved by 1.52% in high-resolution and 2.28% in low-resolution, respectively.



(a) Left: Probability Map of Soil before Post-processing 88-band in High Resolution. Right: Probability Map of Soil after Post-processing 88-band in High Resolution.



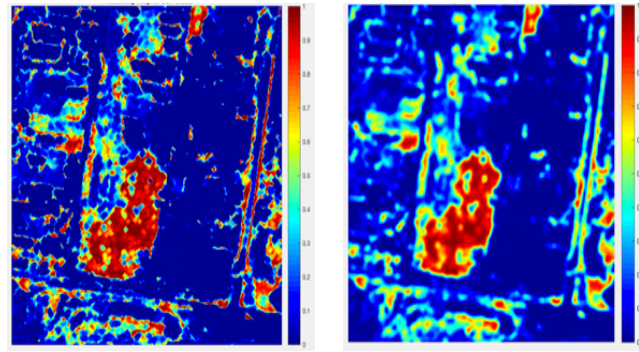
(b) Left: Probability Map of Soil before Post-processing 88-band in Low Resolution. Right: Probability Map of Soil after Post-processing 88-band in Low Resolution.



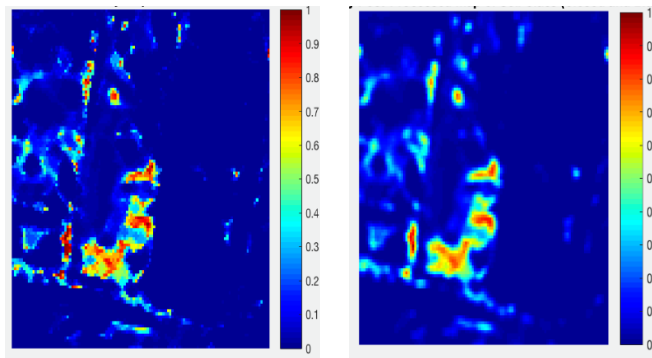
(c) Left: ROC Curves before and after Post-processing in High Resolution. Right: ROC curves before and after Post-processing in Low Resolution.

Fig. 28. Post-processing Results for Testing Image Dated 3/19/2010.

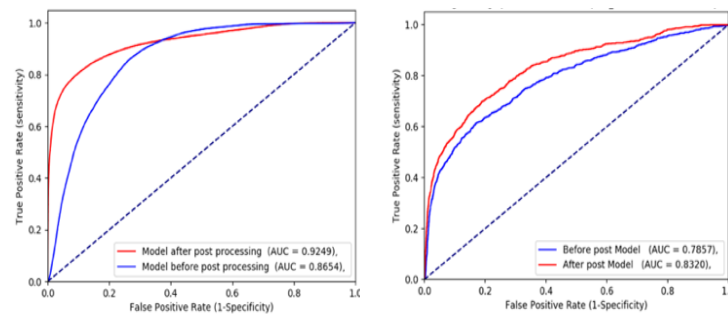
Fig. 29(a) shows detection results before and after the post processing step on the soil prediction map produced by using 88-channel hyperspectral data in high resolution, and Fig. 29(b) shows the corresponding results for the low-resolution case. Figs. 29(c) shows the ROC curves before and after the application of the post-processing step in both the high-resolution and low-resolution cases. The AUC scores were improved by 5.95% in high-resolution and 4.63% in low-resolution, respectively.



(a) Left: Probability Map of Soil before Post-processing. Right: Probability Map of Soil after Post-processing in High Resolution.



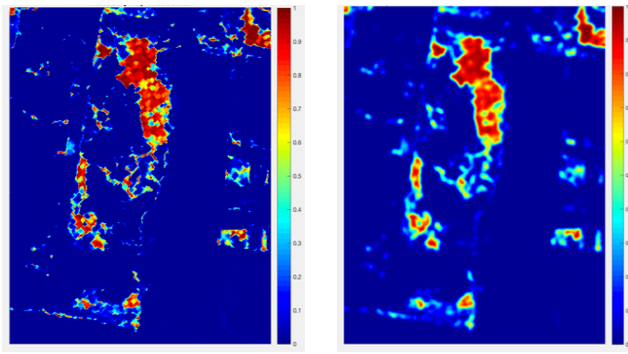
(b) Left: Probability Map of Soil before Post-processing. Right: Probability Map of Soil after Post-processing in Low Resolution.



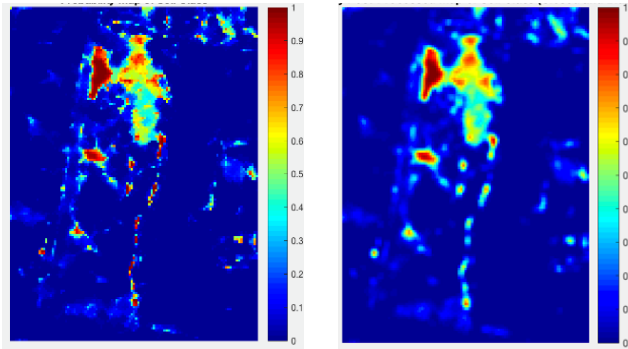
(c) Left: ROC Curves before and after Post-processing in High Resolution. Right: ROC Curves before and after Post-processing in Low Resolution.

Fig. 29. Post-processing Results for Testing Image Dated 10/11/2010.

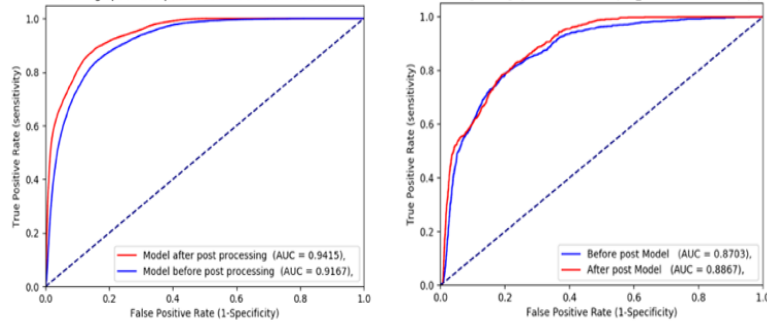
Fig. 30(a) shows detection results before and after the post-processing step on the soil prediction map produced by using 88-channel hyperspectral data in high resolution, and Fig. 30(b) shows the corresponding results for the low-resolution case. Fig. 30(c) shows the ROC curves before and after the application of the post-processing step in both the high resolution and low-resolution cases. The AUC scores were improved by 2.48% in high-resolution and 1.64% in low-resolution, respectively.



(a) Left: Probability Map of Soil before Post-processing. Right: Probability Map of Soil after Post-processing in High Resolution.



(b) Left: Probability Map of Soil before Post-processing. Right: Probability Map of Soil after Post-processing in Low Resolution.



(c) Left: ROC Curves before and after Post-processing in High Resolution. Right: ROC Curves before and after Post-processing in Low Resolution.

Fig. 30. Post-Processing Results for Testing Image Dated 10/11/2010.

Table 4 summarizes all testing results in this section. It is observed that the CNN model achieved significantly better soil detection performances by using the 88-channel synthetic hyperspectral data for both the high-resolution and low-resolution cases. The pan-sharpened high-resolution data have a 0.46m spatial resolution while the low-resolution data have a 1.84m spatial resolution. The detection performance also benefited from the high-resolution images.

TABLE 4. AUC Scores of CNN Models on Testing Image

Soil Detection CNN Models	AUC Scores		
	Testing Image 3/19/2010	Testing Image 10/11/2010	Testing Image 12/2/2010
8_High_Resolution	0.6745	0.8582	0.9106
88_High_Resolution	0.9619	0.8654	0.9167
Improvement before Post-processing	0.2874	0.0072	0.0061
Improvement after Post-processing	0.3026	0.0667	0.0309

Soil Detection CNN Models	AUC Scores		
	Testing Image 3/19/2010	Testing Image 10/11/2010	Testing Image 12/2/2010
8_Low_Resolution	0.7512	0.7715	0.7651
88_Low_Resolution	0.8484	0.7857	0.8703
Improvement before Post-processing	0.0972	0.0142	0.1112
Improvement after Post-processing	0.1272	0.0605	0.1276

3.5 Conclusion

In this deep learning based hyperspectral image processing application, we implemented a CNN model for soil detection. The detection performance has been significantly improved by using 88-channel synthetic hyperspectral bands generated by the EMAP method. We also demonstrated that pan-sharpening and morphological post-processing can further improve the soil detection performance. These results indicated that even though the synthetic hyperspectral bands may not have the same physical meanings as the real hyperspectral bands, they hold the correlations with the spatial characteristics of the objects. Along with the spectral information in the original bands, the synthetic hyperspectral data with the increased spatial resolution is a good alternative for improving object detection and classification in remote sensing applications when real hyperspectral data is not available. For the future work, we will investigate furtherly if there is a subset of the synthetic hyperspectral bands which highly correlated to the soil class and more efficient in detecting the soil class by using non-linear dimension deduction methods such as principal component analysis or deep autoencoder, by deducting the dimensions of the bands, it will highly likely accelerate the model training and expedite the convergence, and potentially improve the detection accuracy and increase the robustness of the CNN model.

CHAPTER 4

EFFECTIVE REFUGEE TENT EXTRACTION BY FCN

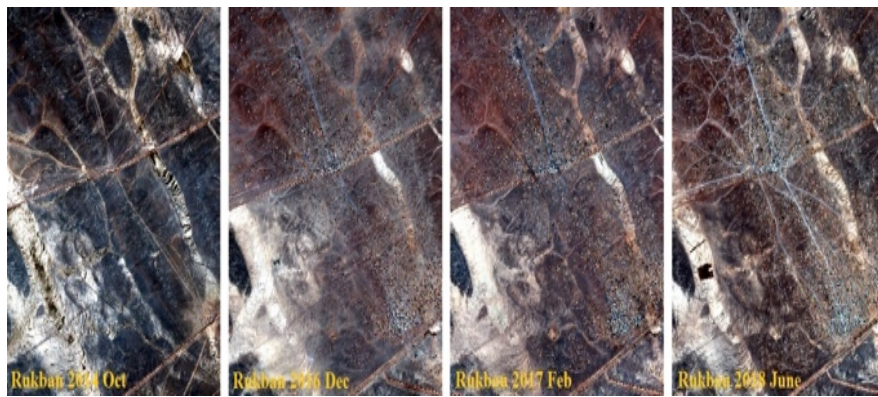
Rukban is an arid remote desert area crossing the border between Syria and Jordan. Thousands of Syrian refugees have fled to this area since the Syrian civil war in 2014. In the past four years, the number of the shelters for the forcibly displaced Syrian refugees in this area has increased rapidly into the tens of thousands. The fast-growing population resulted in severe lack of life-support resources such as water, food, and medicine. Estimating the location and number of refugee tents has become a key factor in maintaining the sustainability of the shelter camps. However, these shelters/tents are usually small in size, irregular in shape, and sparsely distributed in a 35 square miles area and could be easily missed by traditional analysis techniques. Manually counting the number of shelters is labor-intensive and prohibitive given the large quantities. In this section, we proposed a deep Fully Convolutional Neural Network (FCN) method to automatically detect and count the refugee shelters/tents in the Rukban area by using the Worldview-2 (WV-2) satellite images. We applied transfer learning with the pre-trained VGG-16 model for improved accuracy and faster network convergence. We also implemented the traditional Spectral Angle Mapper (SAM), deep Convolutional Neural network (CNN) and Mask R-CNN methods as comparison, our experiment results show that our proposed FCN method achieved significantly better performance than the other models and greatly reduced computational complexity in small ground objects detection. FCN model improved the overall accuracy by 4.49%, 3.54% and 0.88% as compared to the CNNs, SAM and Mask R-CNN models, and improved precision by 34.61%, 41.99% and 11.87%, respectively.

4.1 Introduction

According to the United Nations, the civil war in Syria has resulted in the largest global refugee crisis and humanitarian disaster in human history. More than 5.6 million people were forced to flee their homes from the ISIS-held part of Syria to the neighboring countries seeking asylum [108]. A number of refugees between 40,000 to 50,000 mostly women and children were desperately stranded in an isolated formerly barren desert area known as Rukban near the southern border of Syria and Jordan. Since the Rukban refugee camp was established in 2015, this area started getting crowded with acres of temporary or long-term white refugee tents. In just about four years, the number of refugee tents and shelters has increased roughly from 132 to 11,702, an almost 89 times increase according to the UNOSAT [109]. Challenged by the severe natural condition, the threat of disease and deaths are growing every day in Rukban, the fast-growing population are in urgent need of humanitarian assistance including food, water, medicine, and lifesaving emergency aid. To provide quantitative perspective to the humanitarian aid organizations such as the United Nations High Commissioner for Refugees (UNHCR) for effective campsites planning, fields operations and rescue effort, accurate mapping, and estimation of the number of refugee shelters have become the key factors and critical consideration maintaining the sustainability and viability of the settlement and the well-being of the refugee population. Fig. 31(a) shows a typical refugee shelters/tents camp, the white structures are the most commonly seen refugee tents in this area. Figure 31(b). demonstrates the fast-growing refugee camps in this area from October 2014 to June 2018.



(a)



(b)

Fig. 31. A Typical Refugee Camp in the Rukban Area.

In this deep learning-based remote sensing image processing application, we proposed a deep fully convolutional neural network (FCN) method for accurate refugee shelters/tents detection by analyzing the Worldview-2 satellite imagery. Our contributions are summarized as below:

- 1) We successfully learned an end-to-end and pixel-to-pixel FCN model which consists of convolutional and deconvolutional layers by semantic segmentation for the small ground object detection to detect the refugee tents in Worldview-2 satellite imagery.

- 2) We implemented bilinear interpolation deconvolution to up-sample the feature maps in the convolutional layers, which makes the FCN model is able to utilize both the spectrum and spatial information in the remote sensing imagery to generate outputs.
- 3) We applied transfer learning to initialize the FCN model with the pre-trained VGG-16 model. We fine-tuned the FCN model with a small training dataset which was manually labeled in the image. The transfer learning strategy mitigated the lack of training data issues and significantly improved the performances of the FCN model.
- 4) We firstly introduced and developed the FCN approach for detecting small ground objects with sparse distribution using large scale satellite images which has not been done before based on our research. To the best of our knowledge, this is the first attempt to apply FCN with transfer learning for refugee tents detection.
- 5) The FCN model is scale-free, it is also able to analyze images with any size that is larger than the input patch size. Our experiment results show that our proposed method achieved significantly better performances as compared to the CNN and Mask R-CNN models.

4.2 Motivation

CNNs have been successfully applied in a wide range of object detection and classification applications. CNNs were firstly introduced by LeCun in 1998 [86] and later were improved and won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [87]. CNNs were inspired by mammal's natural vision systems and one important property of CNN model is the "end-to-end" learning: the model simultaneously and automatically learns feature representations and tunes parameters to generate final output - no feature extraction and post-processing procedures are needed.

CNNs models usually adopt pre-defined fixed-size receptive fields as convolutional filters to extract features and utilize patch-wise training to predict center pixel. When patch size is substantially larger than the object size, CNN models will smooth out the details of edges and boundaries of the objects to be detected. Accordingly, small objects such as refugee tents are prone to be fragmented, misclassified or ignored by CNN models. Furthermore, semantic information in the image is not utilized in the patch-wise CNN models for object detection.

Region-based CNN models such as R-CNN [110] and its improved variants Fast R-CNN [111] and Faster R-CNN [112] have been developed for object detection. Those models utilize region proposals to group adjacent pixels for object detection. Mask R-CNN [113] extended the Faster R-CNN model by adding an FCN layer on top to generate a pixel-wise binary mask for better segmentation. However, the detection of cluttered, small objects in remote sensing imagery remains as a challenging task for Mask R-CNN as verified in our experiments.

To overcome the limitations mentioned above, we proposed an FCN model for refugee tent extraction in multispectral satellite images. FCN was firstly proposed by Long et al. [114]. A typical FCN model adopts the backbone structure of CNNs and uses convolutional layers to extract coarse feature maps. It then utilizes up-sampling deconvolutional layers to form pixel-wise label maps for the whole input image. FCN performs end-to-end, pixel-to-pixel inference for semantic segmentation in images with arbitrary sizes. Through training, FCN incorporates both the spectral and spatial information from input image to form a segmentation output. FCN does not involve any region proposal so it is very effective to extract crowded small objects in the image. To the best of our knowledge, our experiment is the first attempt to apply FCN for refugee tent extraction in remote sensing imagery.

4.3 Methodology

4.3.1 FCN Model

The system architecture and workflow of the FCN model is depicted in Fig. 32. The backbone of our FCN model is inherited from the structure of the VGG-16 model, VGG-16 is a convolutional neural network proposed by Karen Simonyan and Andrew Zisserman [87]. The model did exceptionally well in ILSVRC14, it achieved 92.7% top-5 test accuracy on ImageNet - the image dataset consists over 14 million images belonging to 1000 classes. Starting with the original VGG-16 structure, we discard the final classifier layer, then convert all fully connected layers to convolution layers (Conv6-7), and keep all other layers in our model. In addition, at each of the coarse prediction output locations: the Pool3, Pool4 and Pool5 layer, convolutional layers with $1 \times 1 \times$ dimensions of the classes number are appended after to predict the scores for each of the classes (Predict1-3); following each of the prediction layers (Predict1-3), a deconvolution layer is appended. The purpose for appending a deconvolutional layer at each of the coarse prediction locations is to up-sample the coarse outputs to pixel-based dense predictions, it also recovers the prediction maps to the corresponding original input sizes. The deconvolutional layers are structured hierarchically, which use a skip structure summing the results of each prediction layer and the lower prediction layer by $2 \times$ bilinear interpolation up-sampling.

The FCN model contains 15 convolutional layers and 5 max-pooling layers. Each convolutional layer is followed by a ReLU activation layer. All convolutional layers have the same filter sizes as these of VGG-16. The convolution stride is 1 pixel with zero paddings. All pooling layers use 2×2 windowsw to scale down to half size, requiring a minimum input image size of 32×32 .

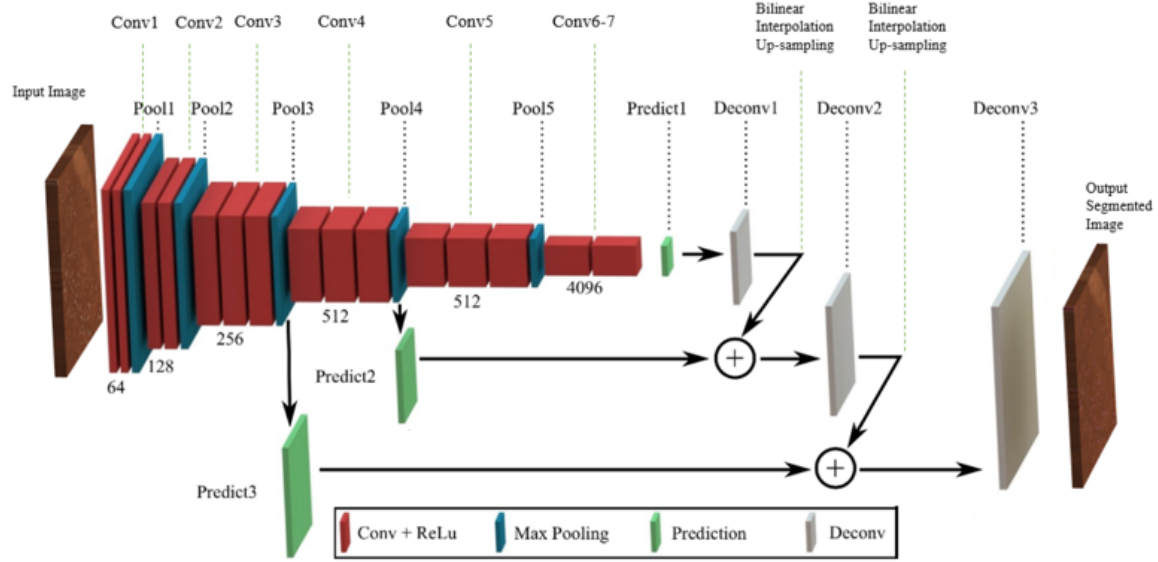


Fig. 32. System Architecture of Our FCN Model.

4.3.2 SAM Model

SAM [115] is a classical remote sensing image processing technique which is widely used for data analysis in geophysics, oceanography, and atmospheric science. SAM exploits the spectral signature of an object at the pixel level. It computes the spectral angles between the target pixel and a testing pixel in each band. The smaller the angle, the more similar is the two pixels. Based on the spectral angle, an object is identified by grouping similar pixels. The spectral angle is computed as,

$$\theta(x, y) = \cos^{-1} \left(\frac{\sum_{i=1}^N x_i y_i}{(\sum_{i=1}^N x_i^2)^{1/2} (\sum_{i=1}^N y_i^2)^{1/2}} \right) \quad (27)$$

where x is the target pixel, y is the test pixel and N is the number of bands. After computing the spectral angles between all paired pixels, a threshold value is defined to exclude pixels that are not similar, and an object is identified by grouping similar pixels.

4.3.3 CNN Model

We implemented three CNN models with different structures in this study denoted as CNN-7, CNN-5, and CNN-3. CNN-7 takes image patches with a size of 7x7 as input and consists of four convolutional layers and one fully connected layer as shown in Fig. 33. The first three convolutional layers have 20, 20 and 100 3x3 filters, respectively, and the fourth convolutional layer has 100 1x1 filters. The fully connected layer contains 100 hidden units. Each convolutional layer is followed by a ReLU layer and a Dropout layer with a dropout rate of 0.1 to avoid overfitting [116]. CNN-5 takes image patches of 5x5 as input and consists of two convolutional layers and a fully connected layer with 100 hidden units. Each convolutional layer has 20 3x3 filters and is followed by a ReLU layer and a Dropout layer with a dropout rate of 0.1. The size of the input image patch for CNN-3 is 3x3 and there are two convolutional layers having 20 2x2 filters followed by a fully connected layer with 100 hidden units. Each convolutional layer is followed by a ReLU layer and a Dropout layer with a dropout rate of 0.1. All CNN models use a SoftMax layer at output for classification.

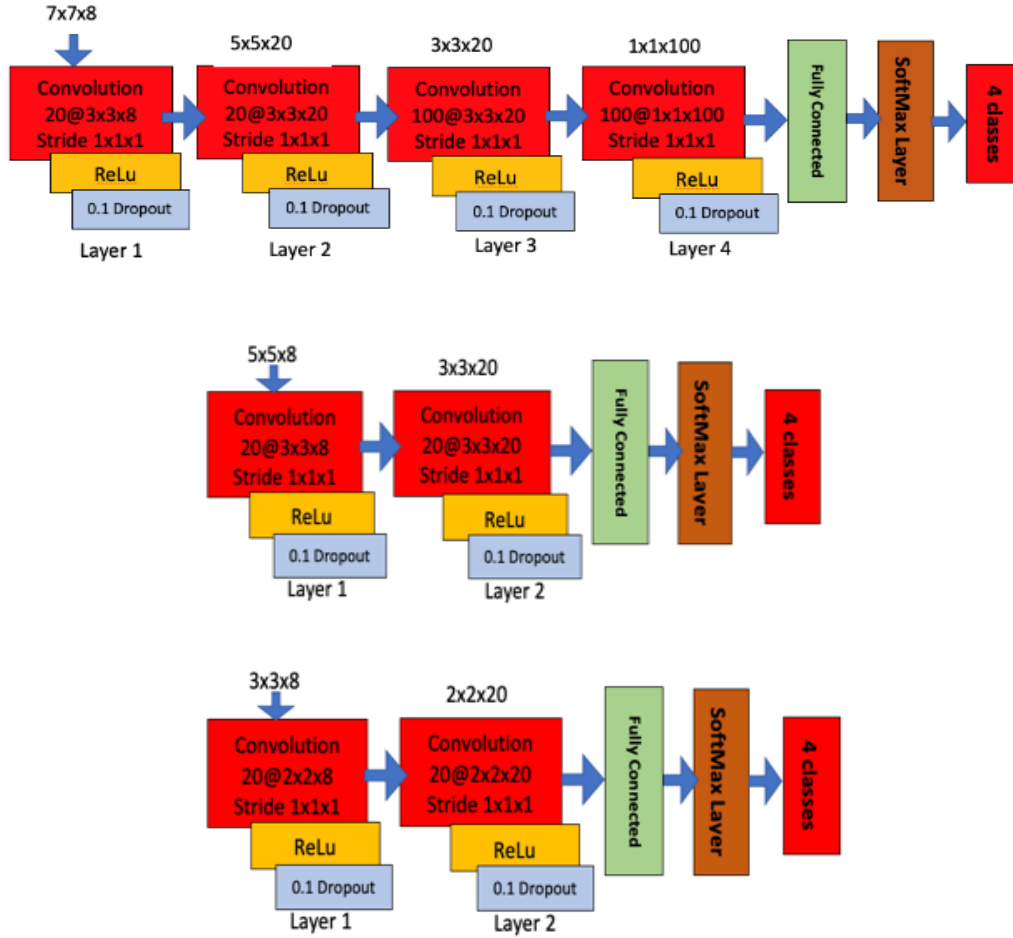


Fig. 33. CNN Structures. Top: CNN-7. Middle: CNN-5. Bottom: CNN-3

In order to identify the tent class, to properly train a CNN classifier model, we need to collect all classes appearing in the image; however, we are just interested in the tent class, and the training samples for the tent and non-tent classes are most likely imbalanced. We followed the following steps to resolve the imbalance challenge. First, we randomly sampled all training patches from each training class to the size of the training class with the smallest patches size, then we combined all selected data from all classes to train the CNN model. At the output layer, we apply the conversion as,

$$P'_{\text{Tents}} = \frac{P_{\text{Tents}}}{P_{\text{Tents}} + \max(P_{\text{Non-Tents}})} \quad (28)$$

$$P'_{\text{Non-Tents}} = \frac{\max(P_{\text{Non-Tents}})}{P_{\text{Tents}} + \max(P_{\text{Non-Tents}})} \quad (29)$$

to convert the multi-classes problem to a two-class problem. P_{Tents} and $P_{\text{Non-Tents}}$ are the probabilities of the tents and all other non-tents classes in the testing stage. P'_{Tents} and $P'_{\text{Non-Tents}}$ are the converted 2-classes probabilities of tents class and non-tents class.

4.3.4 Mask R-CNN

Mask R-CNN model [10] is built on top of Regional-based CNN (R-CNN), Fast R-CNN and Faster R-CNN [9].

The main idea in R-CNN is two-stage training. In the first stage, it uses selective research to identify a number of bounding box candidates for the objects to detect which is also called the region of interest (RoI). In the second stage it uses regular CNN to extract features from each region independently for classification. Then it continues fine-tuning the CNN with the proposed region with $K+1$ class where K is the class number and the extra one class is for the background. The Non-Maximum Suppression function is used to search multiple bounding boxes for the same object: it will sort all the bounding boxes with confidence score and discard boxes with low confidence scores and then repeat this step until the remaining boxes with the highest intersection over union (IoU) score.

However, the R-CNN model is expensive and slow. On top of the R-CNN structure, instead of using CNN to extract features for each independent region proposal, Fast R-CNN integrate all CNNs to one CNN forward pass over the entire image and the region proposal to share the feature matrix is used for learning the object classifier and bounding-box regressor. This step greatly

speeds up R-CNN. So, there are two steps training in Fast R-CNN, it will firstly pre-train a CNN on image classification tasks, then use selective search to propose regions. Then it replaces the last max-pooling layer of pretrained CNN with RoI pooling layer. And then replace the last fully connected layer and the last SoftMax layer of K classes to K+1 class. In the last step, a SoftMax estimator of K+1 class will output a discrete probability distribution per RoI. And a bounding-box regression model which predicts offsets to the original RoI for each of K classes.

Faster R-CNN uses Fast R-CNN to initialize the regional proposal network (RPN) for the region proposal task, it only fine-tunes the RPN layers while keeping the shared convolutional layers. Mask R-CNN extends Faster R-CNN to pixel-level image segmentation. It added a third network for predicting an object mask in parallel with current network. This mask network is a fully convolutional network applied to each RoI to form a segmentation mask at pixel-level. Fig. 34.

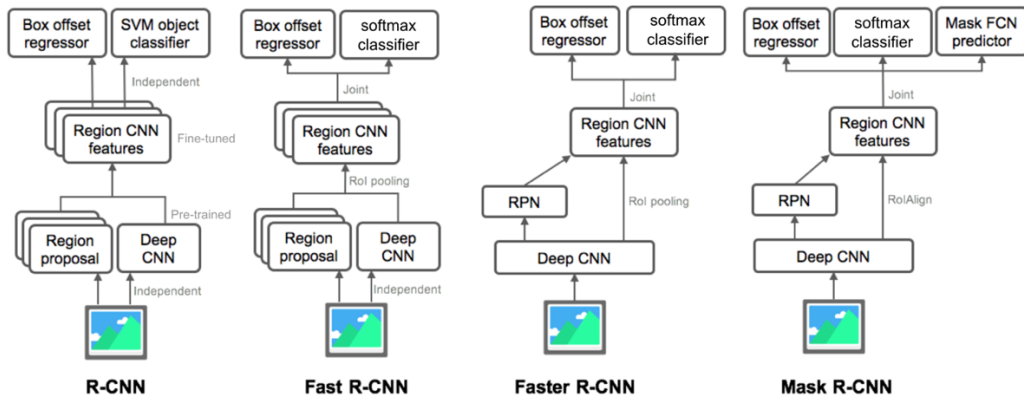


Fig. 34. Summary of Models of R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN

In our experiment the network structure of Mask R-CNN model is as Fig. 35 shown. we used the pre-trained ResNet-50 architecture as the backbone to extract feature maps for a fair comparison.

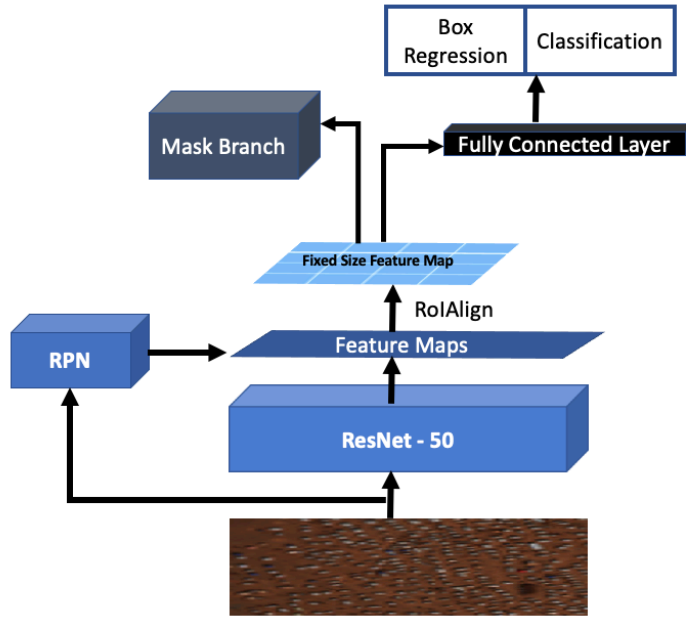


Fig. 35. Mask R-CNN with ResNet-50 Backbone for Tent Detection

4.3.5 Data Preparation

Survey data from UNOSAT and satellite images from the WV-2 satellite are used in this study. UNOSAT conducted refugee camp analysis by assessing satellite images collected since 2014. The number of tents on 20 non-consecutive dates from the year of 2014 to the year of 2019 have been reported. We used this dataset to evaluate our proposed model. The WV-2 satellite image dataset contains multispectral images collected in an area of 25 kilometers southwest of the Al Waleed along the Jordanian border side by Maxar on February 13, 2016 (Time-1) and February

17, 2017 (Time-2). This data has 8 channels and the size of the image is 2422x1727 pixels as shown in the first column of Fig. 40.

To train the CNN model, we created masks for 4 classes of visually observed ground objects from the Time-1 image as the training data for the CNN model, the four classes are: (a). Tents (b). Rocks (c). Sands (d). Roads, as shown in Fig. 36. We extracted patches with corresponding sizes to train the three CNN models. The numbers of patches we extracted are shown in Table 5. The numbers of patches for the four classes are not the same and the tent class has the least patches (around 26k). We randomly sub-sampled the other three classes to balance the training data.

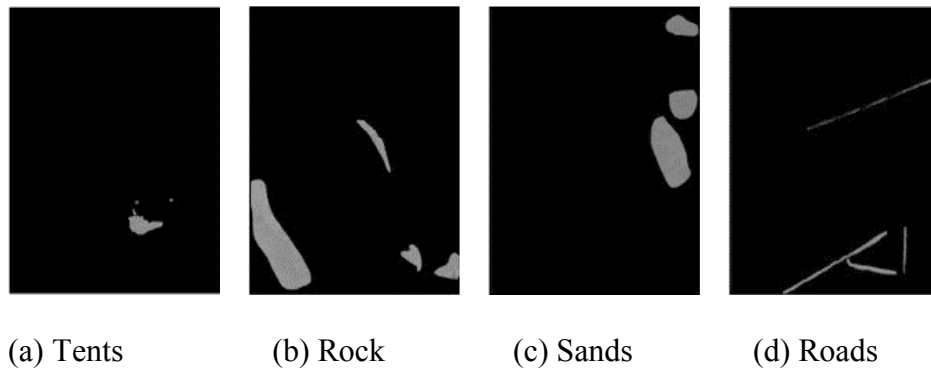


Fig. 36. Masks of the 4 Classes for Training CNN

TABLE 5. Number of Extracted Patches for the 4 Classes

Class Name	Class ID	Number of Patches 7*7*8
Tents	1	26192
Rocks	2	259319
Sands	3	189979
Roads	4	37907

Three small areas in sizes of 141×201 , 201×206 and 101×101 from Time-1 image are selected as training data for FCN model and Mask R-CNN model, they are as shown in the top row in Fig. 37. The corresponding ground truth mask for the tent class are shown in the bottom row in Fig. 37.

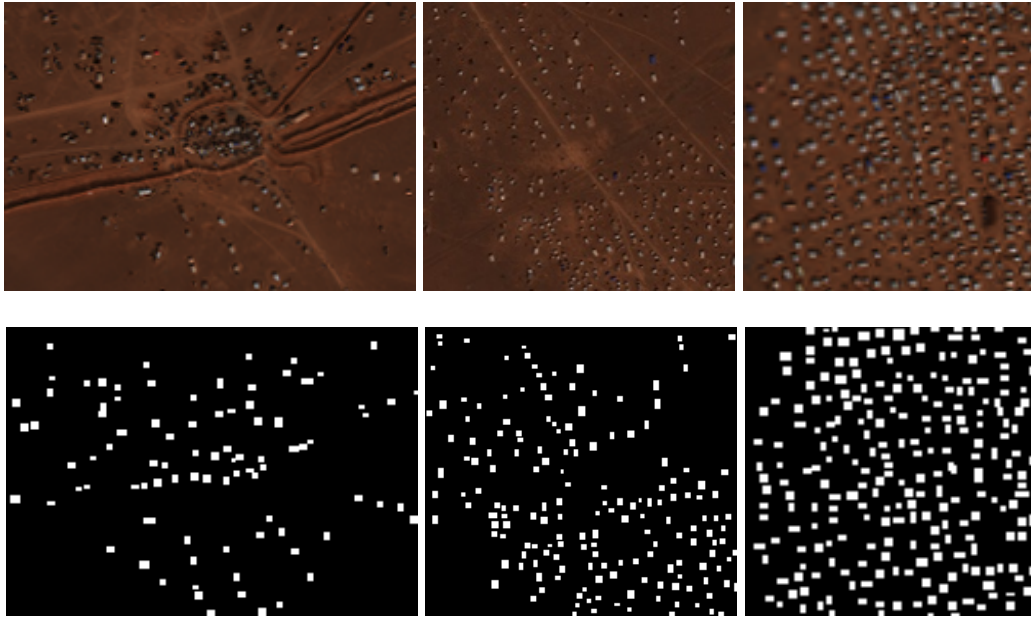


Fig. 37. Training Data for FCN.

Another cropped area in size of 384×384 is selected from Time-1 image for model validation as shown in Fig. 38, where the ground truth mask was also created manually.

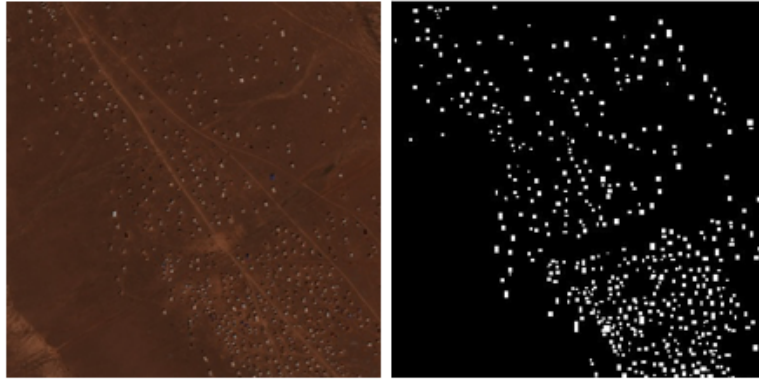


Fig. 38. Validation Data and its Ground Truth from Time-2

4.3.6 Performance Metrics

Accuracy, precision, recall, F1-score, precision-recall score, and receiver operating characteristics (ROC) curve are computed to evaluate the performance of all of the models. Accuracy is simply the ratio of the correctly predicted observation to the total observations; precision is the ratio of correctly predicted positive observations to the total predicted positive observations, we use precision to show how precise/accurate of the model; recall is the ratio of correctly predicted positive observations to the all observations in the actual class, we use recall to calculate how many of the actual positives the model catches by labeling it as positive; F1-Score is the weighted average of precision and recall, we use F1-score to check the balance between precision and recall when there is an uneven class distribution. We also evaluate our model by ROC curve and precision-recall score. The ROC curve and area under the curve (AUC) are usually used as performance metrics when there are roughly equal numbers of observations for each class. The precision-recall score is more often used as the performance metrics when there are extremely imbalanced classes.

4.4 Model Training and Experimental Results

4.4.1 Patch Size Determination for FCN

We trained the FCN model with four input image sizes of 32×32 , 64×64 , 96×96 and 128×128 , on the training data and validated the trained models FCN-32, FCN-64, FCN-96 and FCN-128 on the validation data. The validation results are shown in Table 6. The tent detection maps are shown in Fig. 39. It is observed that FCN-32 has the best performance by all metrics. Therefore, we selected FCN-32 for the subsequent experiments.

TABLE 6. FCN Performances with Different Patch Sizes

	Accuracy	Precision	Recall	F1-Score	P-R	AUC	IoU
32	0.9654	0.6235	0.5288	0.5722	0.3503	0.7571	0.6844
64	0.9623	0.5827	0.4871	0.5306	0.3063	0.7355	0.6643
96	0.9595	0.5430	0.4706	0.5042	0.2787	0.7265	0.6498
128	0.9567	0.5059	0.4399	0.4706	0.2471	0.7101	0.6347

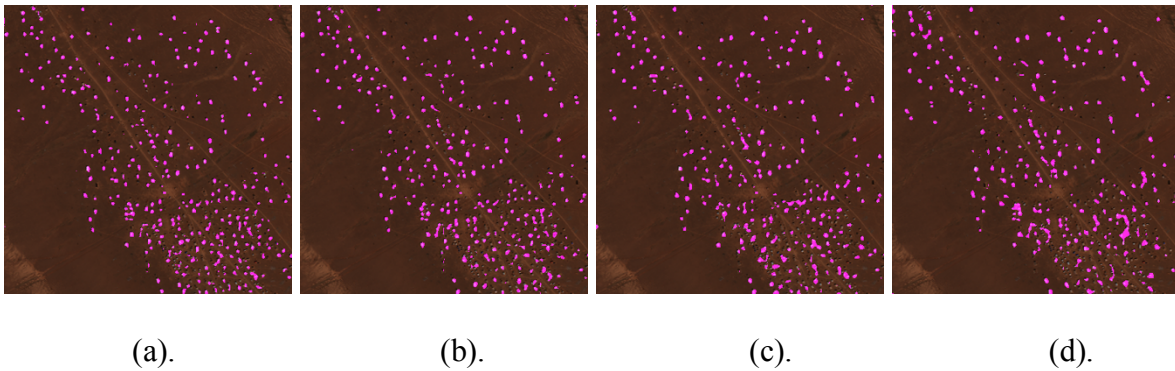


Fig. 39. FCN Tent Detection Maps in a Common Area from the Cropped Time-2 Data

4.4.2 Model Comparison

After the input block size of FCN is determined, we compared SAM, CNN-3, CNN-5, CNN-7 and Mask R-CNN with FCN-32 on validation data, we compare and evaluate all models by using average accuracies, precisions, recalls, F1-scores, precision-recall scores and average intersection over union (IOU) scores. The results are shown in and the results are shown in Table 7. Threshold values of 0.15 and 0.99 were chosen for SAM and CNN to obtain the binary results. The tent detection maps for Time-1, Time-2 and the validation image by different models are shown in Fig. 40. From the results, we can observe that the FCN-32 model has the best performances in all metrics among all compared models. The FCN model improved AUC, IoU and average accuracy by 3.41%, 11.51% and 4.49%, respectively, as compared to the best CNN model. It also improved AUC, IoU and average accuracy by 17.46%, 16.23% and 3.54%, respectively, as compared to the SAM method. In addition, the FCN model achieved similar average accuracy and AUC as compared to Mask R-CNN, but improved IoU by 3.05%.

TABLE 7. Model Comparison on Validation Data

	Accuracy	Precision	Recall	F1-Score	P-R	AUC	IOU
SAM	0.9304	0.2037	0.2071	0.2054	0.0766	0.5851	0.5221
CNN-3	0.9209	0.2775	0.4148	0.3599	0.1632	0.7256	0.5693
CNN-5	0.8597	0.1537	0.6723	0.2346	0.0980	0.6856	0.4947
CNN-7	0.8007	0.1061	0.4510	0.1740	0.0737	0.6491	0.4459
Mask RCNN	0.9570	0.5049	0.5371	0.5205	0.2913	0.7565	0.6539
FCN-32	0.9658	0.6236	0.5288	0.5754	0.3533	0.7597	0.6844

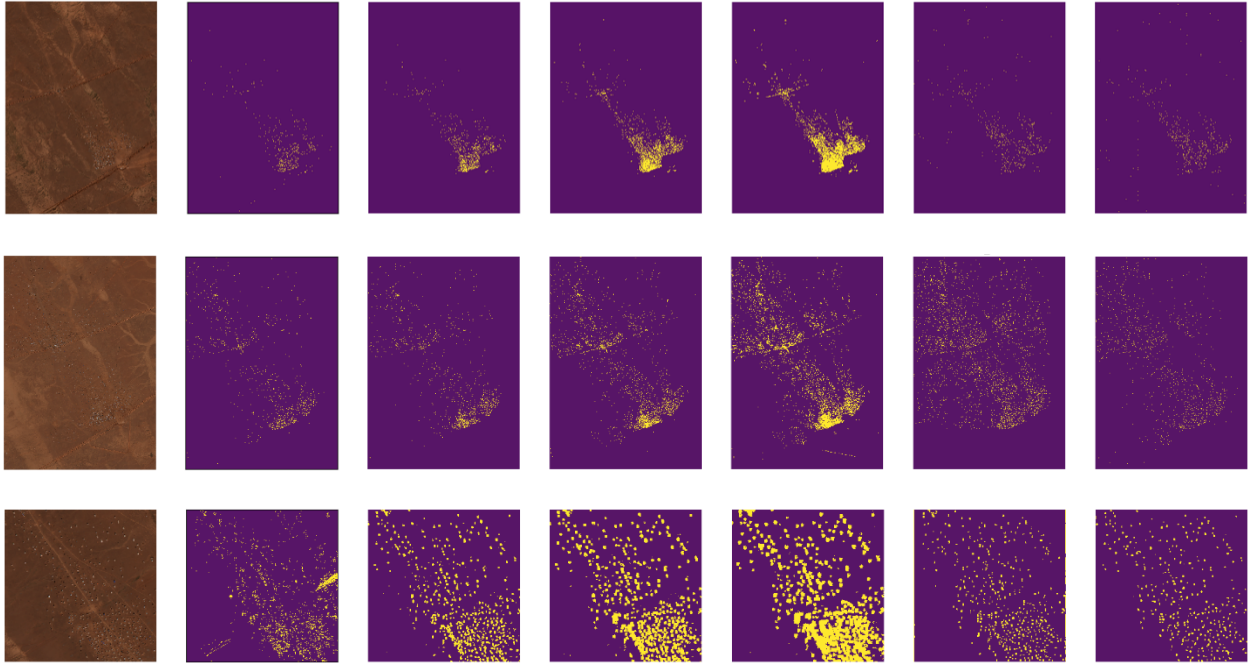


Fig. 40. Tent Extraction Maps in Time-1 (1st row), Time-2 (2nd row) and Validation Images (3rd row). From the left to right in each row: The Original RGB Images, Results by SAM, CNN-3, CNN-5, CNN-7, Mask-RCNN and FCN-32.

4.4.3 Estimation of the Number of Tents

According to the report by UNOSAT, a standard refugee tent is semi-circular or tunnel shaped with center height of 2.1 meters, width of 5 meters and length of 3.8 meters. It has 16 square meters main floor area, plus two 3.5 square meters vestibules, for a total area of 23 square meters. The spatial resolution of WV-2 multispectral image is 1.84 meters so that a tent consists of approximately 7 pixels. We counted the number of detected pixels and the estimated number of tents by different models is shown in Table 8. Compared to the UNOSAT survey results and the ground truth data, FCN achieved the best results. In the validation image, in particular, FCN has

an error of 1.55% and wins over the second-best method by a large margin (Mask R-CNN, 12.90%).

The results as shown in Table 8.

TABLE 8. Tent Number Estimation by Different Models

	Time-1-image 2016-02	Time-2-image 2017-02	Validation data
UNOSAT	3365	6460	775*(manual)
SAM	2643(21.25%)	5381(16.70%)	906(16.90%)
CNN-3	5495(63.74%)	6703(3.76%)	1917(147.35%)
CNN-5	11600(245.65%)	13724(112.45%)	3348(332.00%)
CNN-7	19788(489.33%)	27045(318.65%)	4735(510.97%)
Mask R-CNN	3245(3.31%)	7486(15.88%)	875(12.90%)
FCN-32	3365(0.27%)	6672(3.28%)	763(1.55%)

4.5 Conclusion

In this chapter, we proposed a fully convolutional neural network model for refugee tent extraction and achieved the best results as compared to the traditional SAM method, CNN models and the Mask R-CNN model. However, recent study suggested that the FCN model may lose the global context of the input image in its deep layers. We will investigate to incorporate global scene-level context to improve FCN. In addition, we will study the impacts of cloud on tent extraction in remote sensing images as our future work. This proved our ideas proposed at the beginning of this paper: pixel-based FCN model detect and infer better than the patch-based CNN models for small ground objects detection in remote sensing images especially when the objects are small in size, irregular in shape with a very sparse distribution in large regions. Besides, the FCN model

tremendously reduced the complexity of collecting procedure for the training data, only the ground truth of the target objects is needed to be created to train an FCN model.

CHAPTER 5

DATA AUGMENTATION BY GAN

RS data have a high dimensionality and a sufficiently large dataset is essential to effectively train a deep learning model for RS classification. In practice, data augmentation methods are usually used to enlarge the training dataset by rotating, scaling, flipping, and transforming. However, it has limited capability to capture the data distribution with background context information and often yields diversified data. In this chapter, we investigate a Generative Adversarial Network (GAN) based data augmentation model, named SinGAN, to generate training samples. We tested the generated data samples with the FCN network proposed in Chapter 4 and compared the result with traditional data augmentation methods.

5.1 Introduction

It is very important to have large high-quality datasets to train deep models, meaning the dataset should not only be sufficiently large but also should cover as many data variations as possible. There are many reasons that large datasets can not be obtained such as privacy or cost issue. Under these situations, if we would like to train or improve the performance of deep models, one possible approach is to generate synthetic datasets for training. There were many efforts had been made in the past to enlarge the training samples, such as oversampling the minority data in an imbalanced dataset, or generating new training samples by flipping, rotating, scaling, or adding noises to original data to reflect the real-world changes.

The GAN model [117] is a deep generative model that can be trained to generated images from random noise. A GAN model consists of two parts: a generator and a discriminator. The

network structure is shown in Fig. 41. The generator captures distribution of training data and generates synthetic data samples. The discriminator estimates the probability that if a sample is from the training dataset rather than the generator. At the initial training stage, random noises are generated as inputs to the generator. After the generator generates a fake image, this fake image and the real image will be fed to the discriminator for classification. The generator is constantly trying to fool the discriminator by generating better fake images while the discriminator is trained to become better to distinguish the real and fake images. Through this two-player min-max adversarial training process, GAN is getting better and better in estimating the potential distribution of the training data. Finally, it will be capable of generating realistic images following a similar distribution as the training data. GAN has been widely applied in various computer vision applications such as super-resolution [118], image-to-image translation [119], style transfer [120], photo blending [121] and image inpainting [122] etc.

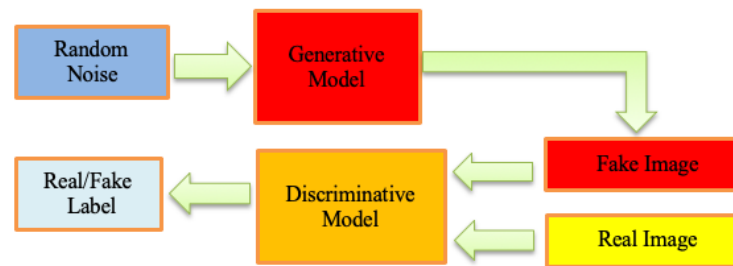


Fig. 41. GAN Network Model

Many researchers have noticed that GANs are capable of generating new data via the adversarial training process and are promising solutions to tackle the lacking of training data issue. In remote sensing field, Ma et al. [123] proposed a data argumentation method by GAN in scene classification. Lin et al. [124] proposed a multiple-layer feature-matching GAN model to help the

classification of RS images. Yan et al. [125] used GAN for generating simulated training samples for ship detection. Zhang et al. [126] used GAN for simulated SAR datasets.

In this chapter, we proposed a GAN-based approach to generate refugee tent samples to augment the original data samples to furtherly improve the performance of the FCN model proposed in Chapter 4. Meanwhile, we compared the GAN-based data augmentation method with other traditional image augmentation methods. Our experiment results show that GAN improved the overall classification performance of the FCN model, and the IoU has been improved ranging from 0.2% to 1.5% with different sample sizes and input image sizes.

5.2 Methodology

5.2.1 SinGAN Model

SinGAN model is proposed by Tamar et al. [127] in 2019. SinGAN is an unconditional generative model that can learn from one single image and generates high-quality and diverse samples that carry the same visual content as the original image. It can be used in a number of image manipulation tasks such as super-resolution, paint-to-image conversion, image harmonization and editing through learning from a single image. The network structure is shown in Fig. 42. It consists of a pyramid of fully convolutional GANs to generate new samples of arbitrary size and aspect ratio. Each pair of generators and discriminators learns representation at different scales. The generator and discriminator at the lowest scale learn coarse features like background, and the high-level pair learn fine details like edges and corners. The input image is down-sampled to the corresponding size and input to the discriminator along with the generator's output.

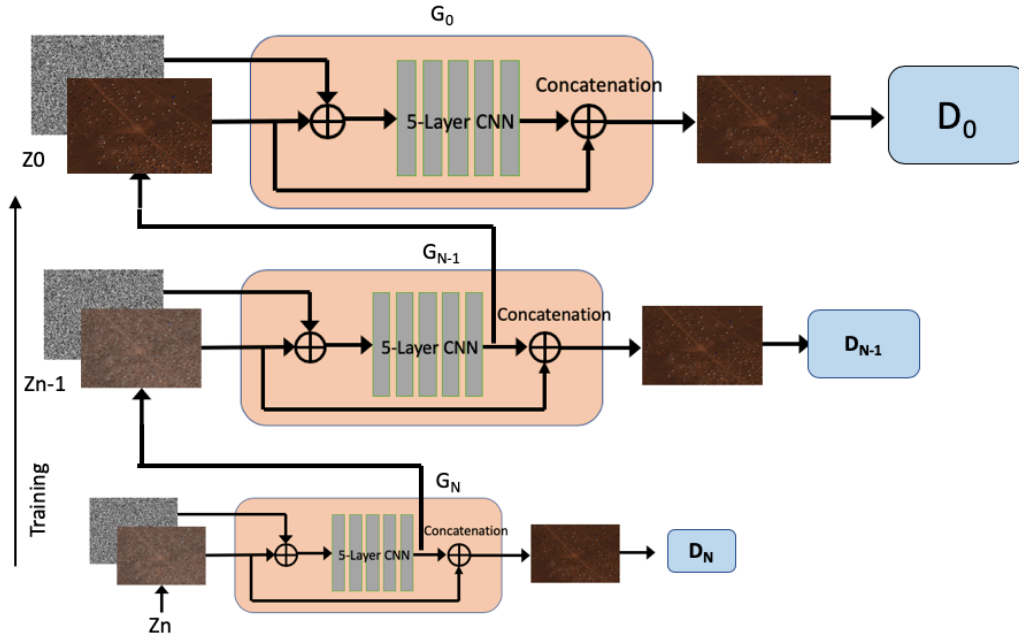


Fig. 42. SinGAN's Multi-scale Pipeline.

As shown in Fig. 43, the input to the generator at each level are the random noises with the generated image from the lower-level generator. Since SinGAN is generating images from a single image, a sliding window moves over the whole images to gathering the training samples. The input patch size decreases when the image input is getting bigger in the upper-level network. During the training, the random noises z_N with the unsampled generated image x_{n+1} (to the noise size) from the lower level are concatenated together as input to the following convolutional layers, and the output of the convolutional layers is concatenated again with the generated image from the lower-level network. Then this output will be input to the discriminator along with the down-sampled image to the discriminator. The discriminator consists of five convolutional layers as well, but there is no concatenation of the real and generated image.

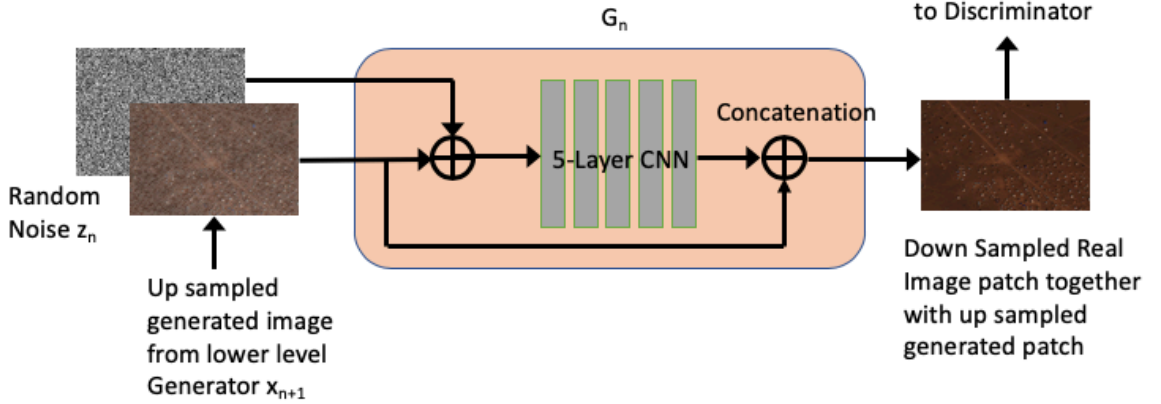


Fig. 43. Single Scale/Level Generation.

5.2.2 Training and Loss Function

As shown in Fig. 43, the SinGAN network is trained hierarchically, from lower-level to higher-level in a stacked GAN model. Once one level of GAN is trained, it will be kept fixed. The SinGAN loss consists of two loss functions: the adversarial loss and the reconstruction loss,

$$\min_{G_n} \max_{D_n} \mathcal{L}_{\text{adv}}(G_n, D_n) + \alpha \mathcal{L}_{\text{rec}}(G_n) \quad (34)$$

The adversarial loss is used to penalize the network for matching the distribution of the generated samples and the distribution of original samples by calculating the distance between the distributions of real data and data generated by the generator. The reconstruction loss in Eq. 35 is used to penalize the network for generating samples that look like the original samples. The root mean squared error loss (RMSE) is used for calculating the reconstruction loss,

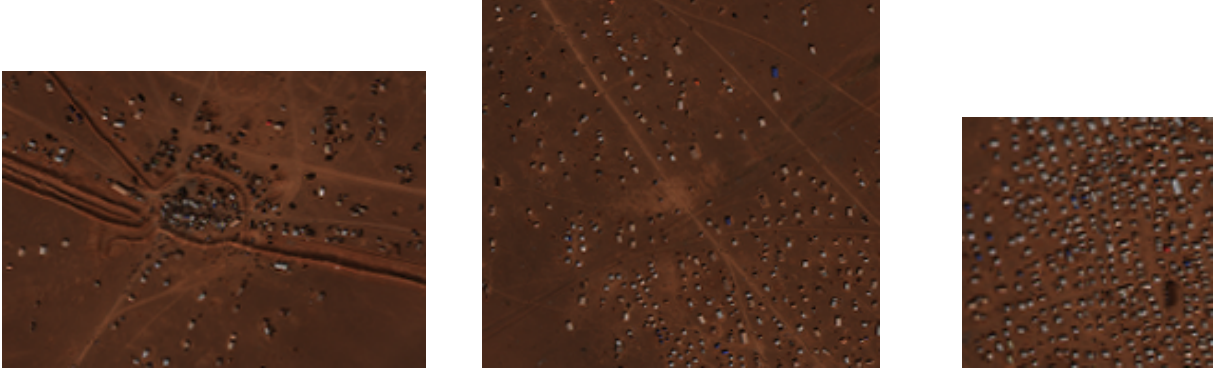
$$\mathcal{L}_{\text{rec}} = \| G_n(0, (\tilde{x}_{n+1}^{\text{rec}})^{\uparrow r}) - x_n \|^2 \quad (35)$$

In the first level of SinGAN, it only has noise input, so the reconstruction loss is,

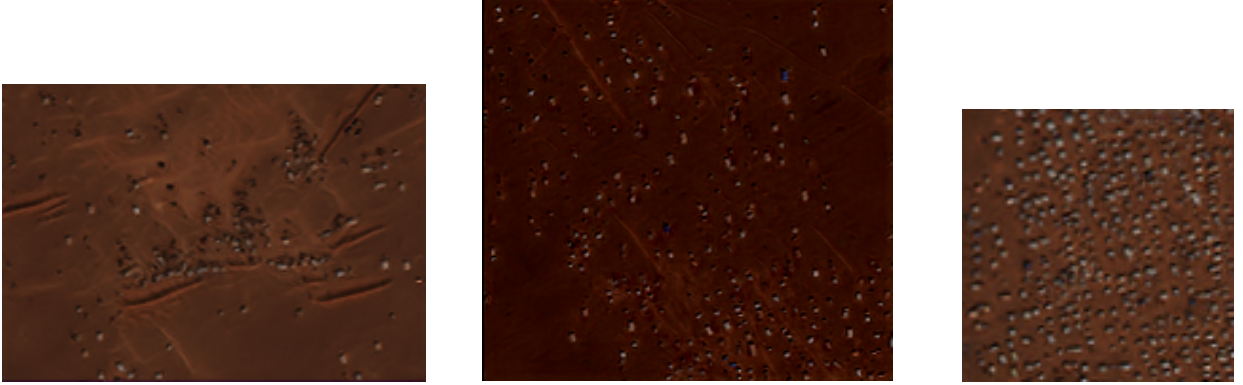
$$\mathcal{L}_{\text{rec}} = \| G_N(z^*) - x_N \|^2 \quad (36)$$

5.2.3 Augmenting Tent Image by SinGAN

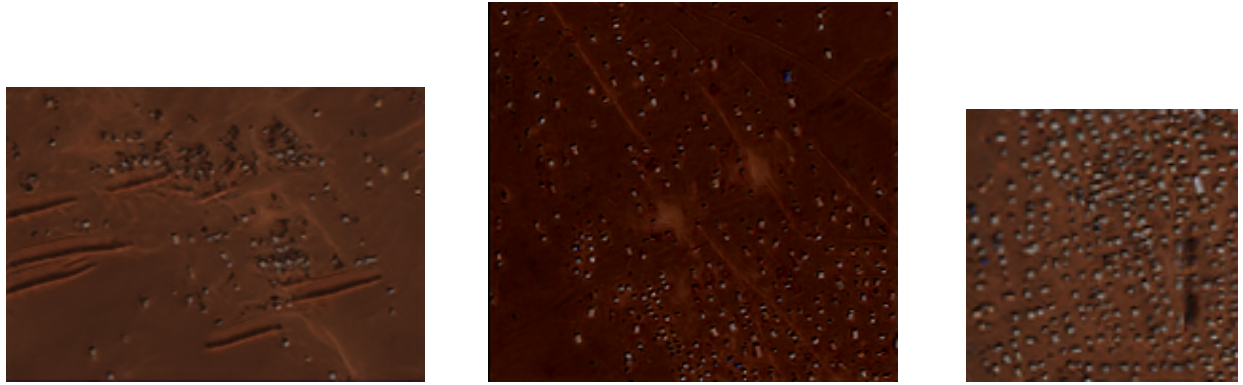
Since SinGAN can generate images from a single image. We use the three training images in section 4.3.5 to generate synthetic tent images. The original training image set is O1 as shown in Fig. 44(a). G1, G2 and G3 are the three sets of the GAN-generated samples, as shown in Fig. 44(b), 44(c), and 44(d).



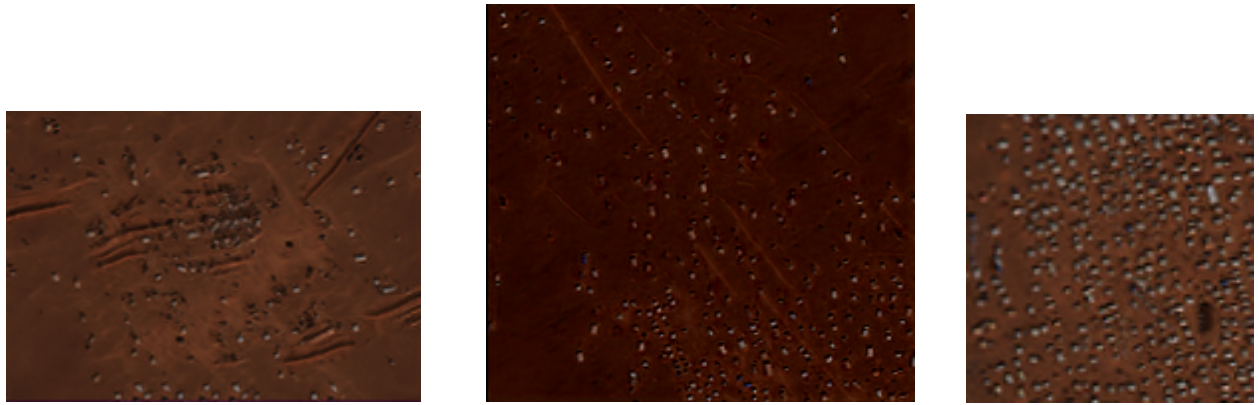
(a) Original Training Samples O1



(b) GAN-generated Dataset G1



(c) GAN-generated Dataset G2



(d) GAN-generated Dataset G3

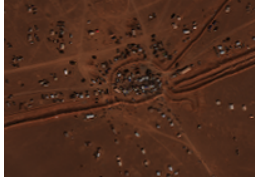
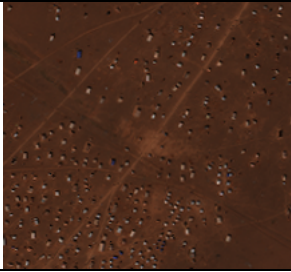

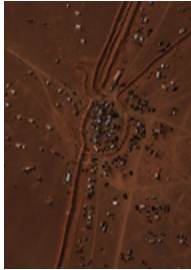
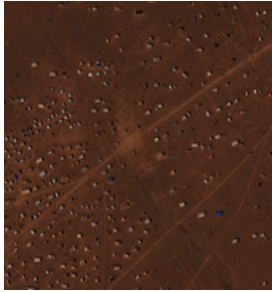
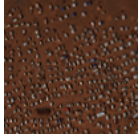
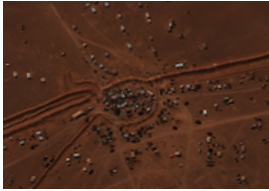
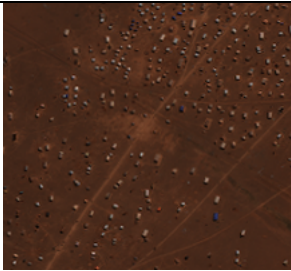

Fig. 44. GAN Generated Samples. (a) Original Training Samples O1, (b)-(d): SinGAN-generated Data G1, G2 and G3

5.2.4 Augmenting Tent Images by Flipping and Rotating

There are many traditional methods of creating new data samples such as flipping, rotation, scaling, and adding noises. In this section, we applied horizontal flipping, vertical flipping and rotating 90 degrees to the right to the original training images to augment data. To flip the image horizontally, we reverse the order of the columns of the image matrix. To flip the image vertically, we reverse the order of the rows of the image matrix. To rotate the image, we transform each row

of the source matrix into the column of the final image. After horizontal flipping, right rotating for 90 degrees and vertical flipping the original training images, the augmented images results are shown in Table 9. We combined all the images in Table 9 as dataset-FR.

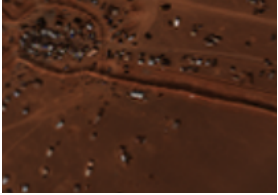
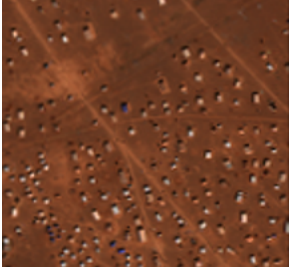

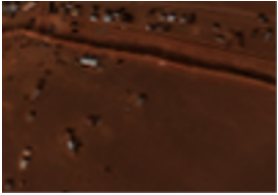
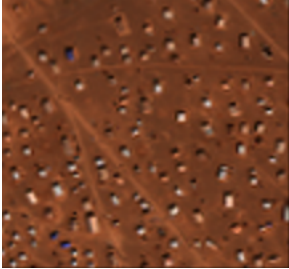
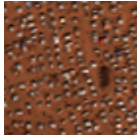
TABLE 9. Datasets-FR by Horizontal Flipping (first row), Right Rotating 90 Degrees (second row) and Vertical Flipping (second row) of the Original Training Images.


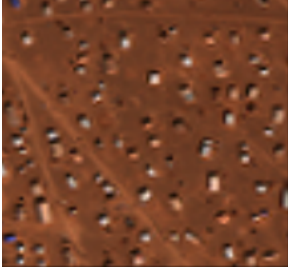

	Training Image -1	Training Image -2	Training Image -3
Horizontal Flipping			
Right Rotating 90 Degrees			
Vertical Flipping			

5.2.5 Augmenting Tent Images by Scaling and Cropping

Scaling images is another common image manipulation technique. Image scaling is usually done by scaling the images by a particular factor and cropping the scaled image to the original size. For example, reducing or increasing the size of an image by a factor of 2 and then cropping the scaled image to the original size. In our experiment, we scaled the original images by factors of 1.5, 2.0, and 2.5, and then we cropped an area with the same sizes of the original images. Some augmented images are shown in Table 10. We combined all scaled images into dataset-SC.

TABLE 10. Image Dataset-SC by Scaling and Cropping the Original Images by Factors of 1.5, 2.0 and 2.5.


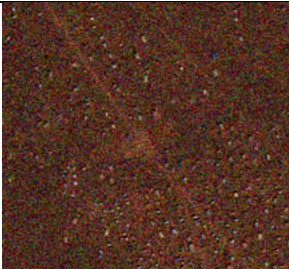
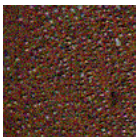
	Training Image -1	Training Image -2	Training Image -3
Scaling 1.5 and crop			
Scaling 2.0 and crop			

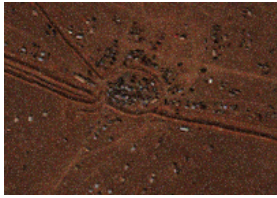
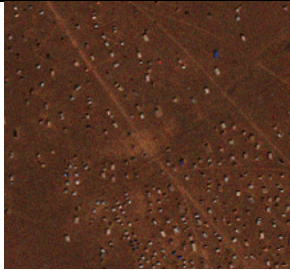
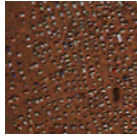
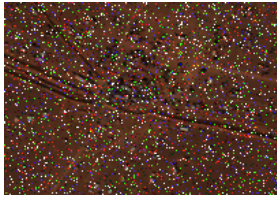
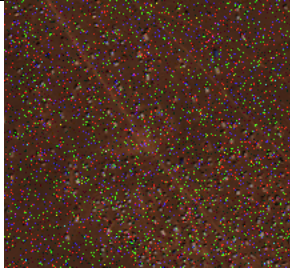
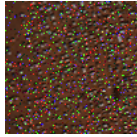
Scaling 2.5 and crop			
----------------------	---	--	---

5.2.6 Augmenting Tent Images by Adding Noise

Noise always presents in digital images during image acquisition, coding, transmission, and processing steps. Adding noises to images is also a classic approach to create new data samples and to improve the robustness of image classification models. In this section, we add Gaussian, Poisson, and salt and pepper noises to the original training datasets and then combined all images with added noises into the new training dataset-N. The augmented images are as shown in Table 11.

TABLE 11. Images Dataset-N by Adding Gaussian, Poisson, and Salt and Pepper Noises.

	Training Image -1	Training Image -2	Training Image -3
Gaussian			

Poisson			
Salt and Pepper			

5.3 Model Training and Experimental Results

In the model training stage, we re-train the FCN model by using different image datasets generated by GAN and the other traditional image translation methods in a two-stage setting.

In the first stage, we re-train the FCN model with the original training dataset O1 and gradually add GAN generated data with different sample sizes. We include one, two, and three times the GAN generated data into the original dataset O1. We also re-train the FCN models with four input sizes: 128×128 , 96×96 , 64×64 , and 32×32 . The way we collect the training blocks in different sizes is the same as Chapter 4, we use stride 2 and sliding windows to collect the training samples.

In the second stage, we re-train the FCN model with different image datasets generated by traditional transformation methods: they are dataset-FP, dataset-SC and dataset-N. A combination dataset dataset-C is also created by including horizontally flipped images from Table 9, scaled

images by a factor of 2 from Table 10, and images samples by adding Gaussian noises from Table 11. The training samples are collected with various input sizes by sliding window.

In addition, a very small dataset dataset-LS which only includes 200 samples from each original training image is included for comparison purposes. In summary, we use the following training schemes to re-train the FCN model:

- FCN_(block_size): These are the baseline models trained by the original training dataset O1, as shown in Fig. 44a).
- FCN_(block_size) _LS: Models trained by 600 samples from the original training samples by randomly selecting 200 samples from each of the original training image.
- FCN_(block_size) _X1: Models trained by adding extra GAN samples X1 to the original training samples. X1 is G1. G1 is as shown in images in Fig. 44b).
- FCN_(block_size) _X2: Models trained by adding extra GAN samples X2 to the original training samples. X2 consists of G1 and G2. G1 and G2 are as shown in images in Fig. 44b) and Fig. 44c), respectively.
- FCN_(block_size) _X3: Models trained by adding extra GAN samples X3 to original training samples. X3 consists of G1, G2, and G3. G1, G2, and G3 are as shown in images in Fig. 44b), Fig. 44c) and Fig. 44d), respectively.
- FCN_(block_size)_FR: Models trained by adding all training samples created from horizontal flipping, vertical flipping, and right rotating 90 degrees of the original training images, as shown in Table 9.
- FCN_(block_size)_SC: Models trained by adding all training samples created by scaling and cropping the original dataset by a factor of 1.5, 2.0, and 2.5 to the original training images, as shown in Table 10.

- FCN_(block_size)_N: Models trained by adding all training samples created by adding Gaussian, Poisson, and salt and pepper noises to the original training samples, as shown in Table 11.
- FCN_(block_size)_C: Models trained by adding extra training samples, which are combined by horizontally flipped images from Table 9, scaled images to the factor of 2 from Table 10 and samples by adding Gaussian noise as shown in Table 11.

We trained the FCN models with the corresponding synthetic datasets and validated the model on the same validation image as used in Chapter 4. We evaluated the performance of the FCN models in terms of precision, recall, F-1 score, precision-recall curve scores, the area under the ROC curve (AUC) and the intersection over union (IoU) scores. Table 12-15 shows the validation performance matrices of the FCN models by adding different synthetic datasets with different input sizes.

TABLE 12. Validation Performance Matrices by Synthetic Datasets with Input Size of 128x128

	Average Accuracy	Precision	Recall	F1-Score	AP (P-R)	AUC (ROC)	IoU	Sample Size
FCN_128_LS	95.62%	21.18%	0.29%	0.57%	4.39%	50.12%	47.95%	600
FCN_128	95.67%	50.59%	43.99%	47.06%	24.71%	71.01%	63.47%	1628
FCN_128_X1	95.75%	51.26%	44.25%	47.50%	25.11%	71.17%	63.40%	2849
FCN_128_X2	95.72%	50.81%	44.93%	47.69%	25.22%	71.48%	63.47%	4070
FCN_128_X3	95.81%	49.87%	48.20%	49.02%	26.28%	73.00%	64.01%	5235
FCN_128_FR	95.83%	52.71%	38.72%	44.65%	23.07%	68.57%	62.25%	5235
FCN_128_SC	95.97%	54.79%	41.30%	47.10%	25.18%	69.88%	63.35%	5235
FCN_128_N	95.81%	52.05%	43.72%	47.53%	25.20%	70.95%	63.45%	5235
FCN_128_C	95.86%	53.05%	41.16%	46.35%	24.39%	69.75%	62.98%	5235

TABLE 13. Validation Performance Matrices by Synthetic Datasets with Input Size of
96x96

	Average Accuracy	Precision	Recall	F1-Score	AP (P-R)	AUC (ROC)	IoU	Sample Size
FCN_96_LS	93.07%	8.75%	6.31%	7.34%	4.62%	51.66%	48.43%	600
FCN_96	95.95%	54.30%	47.06%	50.42%	27.87%	72.63%	64.98%	3180
FCN_96_X1	95.91%	53.07%	50.06%	51.52%	28.73%	74.02%	65.26%	5684
FCN_96_X2	95.58%	49.14%	49.93%	49.53%	26.71%	73.79%	64.20%	8312
FCN_96_X3	95.97%	53.62%	54.08%	53.85%	30.99%	75.98%	66.36%	10851
FCN_96_FR	95.87%	53.17%	41.90%	46.97%	24.80%	70.11%	63.20%	10851
FCN_96_SC	95.91%	53.23%	47.92%	50.44%	27.77%	73.01%	64.77%	10851
FCN_96_N	95.92%	53.44%	47.99%	50.57%	27.90%	73.04%	64.84%	10851
FCN_96_C	95.92%	53.84%	42.57%	47.55%	25.42%	70.46%	63.52%	10851

TABLE 14. Validation Performance Matrices by Synthetic Datasets with Input Size of
64x64

	Average Accuracy	Precision	Recall	F1-Score	AP (P-R)	AUC (ROC)	IoU	Sample Size
FCN_64_LS	94.95%	8.83%	1.74%	2.92%	4.42%	50.46%	48.21%	600
FCN_64	96.23%	58.27%	48.71%	53.06%	30.63%	73.55%	66.43%	5244
FCN_64_X1	96.09%	55.19%	53.61%	54.39%	31.60%	75.82%	66.68%	9884
FCN_64_X2	96.05%	54.58%	53.49%	54.02%	31.21%	75.73%	66.48%	14291
FCN_64_X3	96.24%	57.03%	54.78%	55.89%	33.21%	76.46%	67.47%	18715
FCN_64_FR	96.03%	55.47%	43.84%	48.97%	26.75%	71.12%	64.19%	18715
FCN_64_SC	96.07%	55.61%	46.69%	50.76%	28.28%	72.50%	64.70%	18715
FCN_64_N	96.11%	55.88%	49.93%	52.74%	30.08%	74.07%	65.92%	18715
FCN_64_C	96.09%	56.38%	43.92%	49.37%	27.20%	71.19%	64.40%	18715

TABLE 15. Validation Performance Matrices by Synthetic Datasets with Input Size of 32x32

	Average Accuracy	Precision	Recall	F1-Score	AP (P-R)	AUC (ROC)	IoU	Sample Size
FCN_32_LS	90.46%	7.38%	10.37%	8.63%	4.66%	52.23%	47.46%	600
FCN_32	96.54%	62.35%	52.88%	57.22%	35.03%	75.71%	68.44%	7820
FCN_32_X1	96.43%	59.52%	55.68%	57.54%	35.07%	76.98%	68.36%	15236
FCN_32_X2	96.42%	59.42%	55.62%	57.46%	34.96%	76.95%	68.32%	21879
FCN_32_X3	96.44%	59.43%	56.86%	58.12%	34.97%	77.55%	68.65%	28968
FCN_32_FR	96.26%	58.50%	47.60%	52.49%	30.12%	73.04%	65.88%	28968
FCN_96_SC	95.91%	53.23%	47.92%	50.44%	27.78%	73.01%	64.77%	28968
FCN_96_N	95.92%	53.44%	47.99%	50.57%	27.90%	73.04%	64.84%	28968
FCN_96_C	95.92%	53.84%	42.57%	47.55%	25.42%	70.46%	63.52%	28968

5.4 Conclusions

We compared the experiment results with additional GAN-generated samples for training to the results with traditional data generated by augmentation methods in terms of average accuracy, precision, recall, F-1 score, precision-recall, the AUC score of the ROC curve and the IoU scores. Our experimental results show that with additional GAN training samples, AUC scores of ROC curve are improved between 0.6-3.3%, recalls are improved between 1.2-7.2%, and the improvements of average accuracies and precisions are very limited.

The experimental results show that the F-1 scores are improved between 0.9-3.4%. The precision-recall scores are improved between 0.04-3.1%. For semantic segmentation problems, IoU is the most important metric. By adding GAN samples, the average IoU scores are improved between 0.2-1.5%.

The largest improvement of the FCN model is found when the input patch size is large and the available training samples are small such as FCN-128. The best performance of FCN is found

with input size of 32x32 by the largest added GAN samples dataset (X3). The IoU score is 68.65%, which is the highest among all trained models.

We found that the traditional image manipulation and transformation methods do not help improving the deep model training. There might be many causes, but the most important reason is that the traditional image transformation methods do not learn from data. As a comparison, GAN method does not only increase data sample sizes while keeping the characteristics of the objects to detect, but also learns from data distribution to generate new diversified data samples preserving the context information in the original images.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The overall goal of this dissertation is to investigate the framework and workflow of deep learning model and apply them to resolve practical remote sensing image processing problems such as image classification, object detection, and semantic segmentations. There are three research topics are proposed in this dissertation.

The first topic is using CNN model to tackle the hyperspectral image classification problem. In this research, we present a simple four layers deep convolutional neural network (CNN) model for soil detection by using the combination of eighty synthetic hyperspectral bands and its original eight multispectral bands. We applied the CNN model onto a set of high-resolution data created by pan-sharpening the original multispectral bands and its synthetic hyperspectral bands. By using the pan-sharpened synthetic hyperspectral bands, the performance of the CNN model for soil detection has been significantly improved [31].

The second topic in this dissertation is to accurately extract the refugee tents near the Syria-Jordan border by using deep learning models. In this research, we present an FCN model to tackle the small ground objects detection problem and applied it to the refugee tents extraction problem. In this research we also compared the proposed approach with the traditional spectral angle mapper (SAM) method, CNNs models, and the Mask R- CNN model. The experimental results show that the FCN model significantly improved the overall performance than the other models [32].

The third topic in this dissertation is data augmentations by applying the GAN network. In this research, we used SinGAN, which is a hierarchically structured deep GAN model. SinGAN is

able to generate data from one single natural image. we compared the GAN data augmentations results with the other traditional image transformation methods such as flipping, rotating, scaling, and adding noises. The experimental results show that the GAN generated samples improved the IoU score of the FCN model by 0.2-1.5%. We also find that the models with finer image input size and larger training samples performs much better than the model with larger image input size and smaller training sample. The traditional image transformation methods do not help in this case.

6.2 Future Work

Even though our proposed work and results have reached satisfactory results, for each of the proposed topics there are still improvements that could be made. For the first research topic in this dissertation, the hyperspectral RS image data provides us very rich spectral and spatial information. In the future, we could apply a very large deep model such as deep residual net ResNet-50 or ResNet-100 to continue to improve the classification results. On the other hand, the computational complexity to process these hyperspectral data is very high, it will be very useful if the spectral dimensionality could be reduced but still keeping the same information from the hyperspectral bands. For the second research topic in this dissertation, the FCN model has reached very good classification results in detecting small ground objects. However, in practical, the RS images are easily impacted by climate and weather such as snow and cloud. In my future work, further research will be focused on eliminating the impact of weather changes in the RS images classification. For the last research topic in this dissertation, the GAN samples improved the performance of the FCN model, however, to create the ground truth mask is laborious. In the future, I will focus on developing an end-to-end deep model by integrating GAN as a part of the

classification model to improve the classification performance without too much interference in the middle of the training process.

REFERENCES

1. Padwick, C., et al. *WorldView-2 pan-sharpening*. in *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA*. 2010.
2. Drusch, M., et al., *Sentinel-2: ESA's optical high-resolution mission for GMES operational services*. Remote sensing of Environment, 2012. **120**: p. 25-36.
3. Thai, B. and G. Healey. *Invariant subpixel target identification in hyperspectral imagery*. in *Algorithms for multispectral and hyperspectral imagery V*. 1999. International Society for Optics and Photonics.
4. Heras, D.B., et al. *Towards real-time hyperspectral image processing, a GP-GPU implementation of target identification*. in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*. 2011. IEEE.
5. Nasrabadi, N.M., *Hyperspectral target detection: An overview of current and future challenges*. IEEE Signal Processing Magazine, 2013. **31**(1): p. 34-44.
6. Stein, D.W., et al., *Anomaly detection from hyperspectral imagery*. IEEE signal processing magazine, 2002. **19**(1): p. 58-69.
7. Kwon, H., S.Z. Der, and N.M. Nasrabadi, *Adaptive anomaly detection using subspace separation for hyperspectral imagery*. OPTICAL ENGINEERING-BELLINGHAM-INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING-, 2003. **42**(11): p. 3342-3351.
8. Pal, M., *Random forest classifier for remote sensing classification*. International journal of remote sensing, 2005. **26**(1): p. 217-222.

9. Belgiu, M. and L. Drăguț, *Random forest in remote sensing: A review of applications and future directions*. ISPRS Journal of Photogrammetry and Remote Sensing, 2016. **114**: p. 24-31.
10. Otukey, J.R. and T. Blaschke, *Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms*. International Journal of Applied Earth Observation and Geoinformation, 2010. **12**: p. S27-S31.
11. Gualtieri, J.A. and R.F. Crompt. *Support vector machines for hyperspectral remote sensing classification*. in *27th AIPR Workshop: Advances in Computer-Assisted Recognition*. 1999. International Society for Optics and Photonics.
12. Heumann, B.W., *An object-based classification of mangroves using a hybrid decision tree—Support vector machine approach*. Remote Sensing, 2011. **3**(11): p. 2440-2460.
13. Maulik, U. and D. Chakraborty, *Remote Sensing Image Classification: A survey of support-vector-machine-based advanced techniques*. IEEE Geoscience and Remote Sensing Magazine, 2017. **5**(1): p. 33-52.
14. Oommen, T., et al., *An objective analysis of support vector machine based classification for remote sensing*. Mathematical geosciences, 2008. **40**(4): p. 409-424.
15. Tan, K. and P.-J. Du, *Hyperspectral remote sensing image classification based on support vector machine*. Journal of Infrared and Millimeter Waves, 2008. **27**(2): p. 123-128.
16. Wang, M., et al., *Remote sensing image classification based on the optimal support vector machine and modified binary coded ant colony optimization algorithm*. Information Sciences, 2017. **402**: p. 50-68.

17. Moustakidis, S., et al., *SVM-based fuzzy decision trees for classification of high spatial resolution remote sensing images*. IEEE Transactions on Geoscience and Remote Sensing, 2011. **50**(1): p. 149-169.
18. Wang, L., W. Sousa, and P. Gong, *Integration of object-based and pixel-based classification for mapping mangroves with IKONOS imagery*. International Journal of Remote Sensing, 2004. **25**(24): p. 5655-5668.
19. Weih, R.C. and N.D. Riggan, *Object-based classification vs. pixel-based classification: Comparative importance of multi-resolution imagery*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2010. **38**(4): p. C7.
20. Bárdossy, A. and L. Samaniego, *Fuzzy rule-based classification of remotely sensed imagery*. IEEE transactions on geoscience and remote sensing, 2002. **40**(2): p. 362-374.
21. Lucas, R., et al., *Rule-based classification of multi-temporal satellite imagery for habitat and agricultural land cover mapping*. ISPRS Journal of photogrammetry and remote sensing, 2007. **62**(3): p. 165-185.
22. Kaur, B. and A. Garg. *Mathematical morphological edge detection for remote sensing images*. in *2011 3rd International Conference on Electronics Computer Technology*. 2011. IEEE.
23. Soille, P. and M. Pesaresi, *Advances in mathematical morphology applied to geoscience and remote sensing*. IEEE Transactions on Geoscience and Remote Sensing, 2002. **40**(9): p. 2042-2055.
24. Garcia-Garcia, A., et al., *A review on deep learning techniques applied to semantic segmentation*. arXiv preprint arXiv:1704.06857, 2017.

25. Guo, Y., et al., *Deep learning for visual understanding: A review*. Neurocomputing, 2016. **187**: p. 27-48.
26. Ioannidou, A., et al., *Deep learning advances in computer vision with 3d data: A survey*. ACM Computing Surveys (CSUR), 2017. **50**(2): p. 1-38.
27. Voulodimos, A., et al., *Deep learning for computer vision: A brief review*. Computational intelligence and neuroscience, 2018. **2018**.
28. Plaza, A., et al., *A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles*. Pattern Recognition, 2004. **37**(6): p. 1097-1116.
29. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
30. Torrey, L. and J. Shavlik, *Transfer learning*, in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. 2010, IGI global. p. 242-264.
31. Lu, Y., et al. *Deep learning with synthetic hyperspectral images for improved soil detection in multispectral imagery*. in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2018. IEEE.
32. Lu, Y., et al., *Deep Learning for Effective Refugee Tent Extraction Near Syria-Jordan Border*. IEEE Geoscience and Remote Sensing Letters, 2020.
33. Lu, Y. and J. Li. *Generative adversarial network for improving deep learning based malware classification*. in *2019 Winter Simulation Conference (WSC)*. 2019. IEEE.
34. Burks, R., et al. *Data Augmentation with Generative Models for Improved Malware Detection: A Comparative Study*. in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2019. IEEE.

35. Avery, T.E. and G.L. Berlin, *Fundamentals of remote sensing and airphoto interpretation*. 1992: Macmillan.
36. Baumann, P.R., *History of remote sensing, satellite imagery, part II*. Last modified, 2009.
37. Tucker, C. and P. Sellers, *Satellite remote sensing of primary production*. International journal of remote sensing, 1986. **7**(11): p. 1395-1416.
38. Hardeberg, J.Y., et al., *Multispectral image acquisition and simulation of illuminant changes*, in *Colour imaging: vision and technology*. 1999, Citeseer.
39. Palubinskas, G., P. Reinartz, and R. Bamler, *Image acquisition geometry analysis for the fusion of optical and radar remote sensing data*. International Journal of Image and Data Fusion, 2010. **1**(3): p. 271-282.
40. Brivio, P., et al., *Integration of remote sensing data and GIS for accurate mapping of flooded areas*. International Journal of Remote Sensing, 2002. **23**(3): p. 429-441.
41. Weng, Q., *Land use change analysis in the Zhujiang Delta of China using satellite remote sensing, GIS and stochastic modelling*. Journal of environmental management, 2002. **64**(3): p. 273-284.
42. Pradhan, B., *Groundwater potential zonation for basaltic watersheds using satellite remote sensing data and GIS techniques*. Central European Journal of Geosciences, 2009. **1**(1): p. 120-129.
43. Lillesand, T., R. Kiefer, and J. Chipman, *Remote sensing and image processing*. 1987, John Wiley & Sons, New York.
44. Williams, D.L., S. Goward, and T. Arvidson, *Landsat*. Photogrammetric Engineering & Remote Sensing, 2006. **72**(10): p. 1171-1178.

45. Vane, G., et al., *The airborne visible/infrared imaging spectrometer (AVIRIS)*. Remote sensing of environment, 1993. **44**(2-3): p. 127-143.
46. Hall, D.K., et al., *MODIS snow-cover products*. Remote sensing of Environment, 2002. **83**(1-2): p. 181-194.
47. Abbas, A., et al., *K-Means and ISODATA clustering algorithms for landcover classification using remote sensing*. Sindh University Research Journal-SURJ (Science Series), 2016. **48**(2).
48. Deng, J., et al., *PCA - based land - use change detection and analysis using multitemporal and multisensor satellite data*. International Journal of Remote Sensing, 2008. **29**(16): p. 4823-4838.
49. Zheng, Y., E.A. Essock, and B.C. Hansen. *An advanced image fusion algorithm based on wavelet transform: incorporation with PCA and morphological processing*. in *Image processing: algorithms and systems III*. 2004. International Society for Optics and Photonics.
50. Kaewpijit, S., J. Le Moigne, and T. El-Ghazawi. *A wavelet-based PCA reduction for hyperspectral imagery*. in *IEEE International Geoscience and Remote Sensing Symposium*. 2002. IEEE.
51. Li, B., H. Zhao, and Z. Lv. *Parallel ISODATA clustering of remote sensing images based on MapReduce*. in *2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. 2010. IEEE.
52. Sun, J., et al., *Automatic remotely sensed image classification in a grid environment based on the maximum likelihood method*. Mathematical and Computer Modelling, 2013. **58**(3-4): p. 573-581.

53. Civco, D.L., *Artificial neural networks for land-cover classification and mapping*. International journal of geographical information science, 1993. **7**(2): p. 173-186.
54. Mas, J.F. and J.J. Flores, *The application of artificial neural networks to the analysis of remotely sensed data*. International Journal of Remote Sensing, 2008. **29**(3): p. 617-663.
55. Qiu, F. and J. Jensen, *Opening the black box of neural networks for remote sensing image classification*. International Journal of Remote Sensing, 2004. **25**(9): p. 1749-1768.
56. Friedl, M.A. and C.E. Brodley, *Decision tree classification of land cover from remotely sensed data*. Remote sensing of environment, 1997. **61**(3): p. 399-409.
57. Yang, C.-C., et al., *Application of decision tree technology for image classification using remote sensing data*. Agricultural Systems, 2003. **76**(3): p. 1101-1117.
58. Shackelford, A.K. and C.H. Davis, *A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas*. IEEE Transactions on GeoScience and Remote sensing, 2003. **41**(10): p. 2354-2363.
59. Somers, B., et al., *Endmember variability in spectral mixture analysis: A review*. Remote Sensing of Environment, 2011. **115**(7): p. 1603-1616.
60. Wu, C. and A.T. Murray, *Estimating impervious surface distribution by spectral mixture analysis*. Remote sensing of Environment, 2003. **84**(4): p. 493-505.
61. Yildirim, I., O.K. Ersoy, and B. Yazgan, *Improvement of classification accuracy in remote sensing using morphological filter*. Advances in Space Research, 2005. **36**(5): p. 1003-1006.
62. Walter, V., *Object-based classification of remote sensing data for change detection*. ISPRS Journal of photogrammetry and remote sensing, 2004. **58**(3-4): p. 225-238.

63. Yu, Q., et al., *Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery*. Photogrammetric Engineering & Remote Sensing, 2006. **72**(7): p. 799-811.
64. Zhou, W., A. Troy, and M. Grove, *Object-based land cover classification and change analysis in the Baltimore metropolitan area using multitemporal high resolution remote sensing data*. Sensors, 2008. **8**(3): p. 1613-1636.
65. Lu, D., S. Hetrick, and E. Moran, *Impervious surface mapping with Quickbird imagery*. International journal of remote sensing, 2011. **32**(9): p. 2519-2533.
66. Blaschke, T., *Object based image analysis for remote sensing*. ISPRS journal of photogrammetry and remote sensing, 2010. **65**(1): p. 2-16.
67. Liu, D. and F. Xia, *Assessing object-based classification: advantages and limitations*. Remote Sensing Letters, 2010. **1**(4): p. 187-194.
68. Shekhar, S., et al., *Spatial contextual classification and prediction models for mining geospatial data*. IEEE Transactions on Multimedia, 2002. **4**(2): p. 174-188.
69. Moser, G., S.B. Serpico, and J.A. Benediktsson, *Land-cover mapping by Markov modeling of spatial-contextual information in very-high-resolution remote sensing images*. Proceedings of the IEEE, 2012. **101**(3): p. 631-651.
70. Hinton, G.E., S. Osindero, and Y.-W. Teh, *A fast learning algorithm for deep belief nets*. Neural computation, 2006. **18**(7): p. 1527-1554.
71. Agarwal, N., et al. *Finding approximate local minima faster than gradient descent*. in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017.
72. Kawaguchi, K. *Deep learning without poor local minima*. in *Advances in neural information processing systems*. 2016.

73. Kawaguchi, K. and L. Kaelbling. *Elimination of all bad local minima in deep learning*. in *International Conference on Artificial Intelligence and Statistics*. 2020.
74. Hanin, B. *Which neural net architectures give rise to exploding and vanishing gradients?* in *Advances in Neural Information Processing Systems*. 2018.
75. Hochreiter, S., *The vanishing gradient problem during learning recurrent neural nets and problem solutions*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998. **6**(02): p. 107-116.
76. Henkelman, G., B.P. Uberuaga, and H. Jónsson, *A climbing image nudged elastic band method for finding saddle points and minimum energy paths*. The Journal of chemical physics, 2000. **113**(22): p. 9901-9904.
77. Pascanu, R., et al., *On the saddle point problem for non-convex optimization*. arXiv preprint arXiv:1405.4604, 2014.
78. Smolensky, P., *Information processing in dynamical systems: Foundations of harmony theory*. 1986, Colorado Univ at Boulder Dept of Computer Science.
79. Freund, Y. and D. Haussler. *Unsupervised learning of distributions on binary vectors using two layer networks*. in *Advances in neural information processing systems*. 1992.
80. Hinton, G.E., *Training products of experts by minimizing contrastive divergence*. Neural computation, 2002. **14**(8): p. 1771-1800.
81. Bottou, L., *Large-scale machine learning with stochastic gradient descent*, in *Proceedings of COMPSTAT'2010*. 2010, Springer. p. 177-186.
82. Bottou, L., *Stochastic gradient learning in neural networks*. Proceedings of Neuro-Nimes, 1991. **91**(8): p. 12.

83. Tieleman, T. and G. Hinton. *Using fast weights to improve persistent contrastive divergence*. in *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009.
84. Hinton, G.E. and R.R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*. science, 2006. **313**(5786): p. 504-507.
85. Hubel, D.H. and T.N. Wiesel, *Receptive fields and functional architecture of monkey striate cortex*. The Journal of physiology, 1968. **195**(1): p. 215-243.
86. LeCun, Y., et al., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
87. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.
88. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
89. Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *ICML*. 2010.
90. Pan, S.J. and Q. Yang, *A survey on transfer learning*. IEEE Transactions on knowledge and data engineering, 2009. **22**(10): p. 1345-1359.
91. Guerette, R.T. and R.V. Clarke, *Border enforcement, organized crime, and deaths of smuggled migrants on the United States–Mexico Border*. European Journal on Criminal Policy and Research, 2005. **11**(2): p. 159-174.
92. Jusionyte, I., *The wall and the wash: Security, infrastructure and rescue on the US - Mexico border*. Anthropology Today, 2017. **33**(3): p. 13-16.

93. Keim, S.M., et al., *Wilderness rescue and border enforcement along the Arizona Mexico border—the Border Patrol Search, Trauma and Rescue Unit*. Wilderness & environmental medicine, 2009. **20**(1): p. 39-42.
94. Lichtenwald, T.G. and F.S. Perri, *Terrorist Use of Smuggling Tunnels*. International Journal of Criminology and Sociology, 2013. **2**: p. 210-226.
95. Sorrensens, C., *Making the subterranean visible: security, tunnels, and the United States–Mexico border*. Geographical Review, 2014. **104**(3): p. 328-345.
96. Cao, L., et al., *Monitoring cross-border trails using airborne digital multispectral imagery and interactive image analysis techniques*. Geocarto International, 2007. **22**(2): p. 107-125.
97. Srinivasan, S., et al. *Airborne traffic surveillance systems: video surveillance of highway traffic*. in *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. 2004.
98. Stich, E.J. *Geo-pointing and threat location techniques for airborne border surveillance*. in *2013 IEEE International Conference on Technologies for Homeland Security (HST)*. 2013. IEEE.
99. Manolakis, D., D. Marden, and G.A. Shaw, *Hyperspectral image processing for automatic target detection applications*. Lincoln laboratory journal, 2003. **14**(1): p. 79-116.
100. Hänsch, R. and O. Hellwich, *Fusion of Multispectral LiDAR, Hyperspectral, and RGB Data for Urban Land Cover Classification*. IEEE Geoscience and Remote Sensing Letters, 2020.
101. Suresh, S. and S. Lal, *A metaheuristic framework based automated Spatial-Spectral graph for land cover classification from multispectral and hyperspectral satellite images*. Infrared Physics & Technology, 2020. **105**: p. 103172.

102. Sudharshan, V., et al., *Object detection routine for material streams combining RGB and hyperspectral reflectance data based on Guided Object Localization*. IEEE Sensors Journal, 2020.
103. Jahan, F., et al., *Inverse Coefficient of Variation Feature and Multilevel Fusion Technique for Hyperspectral and LiDAR Data Classification*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020. **13**: p. 367-381.
104. Dalla Mura, M., et al., *Extended profiles with morphological attribute filters for the analysis of hyperspectral data*. International Journal of Remote Sensing, 2010. **31**(22): p. 5975-5991.
105. Quesada-Barriuso, P., F. Argüello, and D.B. Heras, *Spectral-spatial classification of hyperspectral images using wavelets and extended morphological profiles*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2014. **7**(4): p. 1177-1185.
106. Xia, J., et al., *Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles*. IEEE Transactions on Geoscience and Remote Sensing, 2015. **53**(9): p. 4768-4786.
107. Dao, M., et al. *A joint sparsity approach to tunnel activity monitoring using high resolution satellite images*. in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. 2017. IEEE.
108. H.Saad, M.S., *Misery 'Every Day, Every Hour' in Syrian Camp. And Now, It's Grown Critical*, in *The New York Times*. 2018.
109. UNOSAT. *Shelter Density Map at Rukban Border Crossing: Syria- Jordan Border*. 2018.

110. Girshick, R., et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
111. Girshick, R. *Fast r-cnn*. in *Proceedings of the IEEE international conference on computer vision*. 2015.
112. Ren, S., et al. *Faster r-cnn: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*. 2015.
113. He, K., et al. *Mask r-cnn*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
114. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
115. Yuhas, R.H., A.F. Goetz, and J.W. Boardman, *Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm*. 1992.
116. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting*. The journal of machine learning research, 2014. **15**(1): p. 1929-1958.
117. Goodfellow, I.J., et al., *Generative adversarial networks*. *arXiv e-prints (2014)*. arXiv preprint arXiv:1406.2661, 2014.
118. Wang, X., et al. *Esrgan: Enhanced super-resolution generative adversarial networks*. in *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
119. Yi, Z., et al. *Dualgan: Unsupervised dual learning for image-to-image translation*. in *Proceedings of the IEEE international conference on computer vision*. 2017.

120. Karras, T., S. Laine, and T. Aila. *A style-based generator architecture for generative adversarial networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019.
121. Wu, H., et al. *Gp-gan: Towards realistic high-resolution image blending*. in *Proceedings of the 27th ACM International Conference on Multimedia*. 2019.
122. Bao, J., et al. *CVAE-GAN: fine-grained image generation through asymmetric training*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
123. Ma, D., P. Tang, and L. Zhao, *SiftingGAN: Generating and sifting labeled samples to improve the remote sensing image scene classification baseline in vitro*. IEEE Geoscience and Remote Sensing Letters, 2019. **16**(7): p. 1046-1050.
124. Lin, D., et al., *MARTA GANs: Unsupervised representation learning for remote sensing image classification*. IEEE Geoscience and Remote Sensing Letters, 2017. **14**(11): p. 2092-2096.
125. Yan, Y., Z. Tan, and N. Su, *A data augmentation strategy based on simulated samples for ship detection in rgb remote sensing images*. ISPRS International Journal of Geo-Information, 2019. **8**(6): p. 276.
126. Zhang, M., et al. *Data augmentation method of SAR image dataset*. in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. 2018. IEEE.
127. Shaham, T.R., T. Dekel, and T. Michaeli. *Singan: Learning a generative model from a single natural image*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

VITA

Yan Lu

Department of Computational Modeling and Simulation Engineering

Old Dominion University

Norfolk, VA 23529

Yan Lu received her B.S. and M.S. in Computer Science from Beijing Jiaotong University, China in 2004 and Virginia Commonwealth University, Virginia in 2009 respectively. She received her M.S. in Circuit and System from Chinese Academy of Sciences in 2007. She started pursuing Ph.D. in modeling and simulation at Old Dominion University in 2012.

LIST OF PUBLICATIONS

L. Yan, C. Kwan, and J. Li. “Deep Learning for Effective Refugee Tent Detection near Syrian Jordan Border”, IEEE Geoscience and Remote Sensing Letters. 2020, Jun 24.

K. Chiman, B. Ayhan, B. Budavari, **Y. Lu**, D. Perez, J. Li, S. Bernabe, and A. Plaza. "Deep Learning for Land Cover Classification Using Only a Few Bands." Remote Sensing 12, no. 12 (2020): 2000.

A. Bulent, C. Kwan, Bence. B, Liyun. K; **Y. Lu**, Daniel. P, J. Li, D. Skarlatos, M. Vlachos, “Vegetation Detection Using Deep Learning and Conventional Methods,” *Remote Sensing*. **2020**, 12, 2502.

L. Yan and J. Li. “Malware Classification Using Deep Residual Network with Non-SoftMax Classifier”, Journal of Simulation Engineering, under review.

B. Roland, K. Islam, **Y. Lu**, and J. Li. “Data Augmentation with Generative Models for Improved Malware Detection: A Comparative Study”, In IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference. New York, Oct. 2019.

L. Yan and J. Li. “Generative Adversarial Network for Improving Deep Learning Based Malware Classification”, Winter Simulation Conference, 2019.

L. Yan, J. Graham and J. Li. “Deep Learning Based Malware Classification Using Deep Residual Network”. In Proceedings of the 2019 Modeling, Simulation and Visualization Student Capstone Conference, Suffolk, Virginia, April 18th, 2019. (Gene Newman Award)

L. Yan, D. Perez, M. Dao, C. Kwan, and J. Li. "Deep learning with synthetic hyperspectral images for improved soil detection in multispectral imagery." In Proceedings of the IEEE Ubiquitous

Computing, Electronics & Mobile Communication Conference, New York, NY, USA, pp. 8-10. 2018.

P. Daniel, **Y. Lu**, C. Kwan, Y. Shen, K. Koperski and J. Li. "Combining satellite images with feature indices for improved change detection." In IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference. 2018.

M. Bharat and **Y. Lu**. "Attack tolerant big data file system." ACM Sigmetrics Big Data Analytics Workshop. 2013.

L. Huaxiang, **Y. Lu**, Z. Tang, S. Wang. "Research on System-Level Dynamic Power Management Using Learning Neural Networks", The 2006 International Conference on Artificial Intelligence proceedings, June 26-29, 2006, Las Vegas, US

L. Huaxiang, **Y. Lu**, Z. Tang, and S. Wang. "SOC Dynamic Power Management Using Artificial Neural Network", Lecture Notes in Computer Science, Pages 555-564, Volume 4221/2006.