

DEEP CONVOLUTIONAL NEURAL NETWORKS FOR CLASSIFICATION OF
FUSED HYPERSPECTRAL AND LIDAR DATA

BY

SAURABH MORCHHALE

A Thesis Submitted to the Graduate Faculty of
WAKE FOREST UNIVERSITY GRADUATE SCHOOL OF ARTS AND SCIENCES
in Partial Fulfillment of the Requirements

for the Degree of
MASTER OF SCIENCE
Computer Science

May, 2016

Winston-Salem, North Carolina

Approved By:

V. Paúl Pauca, Ph.D., Advisor

Todd C. Torgersen, Ph.D., Chair

Robert J. Plemmons, Ph.D.

Acknowledgements

I would like to thank the Department of Computer Science at Wake Forest University and appreciate all of the resources and support that the department has provided me with in order to complete my thesis. I want to thank Dr. Plemmons and Dr. Torgersen for serving on my committee. I appreciate their input and the time they had taken to help me complete this thesis.

Most importantly, I thank Dr. Pauca for being a great mentor throughout my master's program and thesis work. The encouragement, guidance and efforts put in by him have played a crucial role in shaping this work.

I would also like to thank Dr. Turkett for helping me with my open-ended questions for this work. I would also like to thank the professors in the department who have been very helpful and encouraging throughout my time at Wake Forest.

I am thankful to my parents and friends for their support and direction not only in my studies, but in life.

I would also like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the data sets used in this study, and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest. I also thank P. Gader, A. Zare, R. Close, J. Aitken, G. Tuell, the University of Florida, and the University of Missouri for sharing the "MUUFL Gulfport Hyperspectral and LiDAR Data Collection" acquired with NGA funding.

This work was supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant no. FA9550-15-1-0286.

Table of Contents

List of Figures	v
List of Tables	vii
Abstract	viii
Chapter 1 Introduction	1
Chapter 2 Hyperspectral and LiDAR Images	4
2.1 Hyperspectral Images and Spectral Bands	4
2.1.1 Spectral Signatures	5
2.2 LiDAR Images	8
2.2.1 Elevation Models	9
2.2.2 LiDAR Intensity	9
2.3 Datasets Used in this Thesis	11
2.3.1 2013_IEEE_GRSS_DF_Constest Dataset	11
2.3.2 MUUFL Gulfport Dataset	11
Chapter 3 Hyperspectral and LiDAR Fusion	12
3.1 Fusion types	12
3.1.1 Pixel/Data Level Fusion	12
3.1.2 Feature Level Fusion	13
3.1.3 Decision Level Fusion	13
3.2 Proposed Pixel Level Fusion	13
Chapter 4 Convolutional Neural Networks	16
4.1 CNN Architecture	16
4.1.1 Input Layer	17
4.1.2 Convolutional Layer	17
4.1.3 Max-Pooling Layer	17
4.1.4 Fully-Connected Layer	18
4.1.5 Output Layer	18
4.2 Backpropagation	19
4.3 Hyper-Parameters	20

Chapter 5	Experimental Results	23
5.1	Experimental Setup	23
5.1.1	Datasets	23
5.1.2	Implementation	23
5.1.3	Computing Environment	23
5.2	Experiments	24
5.2.1	Experiment 1: Repetition of LiDAR information	24
5.2.2	Experiment 2: Classification on HSI vs. Fused HSI-LiDAR	26
5.2.3	Experiment 3: Uncertainties in the Training Data	29
5.3	Additional Results	30
5.3.1	Cross-Validation Techniques	30
5.3.2	Overall Image Classification	30
5.3.3	Relevance of Fused LiDAR Data in Classification	32
5.3.4	Relevance of Repetition of LiDAR Data in Fused Model	33
Chapter 6	Conclusions and Future Works	35
	Bibliography	37
	Vita	41

List of Figures

2.1	A graphical representation of hyperspectral data cube.	6
2.2	Spectral signatures of the 15 classes from 2013_IEEE_GRSS_DF_Contest dataset with 144 spectral bands.	7
2.3	Spectral signatures of the 12 classes from MUUFL Gulfport dataset with 144 spectral bands.	7
2.4	Scaled images of LiDAR information of 320 x 360 scene of MUUFL Gulfport Dataset: Digital Elevation Map (left) and LiDAR Intensity (right) . . .	10
3.1	Fused data vectors for representative classes in 2013_IEEE_GRSS_DF_Contest dataset.	14
3.2	Fused data vectors for representative classes in MUUFL Gulfport dataset. .	15
5.1	Finding optimal LiDAR repetition for the fused hyperspectral and LiDAR data of 2013_IEEE_GRSS_DF_Contest dataset for first 100 epochs based on best overall classification accuracy.	25
5.2	Finding optimal LiDAR repetition for the fused hyperspectral and LiDAR data of MUUFL Gulfport dataset for first 200 epochs based on best overall classification accuracy.	25
5.3	Comparisons of growth in accuracy percentages of CNN model using HSI data and fused HSI and LiDAR data for 2013_IEEE_GRSS_DF_Contest dataset over 200 epochs.	26
5.4	Comparisons of growth in accuracy percentages of CNN model using HSI and fused HSI and LiDAR data for MUUFL Gulfport dataset over 200 epochs. .	27
5.5	The convergence of cost function for 2013_IEEE_GRSS_DF_Contest dataset of CNN model for 200 epochs.	28
5.6	The convergence of cost function for MUUFL Gulfport dataset of CNN model for 200 epochs.	28
5.7	2013_IEEE_GRSS_DF_Contest Dataset: Comparison of (top) Repeated sub-sampling and (middle) Randomization cross-validation techniques, and (bottom) corresponding color codes.	30
5.8	2013_IEEE_GRSS_DF_Contest dataset: CNN using HSI data only (top) and fused HSI and LiDAR data (middle). Google map view (bottom). . . .	31
5.9	MUUFL Gulfport dataset: CNN using HSI data only (left) and fused HSI and LiDAR data (middle). Google map view (right).	32
5.10	Fused data vectors for pixels with similar hyperspectral response.	33

5.11 Fused data vectors for pixels with similar hyperspectral response and Li-DAR values.	34
---	----

List of Tables

2.1	The distribution of data samples across 2013_IEEE_GRSS_DF_Contest training dataset for 15 different classes.	11
5.1	The comparison of HSI vs Fused HSI LiDAR classification accuracies for 2013_IEEE_GRSS_DF_Contest dataset.	29
5.2	The classification accuracies for 2013_IEEE_GRSS_DF_Contest dataset upon introduction of known levels of noise.	29
5.3	The classification accuracies of 4 classes with relatively similar spectral signatures in hyperspectral domain and high contrast in LiDAR values for 2013_IEEE_GRSS_DF_Contest dataset.	32
5.4	The classification accuracies of 4 classes with asphalt material and relatively similar spectral signatures from 2013_IEEE_GRSS_DF_Contest dataset.	33

Abstract

Convolutional neural networks (CNN) have demonstrated excellent performance on various tasks concerning image classification, object and speech recognition, natural language processing (NLP), and are increasingly gaining popularity with geospatial classification. Hyperspectral (HSI) and Light Detection and Ranging (LiDAR) imaging are complementary modalities which are extensively used together for geospatial data collection in remote sensing. HSI data is used to glean information about the material composition and LiDAR data provides information about the geometry of objects in the scene. This thesis proposes a fusion model for HSI and LiDAR to develop a new combined feature, and examines its usage and potential for geospatial classification by applying convolutional neural networks.

Deep convolutional neural networks are implemented on pixel-level fused hyperspectral and LiDAR imagery to assess the classification performance and effectiveness of the proposed fusion model. Two key questions relative to classification performance are addressed: the effect of merging multi-modal data and the effect of uncertainty in the CNN training data. Two recent co-registered HSI and LiDAR datasets are used here to characterize performance. One was collected, over Houston TX, by the University of Houston National Center for Airborne Laser Mapping with NSF sponsorship, and the other was collected, over Gulfport MS, by Universities of Florida and Missouri with NGA sponsorship. Experimental results based on these datasets demonstrate that the proposed method can achieve better classification performance by a margin of over 10% when fused hyperspectral and LiDAR information is used, rather than with hyperspectral data alone and is significantly robust to uncertainties in the training data.

Chapter 1: Introduction

In recent years, it has been shown that remote sensing tasks, such as scene reconstruction, feature enhancement, and classification of targets, are improved when co-registered HSI and LiDAR data are used jointly. This has spurred active research into methods that can reliably fuse and extract information from these complementary sensing modalities. A number of feature-level fusion techniques have been developed that combine features extracted individually to produce a new feature set that better represents the scene [1, 2, 3, 4]. For example, Dalponte, Bruzzone, and Gianelle [5] apply a sequential forward floating selection method to extract features from denoised hyperspectral data. These features are then integrated with corrected elevation and intensity models derived from the LiDAR data and classified using support vector machines and Gaussian maximum likelihood techniques. As part of the winning entry for the 2013 GRSS Data Fusion Contest, Debes *et al.* [1] combined abundance maps obtained through a spectral unmixing procedure with LiDAR data, providing topological information to the classification process. A flexible strategy based on morphological features and subspace multinomial logistic regression was presented in [2] for jointly classifying Hyperspectral Image(HSI) and LiDAR data without the need for regularization parameters.

Various deep learning methods have been used for Hyperspectral Image classification [6, 7, 8]. Deep learning algorithms are based on distributed representations, which are generated by the interactions of factors organized in layers and exploits the idea of hierarchical explanatory factors, where higher level, more abstract concepts are learned from the lower level ones. Deep learning models aim to disentangle these abstractions and pick out which features are useful for learning. These models are often constructed with a greedy layer-by-layer method where specific layers are added, repeated or omitted from the model

experimentally, and varying numbers of layers and layer sizes can be used to provide different amounts of abstraction in the classification model [9].

In deep learning, neural networks, also known as multi-layer perceptrons (MLPs), are the quintessential deep learning models. They consist of three or more layers, and are used to learn deep features of input data, which are features from an abstraction model where a feature in itself is composed of smaller features. They can provide better approximations to nonlinear functions than single-layer classifiers, such as linear support vector machines.

Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs and try to mimic the visual cortex of a living organism, which contains a complex arrangement of cells sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field and are well-suited to exploit the strong spatially local correlation present in natural images. CNN consists of various layers of artificial neuron clusters in which outputs are combined through convolution operations. Some applications of CNNs include material classification, object detection, and face and speech recognition. More recently, the use of deep convolutional neural networks (CNN) has been proposed for classification of hyperspectral imagery [10, 11, 12, 13].

This work is concerned with the classification performance of CNNs when HSI and LiDAR data are combined at the pixel level [14], that is, before feature extraction in the classification process. In this type of fusion LiDAR data is replicated and appended to HSI data for each pixel in the scene. This combined data is then processed with a multilayer CNN [11] to learn the filters producing the strongest response to local input patterns. Pixel-level fusion can have an advantage over other techniques in that it tends to avoid loss of information that may occur during the feature extraction process [15].

Classification performance is evaluated and compared by modifying the CNN parameters and robustness to noise is investigated by introducing classification errors in the training data. We apply our techniques to sample classification problems using two well-known

Hyperspectral and LiDAR datasets that have been recently developed for test purposes.

Chapter 2 explains the concepts related to hyperspectral and LiDAR imaging and the involved datasets. In Chapter 3, different types of hyperspectral and LiDAR fusion are examined, and the proposed fusion technique is discussed. Chapter 4 explains in detail the classification using convolutional neural networks and discusses the intricacies related to CNN model in the spectral domain of the fused feature. Chapter 5 covers the experiments carried out and the results based on class-specific and overall classification accuracies and the impact of presence of uncertainty in the training dataset. Chapter 6 has some final thoughts on the proposed fusion model and its classification using CNN, and provide some future directions in continuing this work.

Chapter 2: Hyperspectral and LiDAR Images

This chapter reviews the acquisition process of Hyperspectral and LiDAR imagery, their applications and data-formats and spectral signatures.

2.1 Hyperspectral Images and Spectral Bands

Hyperspectral imaging (HSI) combines the power of digital imaging and spectroscopy. Imaging spectrometers gather data over a wide and continuous band of the electromagnetic spectrum, which can be used to accurately determine the composition of objects and ground cover in a scene [16]. When the images are acquired at high spatial resolution and co-registered, the resulting data provide a robust and detailed characterization of the earth's surface and its constituent elements.

For each pixel in an image, a hyperspectral camera acquires the light intensity (radiance) for a large number (typically a few tens to several hundred) of contiguous spectral bands. Every pixel in the image thus contains a continuous spectrum (in radiance or reflectance) and can be used to characterize the objects in the scene with great precision and detail and thereby providing a distinguishable spectral signature for every pixel, which can be used by processing techniques to identify and discriminate materials.

There are four prominent techniques of acquiring airborne hyperspectral data: spatial scanning, spectral scanning, non-scanning and spatio-spectral scanning. For remote sensing applications, spatial scanning using a push broom scanner is the most prominent one [16]. In a push broom sensor, a line of sensors arranged perpendicular to the flight direction of the spacecraft is used, and different areas of the surface are imaged as the aircraft flies forward. As a result a set of images is obtained, where each image is the response of a specific sensor. Then these images are geo-referenced using the geographical location

(longitude and latitude), which was acquired by additional GPS systems mounted on aircraft. These geo-referenced images are then combined and formed into a three-dimensional hyperspectral data cube for processing and analysis. The hyperspectral cube in Figure 2.1 [17] displays a two-dimensional projection of a hyperspectral cube. Thus, these complex sensors are developed based on the convergence of two related technologies - spectroscopy and the remote imaging.

Since an entire spectrum is acquired at each point, the user needs no prior knowledge of the sample, and postprocessing allows all available information from the dataset to be mined. Hyperspectral imaging can also take advantage of the spatial relationships among the different spectra in a neighbourhood, allowing more elaborate spectral-spatial models for a more accurate segmentation and classification of the image [18].

Hyperspectral remote sensing is used in a wide array of applications ranging from mining and geology to ecology and surveillance. Additionally, it is rapidly expanding its applications in the fields of astronomy, biology, chemical imaging and physics [16].

A hyperspectral image is a three-dimensional (x, y, λ) data cube for processing and analysis, where x and y represent two spatial dimensions of the scene, and λ represents the spectral dimension (comprising a range of wavelengths) [4].

2.1.1 Spectral Signatures

The spectral signature of a material in an object is a function of the incidental electromagnetic wavelength and material interaction with that section of the electromagnetic spectrum. It is usually recorded as the radiation reflected as a function of the wavelength for a particular surface. Different surface types such as water, bare ground and vegetation reflect radiation differently in various channels, and hence have different spectral signatures.

Most remote sensing applications process hyperspectral or multi-spectral images to extract spectral signatures at each pixel and use them to divide the image in groups of similar

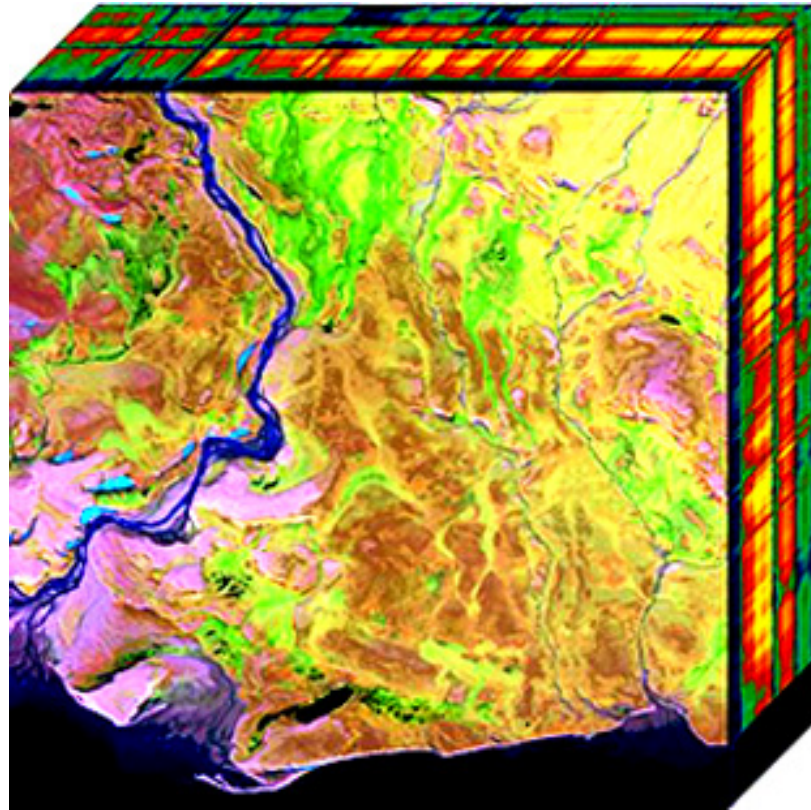


Figure 2.1: A graphical representation of hyperspectral data cube.

pixels (segmentation) using different approaches. As a last step, they assign a class to each group (classification) by comparing with known spectral signatures. Depending on pixel resolution, a pixel can represent many spectral signature "mixed" together - that is why much remote sensing analysis is done to "unmix mixtures". Ultimately correct matching of spectral signature recorded by image pixel with spectral signature of existing elements leads to accurate classification in remote sensing.

As an example, Figure 2.2 plots the spectral signature of the data samples of 15 different classes (materials) from 2013_IEEE_GRSS_DF_Constest dataset across the spectrum of 144 spectral bands. Visually, one can notice the difference in the patterns (spectral signatures) of data-points belonging to different classes. This variation is being leveraged by various classifiers for material or geospatial classification. Figure 2.3 plots the spectral

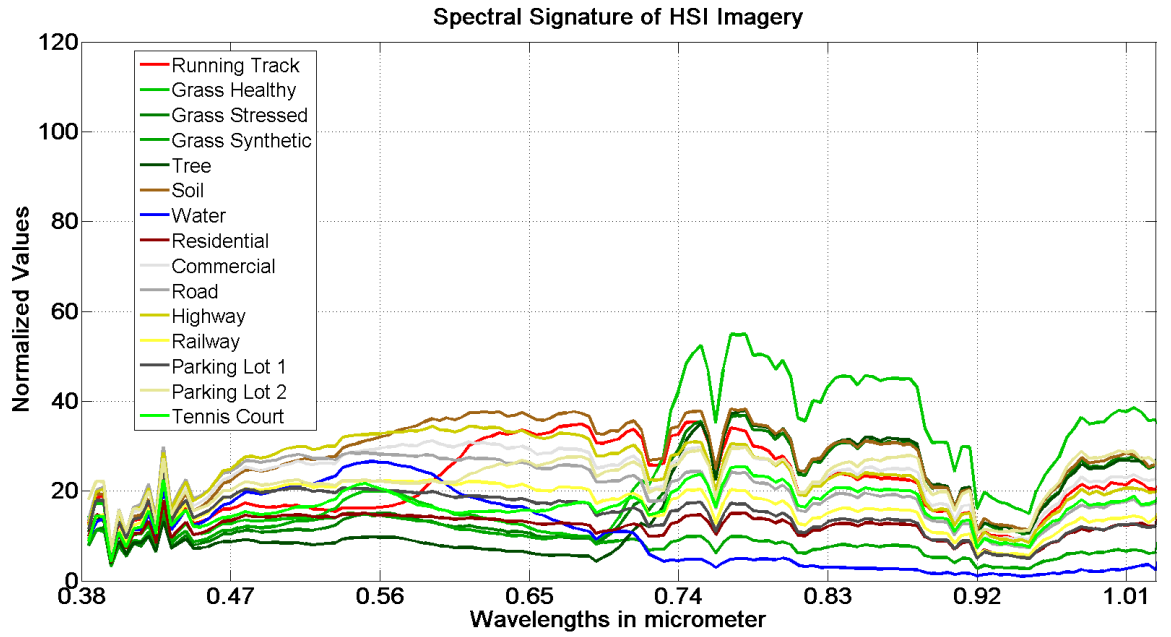


Figure 2.2: Spectral signatures of the 15 classes from 2013 IEEE GRSS DF Contest dataset with 144 spectral bands.

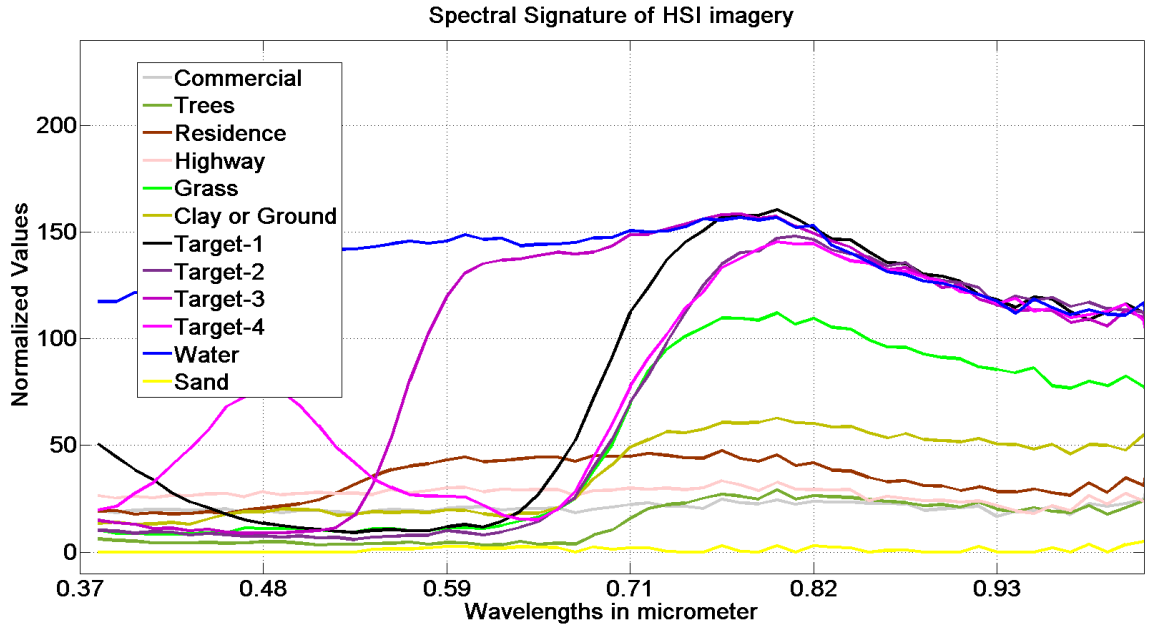


Figure 2.3: Spectral signatures of the 12 classes from MUUFL Gulfport dataset with 144 spectral bands.

signature of the data samples of 12 different classes (materials) from MUUFL Gulfport dataset across the spectrum of 58 spectral bands.

2.2 LiDAR Images

Light Detection and Ranging (LiDAR) is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to objects in a scene. These light pulses combined with other data recorded by the airborne system generate precise, three-dimensional information about the elevation and shape of the Earth's surface characteristics.

A LiDAR instrument principally consists of a laser, a scanner, position and navigation systems (a specialized GPS receiver). LiDAR instruments fitted to aircraft and satellites carry out surveying and mapping by flying over targeted areas. Airplanes and helicopters are the most commonly used platforms for acquiring LiDAR data over broad, continuous areas. A narrow laser-beam can map physical features with very high resolutions; for example, an aircraft can map terrain at 30 cm resolution or better [19].

When an airborne laser is pointed at a targeted area on the ground, the beam of light is reflected by the surface it encounters (backscattering) and sensors record this reflected light to measure a range. When laser ranges are combined with position and orientation data (generated from integrated GPS and Inertial Measurement Unit systems), scan angles, and calibration data, the result is a dense, detail-rich group of elevation points, called a point cloud.

Each point in the point cloud has three-dimensional spatial coordinates (latitude, longitude, and height) that correspond to a particular point on the Earth's surface from which a laser pulse was reflected. The three-dimensional coordinates (latitude x , longitude y , and elevation z) of the target objects are computed from 1) the time difference between the laser pulse being emitted and returned, 2) the angle at which the pulse was fired, and 3) the absolute location of the sensor on or above the surface of the Earth [19]. The point clouds are used to generate other geospatial products, such as digital elevation models (DEM), canopy models, building models, and contours.

2.2.1 Elevation Models

Airborne LiDAR is currently the most detailed and accurate method of creating digital elevation models, replacing photogrammetry [19]. One major advantage in comparison with photogrammetry is the ability to filter out vegetation from the point cloud model to create a digital surface model where areas covered by vegetation can be visualized, including rivers, paths, cultural heritage sites, etc. Based on altitude, airborne LiDAR can be classified as high-altitude and low-altitude applications, the main difference being a reduction in both accuracy and point density of data acquired at higher altitudes. Airborne lidar can also be classified as topographic and bathymetric LiDAR, where the former typically uses a near-infrared laser to map the land, while the latter uses water-penetrating green light to also measure seafloor and riverbed elevations [19].

High-resolution digital elevation maps generated by airborne and stationary LiDAR have led to significant advances in geomorphology, geographic information systems, atmospheric research and meteorology. Today, LiDAR technology has expanded vastly in capability and LiDAR systems are used to perform a range of measurements that include profiling clouds, measuring winds, studying aerosols and quantifying various atmospheric components.

2.2.2 LiDAR Intensity

Intensity is a measure, collected for every point, of the return strength of the laser pulse that generated the point. It is based, in part, on the reflectivity of the object struck by the laser pulse, which in turn is the function of the wavelength used for capturing LiDAR. Also, the strength of the returns varies with the composition of the surface object reflecting the return and thus is equally as good an information source about the ground surface as elevation [20].

Intensity is used as an aid in feature detection and extraction, in LiDAR point classifica-

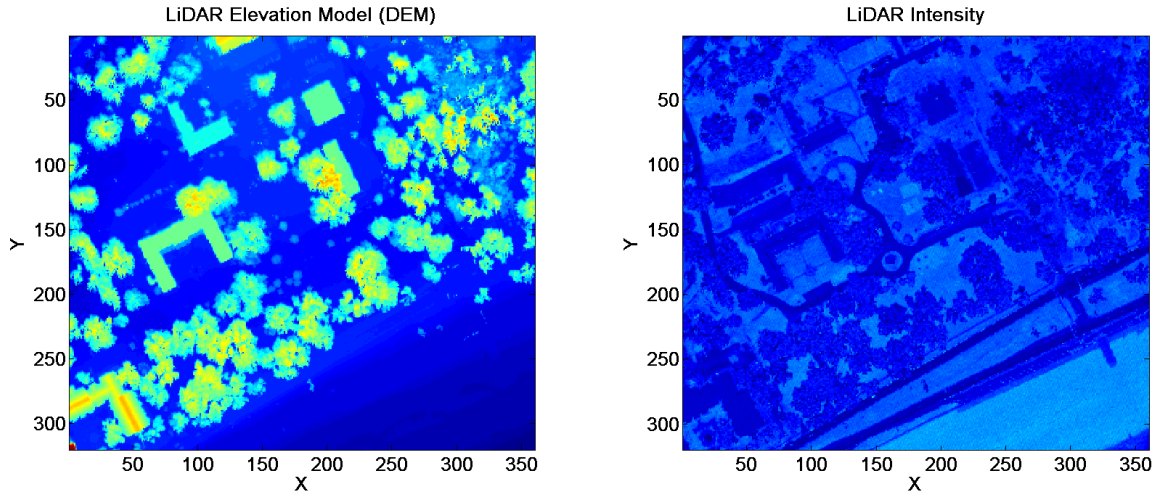


Figure 2.4: Scaled images of LiDAR information of 320 x 360 scene of MUUFL Gulfport Dataset: Digital Elevation Map (left) and LiDAR Intensity (right)

tion, and as a substitute for aerial imagery when none is available. Intensity is relative, not quantifiable, therefore one can expect different values of the same target from same flight and elevation [20].

The file format of the LiDAR will depend on the form of representation of elevation data. In its raw form, LiDAR is a series of points stored as x, y, z where x and y can be longitude and latitude; z can be the elevation in meters or feet or LiDAR intensity or a vector of both or more data for each pixel. Another representation for LiDAR data is as contours lines. The lines can be stored in a number of different DWG and GIS file formats.

A third representation for LIDAR data is as a digital elevation model (DEM). This is a raster format with a matrix (grid) with rows and columns. Each cell is of a fixed sized relative to the earth's surface; each cell holds the average elevation calculated from LiDAR points for that area of the earth's surface represented by the cell.

For our experiments we will use a dataset in raw format of LiDAR images and another dataset with DEM and Laser intensity.

2.3 Datasets Used in this Thesis

We employ two HSI and LiDAR datasets for the work presented in this thesis: the data used for 2013 IEEE GRSS Data Fusion contest and the data acquired over Gulfport, Mississippi.

2.3.1 2013 IEEE GRSS DF Contest Dataset

This dataset was acquired over the University of Houston campus and the neighboring urban area for the 2013 IEEE GRSS Data Fusion Contest [21]. The Hyperspectral imagery has 144 spectral bands in the $0.38\mu\text{m}$ to $1.050\mu\text{m}$ region. It involves two datasets a Hyperspectral image and a LiDAR derived Digital Surface Model (DSM), both at the same spatial resolution (2.5m). There are 1903×329 pixels in the scene and true labels are known for only 2832 pixels. Training dataset is created utilizing 60 labeled pixels (data points) per class and all other labeled pixels are used for testing dataset. Table 2.3.1 shows the counts of data samples present for each class in training and testing datasets. Here is a link to dataset contributors and competition: http://hyperspectral.ee.uh.edu/?page_id=459.

	Healthy grass	Stressed grass	Synthetic grass	Trees	Soil	Water	Residential	Commercial	Road	Highway	Railway	Parking Lot 1	Parking Lot 2	Tennis Court	Running Track
Training Dataset	60	60	60	60	60	60	60	60	60	60	60	60	60	60	60
Testing Dataset	138	130	132	128	126	122	136	131	133	131	121	132	124	121	127

Table 2.1: The distribution of data samples across 2013 IEEE GRSS DF Contest training dataset for 15 different classes.

2.3.2 MUUFL Gulfport Dataset

The Gulfport dataset is composed of HSI and LiDAR datasets that were simultaneously acquired by sensors on the same aircraft, taken on November 8th, 2010, at Gulfport, Mississippi [22]. The ground sampling distance is 1 meter. The original HSI dataset has 72 wavelength bands ranging from $.375\mu\text{m}$ to $1.05\mu\text{m}$. Ignoring the noisy bands, we just used 58 bands from Hyperspectral data, and Digital Elevation Model (DEM) and LiDAR Intensity both derived from co-registered LiDAR data.

Chapter 3: Hyperspectral and LiDAR Fusion

The idea of fusion of Hyperspectral and LiDAR images is to utilize both the knowledge about the Earth’s altitude and the material composition of the Earth’s surface to create more robust classification models. This chapter broadly discusses different types of Hyperspectral and LiDAR fusion and our proposed fusion technique for subsequent classification via convolutional neural networks.

3.1 Fusion types

In general, fusion techniques can be classified into three different levels: pixel/data level, feature level and decision level fusion [14, 23].

3.1.1 Pixel/Data Level Fusion

Image fusion at pixel level means fusion at the lowest processing level referring to the merging of measured physical parameters. It is also commonly referred as the fusion before the feature extraction level [23], in which the data obtained from each sensor is used to form a larger data vector. The new data vector now has a higher dimensionality and represents a pixel in the scene in a different (and hopefully more discriminating) hyperspace thereby suggesting a better classification. In a way it is the combination of raw data from multiple sources into a single input, which is expected to be more informative and synthetic than either of the input data or to reveal the changes between data sets acquired at different times [14]. There has been growing popularity for applications focusing on pixel-level fusion for hyperspectral images [24, 25, 26].

3.1.2 Feature Level Fusion

Feature level fusion utilizes features, like edges, corners, lines, texture parameters in images from different data sources and then combines them into one or more feature maps that may be used instead of the original data for further processing [14]. In feature-level fusion based classification models, the emphasis is on deciding and extracting the distinguishing factors or features that will lead to the improved classification. Recently, various efforts utilizing hyperspectral data and feature level fusion with LiDAR data has been published [27, 28].

3.1.3 Decision Level Fusion

Decision level fusion combines the scores from multiple classification algorithms to yield a final fused decision. When the results from different algorithms are expressed as confidences rather than decisions, it is called soft fusion; otherwise, it is called hard fusion. Methods of decision fusion include voting methods, statistical methods and fuzzy logic based methods [14]. Various decision-based classification had been devised in the recent past for decision level fusion [29, 30, 31].

3.2 Proposed Pixel Level Fusion

Pixel based fusion necessitates accurate geo-coding of Hyperspectral and LiDAR images. In addition, pixel based methods deal with very large data volumes and long computation times are involved, while feature level image fusion uses corresponding structures which makes the data dimensionality a less critical issue, favoring its widespread use for classification purposes.

On the contrary, classification of certain classes with very similar (close) spectral signatures using feature based fusion models may be less accurate at times than models using just hyperspectral data, specifically when the variance amongst the features generated by

these feature based fusion models is much less than the variance of the actual data (as features are just the abstraction of the original data), and hence difficult to distinguish and hard to classify. This necessitates the use of maximum available data i.e. favoring pixel based classification. The pixel based methods have the advantage against feature or information level fusion of using the most of the original data avoiding a loss of information which may occur during a feature extraction process [15].

We want our model to learn to use primarily Hyperspectral information to classify, and LiDAR information to aid or correct the close cases, and not the other way round. Our premise is that LiDAR information impart variance in the spectral signatures of similar classes and thus will help in achieving separation amongst these classes.

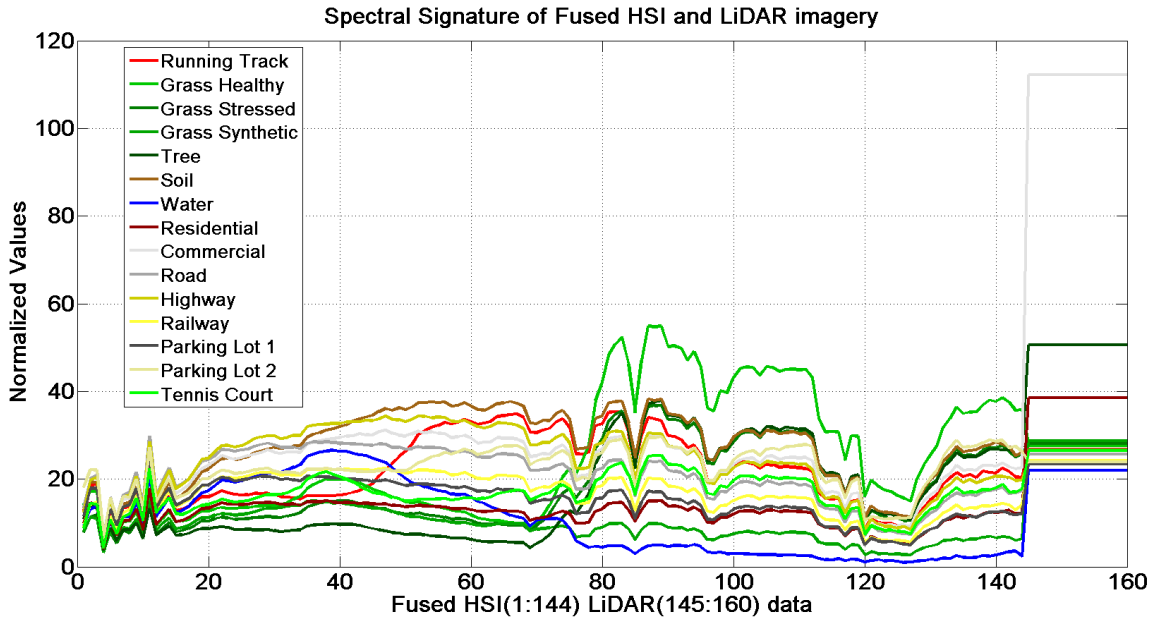


Figure 3.1: Fused data vectors for representative classes in 2013 IEEE GRSS DF Contest dataset.

Formally, we denote our fusion approach as follows. Assume that the LiDAR and HSI datasets are geo-referenced and have been pre-processed to have the same spatial resolution, providing information for the same surface area over the Earth. Let column vector $\mathbf{h}(x, y) \in \mathbb{R}^{M_1}$ denote the spectral response over M_1 channels and $\mathbf{d}(x, y) \in \mathbb{R}^{M_2}$ denote a

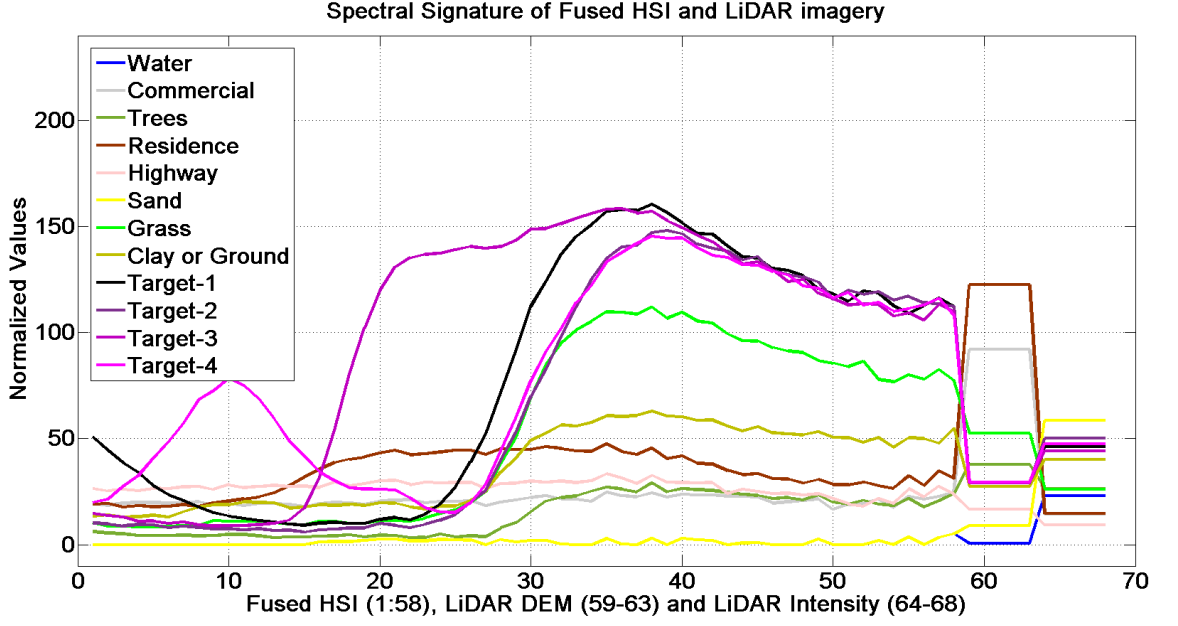


Figure 3.2: Fused data vectors for representative classes in MUUFL Gulfport dataset.

column vector of components derived from the LiDAR data, such as elevation and LiDAR intensity, at each point (x, y) in a regularly spaced grid over the observed surface. Further, each component of $\mathbf{d}(x, y)$ is scaled to ensure balance among the fused data sources. We then define a new data vector for point (x, y) as follows:

$$\mathbf{g}(x, y) = \begin{bmatrix} \mathbf{h}(x, y) \\ \mathbf{d}(x, y) \otimes \mathbf{1} \end{bmatrix}, \quad (3.1)$$

where \otimes is the Kronecker product, $\mathbf{1}$ is a column vector of ones and hence $\mathbf{d}(x, y) \otimes \mathbf{1}$ is a repetition of the LiDAR components. The length of vector $\mathbf{1}$ is an additional parameter that we include in the characterization of performance of CNN for fused LiDAR and HSI data. In general, the repetition of LiDAR data, relative to the length of the HSI vector, can be used as a form of weighting the desired influence of one modality over the other in the classification procedure. Figure 3.1 and 3.2 show representative fused data vectors for our datasets.

Chapter 4: Convolutional Neural Networks

A convolutional neural network (CNN) is a multilayer neural network inspired by the organization of neurons in the animal visual cortex [32]. CNNs have been shown to achieve competitive and even better performance than human beings in some visual problems [?]. The possibility of applying CNNs for HSI classification has been explored in [11]. Here, classification is based on spectral information rather than on edges, blobs, corners, texture as CNN tries to learn the representation of a spectral signature for each class and uniquely distinguish it from all the other classes. In this chapter, we present the particular architecture of our CNN for classification of fused HSI-LiDAR.

4.1 CNN Architecture

Here, we build upon the CNN architecture recently developed in [11] for processing HSI data. CNN is composed of two functional units: Feature Extraction Unit and the Classification Unit. The feature extraction unit gets a fused HSI LiDAR image as input, and it produces a down-sampled activation map as output. This output is fed to the classification unit, where the classification happens with the help of the fully-connected layer of artificial neurons and posterior probabilities are generated by the output layer.

The CNN model is formed by a stack of distinct layers that transform the input volume into an output volume through a differentiable function, and is comprised of one or more convolutional layers and intermediate subsampling layers, followed by at least one fully connected layer.

4.1.1 Input Layer

The input layer consists of the fused vectors $\mathbf{g}(x, y)$ from equation (3.1). For simplicity, we use a single index i to enumerate these vectors in the spatial domain, i.e. $\mathbf{g}_i = \mathbf{g}(x, y)$ for $i \in \{1, 2, \dots, N\}$, where N is the total number of spatial points (x, y) in the dataset.

4.1.2 Convolutional Layer

The convolutional layer performs 1-D convolutions with learnable 1-D filters, which are in turn processed using $\tanh()$ activation function [33] to produce activation or feature maps. These activation maps can be considered to be a collection of responses to local receptive fields spread across the hyperspectral spectrum, very similar to edges, corners and blobs in a feature selection task of regular images. The proposed CNN model uses parameter sharing scheme in convolutional layer to control the number of learnable parameters.

This convolution operation is not done for just a single filter, but is performed separately for a set of K 1-D filter vectors, $\{\mathbf{f}_k\}$, of fixed length. In this layer, each of these filters is convolved with \mathbf{g}_i to produce $\mathbf{u}_{i,k} = \tanh(\mathbf{g}_i \star \mathbf{f}_k)$, where \star is the convolution operation.

4.1.3 Max-Pooling Layer

The activation maps generated by convolutional layer are further processed by the max-pooling layer, which implements $\max()$ function over an input of certain size, to choose the locally maximum activations across the spectrum which are then fed to classifier unit of CNN. In our CNN model, $\mathbf{u}_{i,k}$ are subsampled by taking the maxima over non-overlapping regions of length 2, producing vectors $\mathbf{u}_{i,k}^s$ of half the size. Next, the subsampled vectors $\mathbf{u}_{i,k}^s$ are stacked together as

$$\mathbf{u}_i^s = [\mathbf{u}_{i1}^s, \mathbf{u}_{i2}^s, \dots, \mathbf{u}_{iK}^s]^T$$

4.1.4 Fully-Connected Layer

This layer has exactly the same structure, function and implementation as a hidden layer in a neural network. Neurons in a fully connected layer have full connections to all activations in the previous layer, and are responsible for the major classification process based on the generated activation maps.

The stacked vector from max-pooling layer is then used as input for the hidden neuron layer, producing output vector \mathbf{y}_i . This process is expressed as:

$$\mathbf{y}_i = f(\mathbf{W}^{(h)}\mathbf{u}_i^s) + \mathbf{b}^{(h)}, \quad (4.1)$$

where $\mathbf{W}^{(h)}$ is the weight matrix associated with the hidden neuron layer and $\mathbf{b}^{(h)}$ is a vector of unit bias. The number of rows, P , in $\mathbf{W}^{(h)}$ corresponds to the number of neurons in the layer. The function $f(\cdot)$ is the layer's activation function defined as $f(x) = \tanh(x)$ applied element-wise to the input vector in equation (4.1).

4.1.5 Output Layer

This layer's structure is very similar to fully connected layer, and is composed of the same number of neurons as the number of classes we are trying to classify our input data. Each neuron in output layer yields a probability corresponding to a specific class and the predicted class would be the one with maximum value.

The output of a network needs to be interpretable as posterior probabilities, and thus it is highly desirable for those outputs to lie between zero and one and to sum to one. The purpose of the softmax activation function is to enforce these constraints on the outputs. Statisticians usually call softmax a "multiple logistic" function. It reduces to the simple logistic function when there are only two categories.

Formally, given input \mathbf{y}_i the output layer applies the following operations:

$$\mathbf{t}_i = \exp(\mathbf{W}^{(o)}\mathbf{y}_i + \mathbf{b}^{(o)}), \quad (4.2)$$

$$\mathbf{z}_i = \frac{1}{\|\mathbf{t}_i\|_1} \mathbf{t}_i, \quad (4.3)$$

where the softmax function is applied to $\mathbf{W}^{(o)}\mathbf{y}_i + \mathbf{b}^{(o)}$. Here, $\mathbf{W}^{(o)}$ is the weight matrix associated with the output layer, and $\mathbf{b}^{(o)}$ is a vector of unit bias. The number of rows, C , in $\mathbf{W}^{(o)}$ corresponds to the number of labeled classes specified during the training phase of the CNN. The final output vector \mathbf{z}_i contains the estimated class probabilities for the classification of input vector \mathbf{g}_i . Producing the output \mathbf{z}_i from the input \mathbf{g}_i constitutes the **feed-forward** part of the CNN.

4.2 Backpropagation

Backpropagation is carried out to train a CNN model to minimize a cost function of the model. At the heart of backpropagation is an expression for the partial derivative of the cost function C (also referred to as objective function) with respect to any weight (or bias) in the network. The expression tells us how quickly the cost changes when we change the weights and biases. In CNNs, updating trainable parameters in convolutional kernels ($\{\mathbf{f}_k, k = 1 : K\}$) trains the feature extraction unit to yield better features, while the trainable weights in the neurons of fully-connected ($\mathbf{W}^{(h)}, \mathbf{b}^{(h)}$) and output layer ($\mathbf{W}^{(o)}, \mathbf{b}^{(o)}$) strive for better classification. All of these parameters are trained using the standard gradient descent algorithm. Misclassifications are obtained from the feed-forward process, which are then utilized during backpropagation algorithm to calculate the gradients. These gradients are then used to update the trainable parameters.

Formally, the backpropagation process is specified as follows:

Let $\{\mathbf{g}_i\}_{i=1}^{N_t}$ denote the set of N_t samples used for training and let $\{L_j\}_{j=1}^C$ denote the set of C classification labels. Further, let \mathbf{x} denote a vector of all the trainable parameters,

specifically, $\{\mathbf{f}_k\}$, $\mathbf{W}^{(h)}$, $\mathbf{b}^{(h)}$, $\mathbf{W}^{(o)}$, and $\mathbf{b}^{(o)}$. Recall that \mathbf{z}_i in equation (4.3) is a vector of length C ; let $z_{i,j}$ denote the j^{th} component of vector \mathbf{z}_i .

We define the following objective function:

$$J(\mathbf{x}) = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^C \delta_{i,j} \log(z_{i,j}), \quad (4.4)$$

where $\delta_{ij} = \begin{cases} 1 & \text{if } \mathbf{g}_i \text{ belongs to class } L_j, \\ 0 & \text{otherwise.} \end{cases}$

Equation (4.4) is the well-known logarithmic loss function which is used to maximize predictive accuracy by rewarding correct classifications that are made with a high probability, i.e., whenever $z_{i,j}$ is close to 1 and sample $\mathbf{g}_i \in L_j$.

We minimize (4.4) using a standard gradient descent approach. Starting with an initial guess \mathbf{x}_0 for a local minimum of $J(\mathbf{x})$, we compute:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n \nabla J(\mathbf{x}_n), \quad n \geq 0 \quad (4.5)$$

where $\nabla J(\mathbf{x}_n)$ is the gradient of J . In our implementation the step size α_n is kept constant with $\alpha_n = 0.08$. We stop the iteration when the relative error in the cost function is sufficiently small.

4.3 Hyper-Parameters

A CNN model requires tweaking of various hyperparameters to achieve better classification results, and the optimal or near-optimal value of a hyperparameter varies largely on: (1) the dataset, and (2) other hyperparameters.

The initial learning rate α is associated with both the convolutional and fully-connected layers and is one of the most important hyperparameters. In standard back-propagation algorithms, too low a learning rate makes the network learn very slowly, and a too high

learning rate can make the weights and cost function diverge adversely (so there is no or very little learning). In the presented work, we carried out all the experiments with the learning rate residing in the range of 0.001 to 0.01, which suited well for the convergence of objective function.

Feature extraction is primarily carried out by convolutional layer with the help of convolutional filters. A number of features from the input space are extracted by using a set of filters, and each of these 1-D filters are initialized with a small random value in the range of -0.05 to 0.05. In the proposed model the learned filters turned out to be differential in nature. The more the number of filters used, the more features are extracted. This can affect the classification accuracy adversely when the same feature is being recognized multiple times in feature space, resulting in inconsistent behavior of the classifier. Thus, sufficient number of filters with optimum size, to represent the entire feature space without many duplicates gives the best results.

The hyperparameters that control the size of the output volume of convolution layer are the stride-size and the zero-padding size. All experiments are carried out with a stride size of 1 and no zero padding. Smaller strides work better in practice and size 1 strides allow the setup to leave all down-sampling to the max-pooling layer, with the convolutional layer only transforming the input volume activations.

For the max-pooling layer, down-sampling scale size (i.e. the sub-sampling extent in the activation map) is very important hyperparameter that is used to control down-sampling rate. The max-pooling layer passes on the prominent features of the activation maps (generated by convolutional layer) to the fully-connected layer. The down-sampling scale size decides for the proportion of the activation values being passed on to fully-connected layer.

For a fully-connected layer, the number of neurons is the hyperparameter which needs to be fine-tuned and set optimally, to achieve best classification results. The more the number of neurons in fully-connected layer, more is the number of free parameters and may

affect the performance drastically in certain situations, especially when training dataset is relatively small. Despite of the extraction of good and distinguishable features, too few neurons in a fully-connected layer may degrade a model's classification power.

Chapter 5: Experimental Results

In this chapter, we present many of the experiments carried out to validate the use of pixel/data level fusion and CNN for classification. We are particularly interested in understanding and characterizing the effect of HSI-LiDAR fusion scheme on the classification accuracy of CNN models.

5.1 Experimental Setup

5.1.1 Datasets

We employ `2013_IEEE_GRSS_DF_Contest` dataset, there are 1903×329 pixels in the scene and true labels are known for only a small subset of these pixels. We utilized a total 2832 known (labeled) pixels spread over all $C = 15$ classes. We also employed a subset of the MUUFL Gulfport dataset [34, 35] which consists of a hyperspectral image of 58 spectral bands and co-registered LiDAR elevation and intensity data. There are 320×360 pixels in this dataset. For additional details related to datasets see Chapter 2.

5.1.2 Implementation

We modified programs for Open Source DeepLearnToolbox from Rasmus Berg Palm [36] and used MATLAB for all the experiments. Additionally, we used Spectral Python package (Spy) for data visualization purposes. The code is available upon request.

5.1.3 Computing Environment

We utilized WFU Deac cluster for conducting all of our experiments. Specifically, we submitted jobs to individual nodes in the cluster with 8 cores and 96GB RAM each.

5.2 Experiments

5.2.1 Experiment 1: Repetition of LiDAR information

The experiments are carried out with the purpose of getting optimum hyper-parameters for training a CNN model over the training dataset in order to achieve maximum accuracy of CNN classifier for the testing dataset. The optimization space is searched for an optimal LiDAR repetitions value fetching maximum accuracy over certain epochs. This was followed by conducting experiments to track the performance of the fused hyperspectral LiDAR model by estimating the maximum accuracy achieved over a large range of epochs (10, 50, 100, 200, 1000). The size of the training dataset relative to that of the test dataset is an important consideration of practical value. Too large a training dataset can lead to over-fitting and is also unrealistic in most imaging applications.

For the `2013_IEEE_GRSS_DF_Contest` dataset we selected 900 vectors (60 samples per class) from the fused dataset $\{g_i\}$ for the training phase of the CNN. The remaining 1932 known samples were used for testing and characterizing classification accuracy (a ratio of approximately 1:2). Additionally, we used $K = 40$ convolution filters, $P = 60$ neurons in the fully connected layer and a learning rate of 0.08 for the same. For MUUFL Gulfport dataset, we selected 60 pixels for each of the 12 classes for training and used an additional 1620 labeled pixels for testing, and $K = 20$ convolution filters, $P = 20$ neurons in the fully connected layer and a learning rate of 0.08.

Figure 5.1 plots the varying percentage accuracies for first 100 epochs over a range of LiDAR repetitions (1 to 26), and it suggests that 16 LiDAR repetitions (11% to 14% of HSI data vector) produced best classification results. Figure 5.2 plots the varying percentage accuracies for first 200 epochs over a range of LiDAR repetitions (1 to 16), and indicates a repetition of the LiDAR elevation and intensity components by a factor of 5 to yield the best classification accuracies. Figures 5.1 and 5.2 suggest that finding an optimal LiDAR

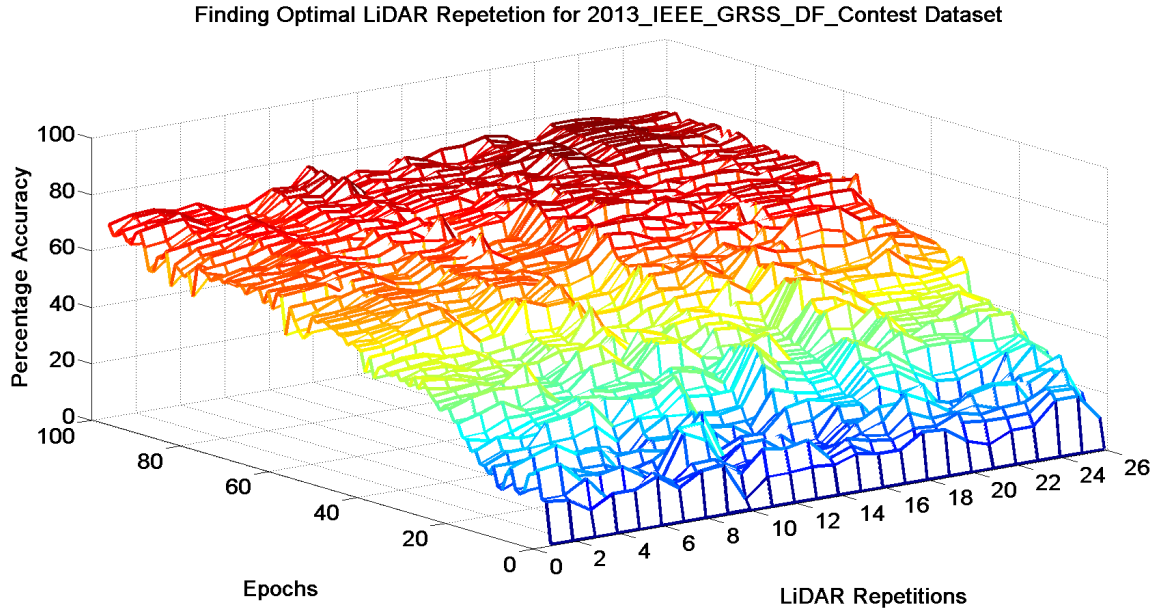


Figure 5.1: Finding optimal LiDAR repetition for the fused hyperspectral and LiDAR data of 2013_IEEE_GRSS_DF_Contest dataset for first 100 epochs based on best overall classification accuracy.

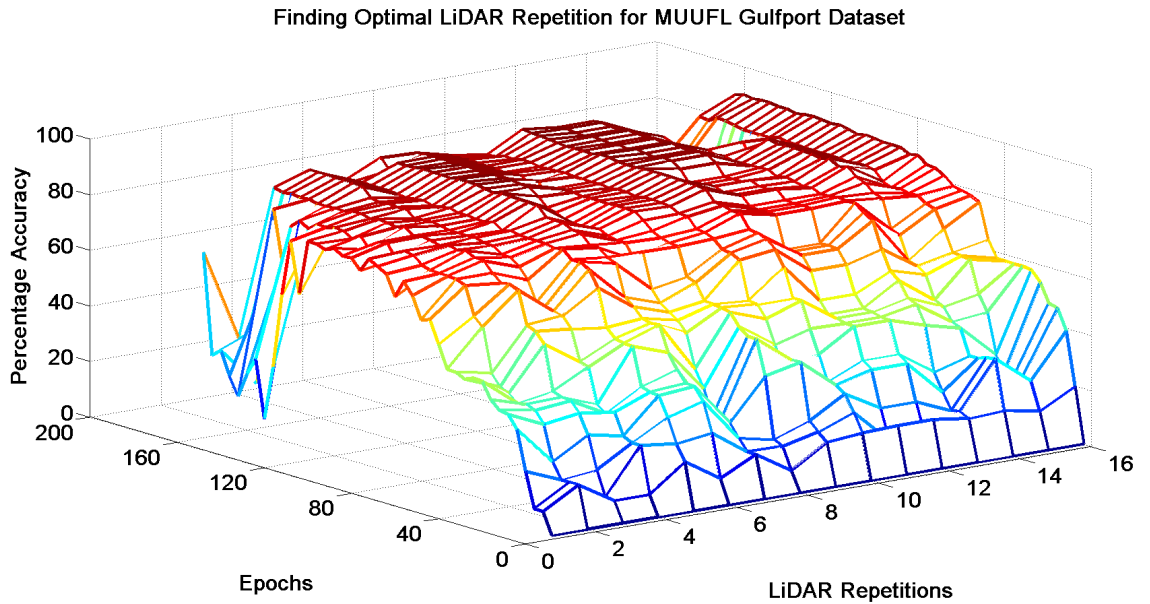


Figure 5.2: Finding optimal LiDAR repetition for the fused hyperspectral and LiDAR data of MUUFL Gulfport dataset for first 200 epochs based on best overall classification accuracy.

repetition is not an ill-posed problem i.e. LiDAR repetition within a certain range for fused HSI-LiDAR model gives us an improvement in the overall classification accuracy as compared to using just hyperspectral data. Too small a recurrence of the LiDAR value yields no improvement while too large a recurrence tends to decrease overall classification.

5.2.2 Experiment 2: Classification on HSI vs. Fused HSI-LiDAR

Experiments using CNN are done to compare classification for Fused HSI-LiDAR model with the one using HSI data alone. For this comparison, first, optimal hyper-parameters for CNN model are estimated using just hyperspectral information, and used as is for hyperspectral and LiDAR fusion variations. The training was performed separately for both the models but using the same training dataset. This was followed by comparison of the trained models using the same test datasets. Plots for overall accuracy and convergence of cost function with respect to increasing number of epochs are also presented for both HSI only and fused HSI-LiDAR data.

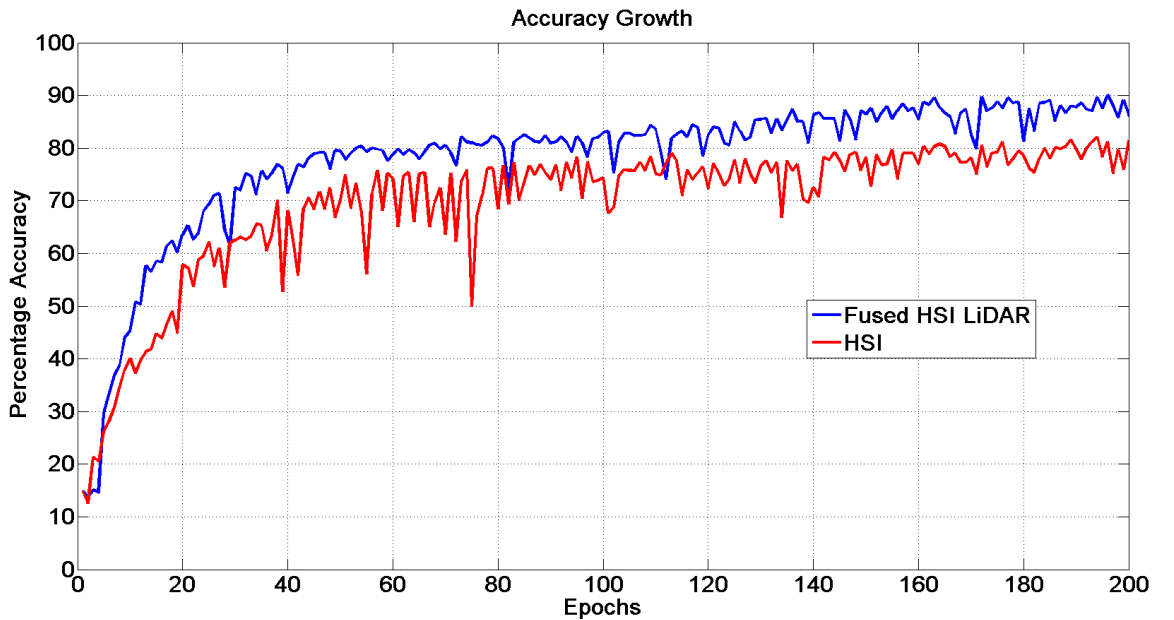


Figure 5.3: Comparisons of growth in accuracy percentages of CNN model using HSI data and fused HSI and LiDAR data for 2013_IIEEE_GRSS_DF_Contest dataset over 200 epochs.

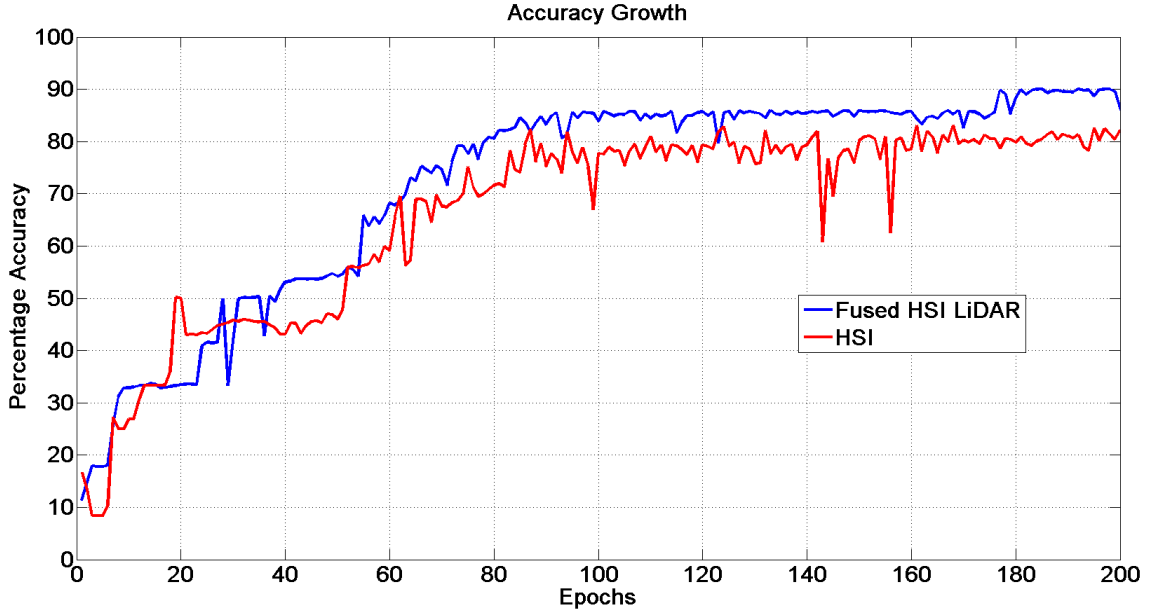


Figure 5.4: Comparisons of growth in accuracy percentages of CNN model using HSI and fused HSI and LiDAR data for MUUFL Gulfport dataset over 200 epochs.

Figure 5.3 compare the classification accuracy obtained with HSI data alone and with fused HSI and LiDAR data for the 2013_IEEE_GRSS_DF_Contest dataset. As can be observed, the accuracy in the CNN output is roughly 10% higher for the fused vectors relative to classification via the HSI data alone. Moreover an accuracy of 80% is reached in 55 iterations (or epochs) compared to 160 for the HSI data alone. Similarly, Figure 5.4 plots the comparison of classification accuracies for MUUFL Gulfport dataset.

Cost functions are plotted to verify the efficacy of fused HSI-LiDAR model over HSI model. For both the datasets, the cost function of CNN model for fused HSI and repeated LiDAR converged quite fast and smoothly as compared to the one for only HSI data.

While the accuracy growth plots compare the overall classification of the CNN model, we are also interested in characterizing the class specific accuracies of the proposed model, and in this regard Table 5.1 shows the complete error matrix for all the classes in the dataset. The rows represent the actual classes while the columns represent the predicted classes. Notice that accuracies of over 98% are achieved for stressed grass, trees, soil, tennis court

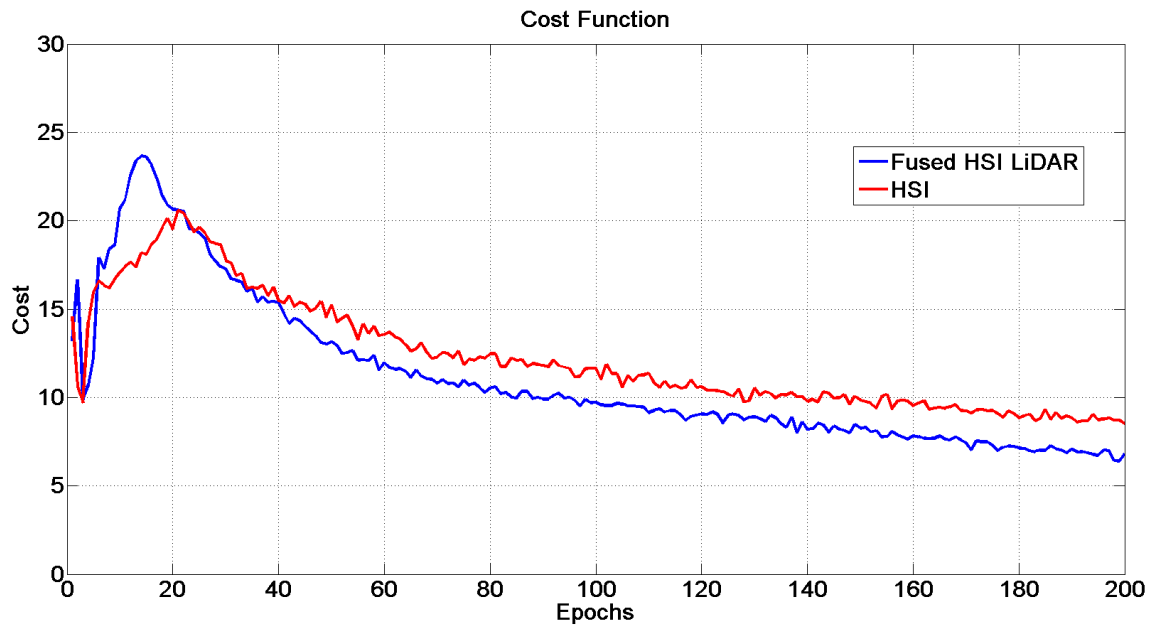


Figure 5.5: The convergence of cost function for 2013_IEEE_GRSS_DF_Contest dataset of CNN model for 200 epochs.

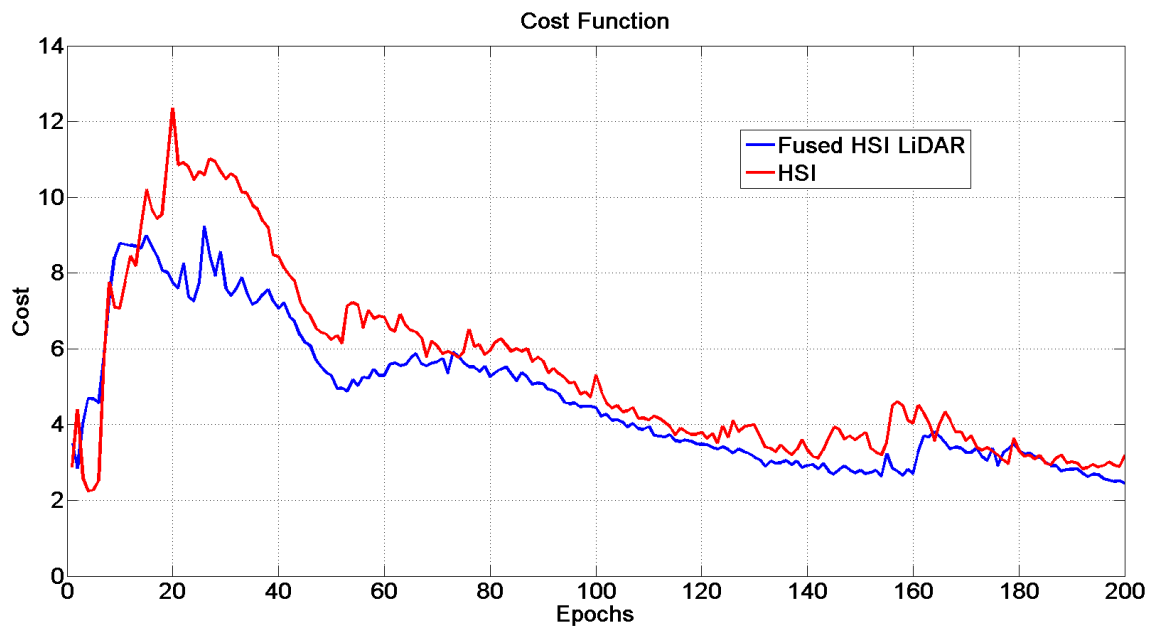


Figure 5.6: The convergence of cost function for MUUFL Gulfport dataset of CNN model for 200 epochs.

Table 5.1: The comparison of HSI vs Fused HSI LiDAR classification accuracies for 2013_IEEE_GRSS_DF_Contest dataset.

HSI Fused	Healthy grass	Stressed grass	Synthetic grass	Trees	Soil	Water	Residential	Commercial	Road	Highway	Railway	Parking Lot 1	Parking Lot 2	Tennis Court	Running Track
Healthy grass	96 % 95 %	4 % 5 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %
Stressed grass	0 % 0 %	100 % 100 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %
Synthetic grass	0 % 0 %	0 % 0 %	98 % 95 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 1 %	1 % 3 %	0 % 1 %	1 % 0 %	0 % 0 %	0 % 0 %
Trees	6 % 2 %	3 % 0 %	0 % 0 %	91 % 98 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %
Soil	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	98 % 98 %	0 % 0 %	0 % 0 %	0 % 0 %	1 % 0 %	0 % 1 %	1 % 0 %	0 % 0 %	0 % 1 %	0 % 0 %	0 % 0 %
Water	0 % 0 %	0 % 0 %	1 % 0 %	0 % 0 %	0 % 0 %	91 % 93 %	3 % 3 %	0 % 0 %	0 % 0 %	2 % 0 %	0 % 0 %	1 % 2 %	2 % 2 %	0 % 0 %	0 % 0 %
Residential	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	85 % 90 %	1 % 4 %	0 % 0 %	0 % 0 %	4 % 6 %	7 % 0 %	1 % 0 %	2 % 0 %	0 % 0 %
Commercial	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	2 % 0 %	0 % 0 %	1 % 1 %	54 % 87 %	10 % 0 %	2 % 0 %	0 % 1 %	3 % 11 %	28 % 0 %	0 % 0 %	0 % 0 %
Road	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 1 %	0 % 0 %	0 % 0 %	2 % 0 %	52 % 82 %	37 % 3 %	5 % 7 %	3 % 2 %	1 % 4 %	0 % 1 %	0 % 0 %
Highway	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	14 % 1 %	62 % 85 %	7 % 6 %	17 % 8 %	0 % 0 %	0 % 0 %	0 % 0 %
Railway	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	17 % 9 %	0 % 0 %	2 % 0 %	3 % 3 %	73 % 83 %	5 % 5 %	0 % 0 %	0 % 0 %	0 % 0 %
Parking Lot 1	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	6 % 0 %	0 % 0 %	2 % 0 %	0 % 5 %	32 % 32 %	15 % 3 %	3 % 1 %	42 % 52 %	0 % 7 %	0 % 0 %	0 % 0 %
Parking Lot 2	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	2 % 3 %	0 % 0 %	8 % 0 %	9 % 0 %	17 % 7 %	5 % 2 %	0 % 6 %	11 % 9 %	46 % 69 %	1 % 2 %	1 % 2 %
Tennis Court	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 1 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	100 % 99 %	0 % 0 %	0 % 0 %
Running Track	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 2 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	0 % 0 %	100 % 98 %	

and running track, and classes like commercial, road, and parking lot 2 have improved the classification accuracies by over 20%.

5.2.3 Experiment 3: Uncertainties in the Training Data

In this experiment, we consider the possibility of misclassification error in the training data due to human or pre-processing oversight. To do this we randomly switch the labels for a percentage of the 900 data vectors \mathbf{g}_i used for training of the CNN. We then classify the 1932 testing data vectors using the so-trained CNN. Table 5.2 shows the effect of up to 20% misclassification error in the training data on the true positive classification. Interestingly, the algorithm appears in most cases to be robust to such error, except for classes with relative low true positive classification.

Table 5.2: The classification accuracies for 2013_IEEE_GRSS_DF_Contest dataset upon introduction of known levels of noise.

% Error introduced	Healthy grass	Stressed grass	Synthetic grass	Trees	Soil	Water	Residential	Commercial	Road	Highway	Railway	Parking Lot 1	Parking Lot 2	Tennis Court	Running Track
No Error	94.93%	100.00%	95.45%	97.66%	98.41%	92.62%	89.71%	87.02%	81.95%	84.73%	82.64%	52.27%	69.35%	99.17%	98.43%
1 %	94.93%	100.00%	95.45%	97.66%	99.21%	91.80%	85.29%	85.50%	84.96%	88.55%	82.64%	57.58%	58.87%	99.17%	97.64%
5 %	95.65%	100.00%	96.21%	99.22%	100.00%	92.62%	91.18%	86.26%	84.96%	81.68%	75.21%	43.94%	61.29%	99.17%	98.43%
10 %	95.65%	100.00%	100.00%	99.22%	100.00%	96.72%	93.38%	84.73%	84.21%	84.73%	77.69%	54.55%	51.61%	99.17%	98.43%
15 %	96.38%	100.00%	100.00%	97.66%	100.00%	95.90%	91.91%	90.84%	71.43%	84.73%	81.82%	47.73%	56.45%	99.17%	98.43%
20 %	97.10%	100.00%	100.00%	98.44%	100.00%	95.08%	91.18%	92.37%	64.66%	79.39%	75.21%	40.15%	58.87%	100.00%	97.64%
25 %	95.65%	100.00%	99.24%	99.22%	100.00%	94.26%	93.38%	95.42%	68.42%	83.21%	73.55%	44.70%	58.87%	99.17%	98.43%
30 %	97.10%	100.00%	100.00%	97.66%	99.21%	95.90%	84.56%	95.42%	78.20%	63.36%	68.60%	40.15%	66.13%	100.00%	97.64%

5.3 Additional Results

5.3.1 Cross-Validation Techniques

The best results for 2013_IEEE_GRSS_DF_Contest dataset was recorded for randomization within the dataset, followed by repeated sub-sampling and K-fold. Figure 5.7 plots the color coded results for repeated sub-sampling and randomization within the training dataset cross-validation techniques, where the former seemed to perform worse than the latter.

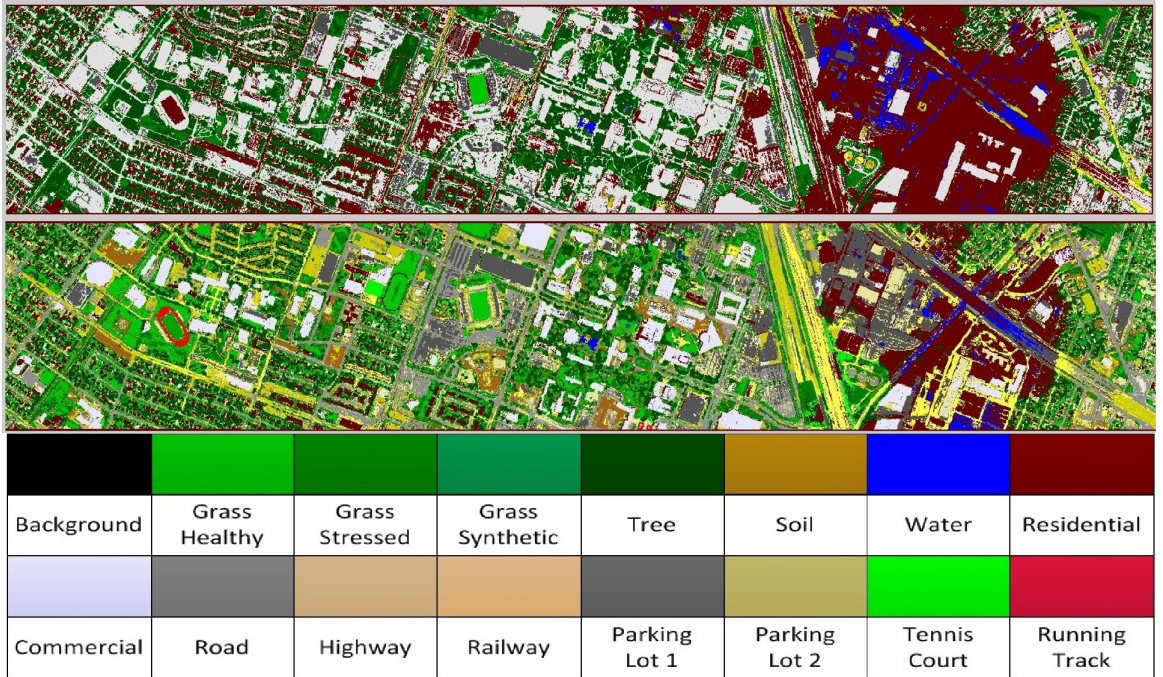


Figure 5.7: 2013_IEEE_GRSS_DF_Contest Dataset: Comparison of (top) Repeated sub-sampling and (middle) Randomization cross-validation techniques, and (bottom) corresponding color codes.

5.3.2 Overall Image Classification

A visual comparison in the classification of all the pixels of the 2013_IEEE_GRSS_DF_Contest dataset is given in Figure 5.8. For these results, 100 known pixels per class were used for training of the CNN instead of 60 (a ratio of 1.13:1 between training and test datasets). This

change in the size of the training data resulted in approximately 1% improvement relative to the error matrix results shown in Table 5.1.

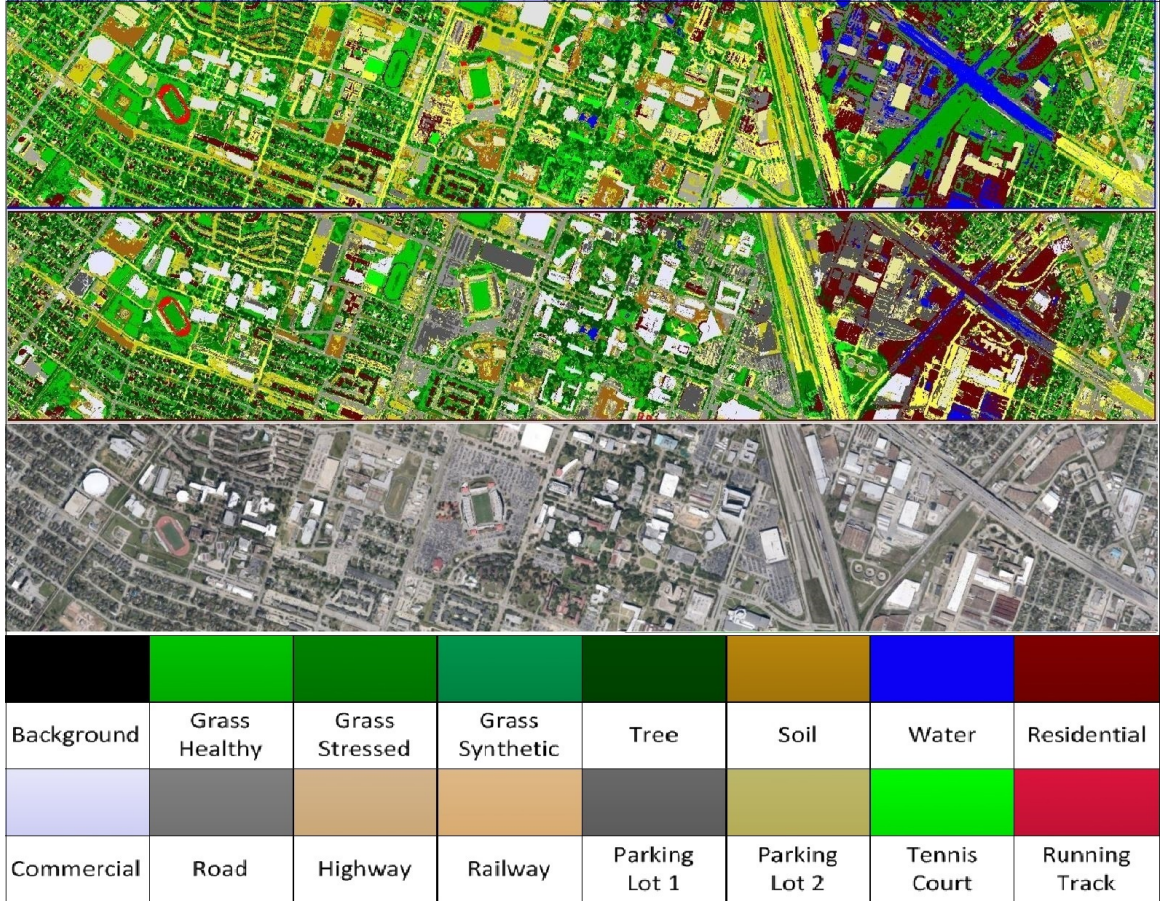


Figure 5.8: 2013_IEEE_GRSS_DF_Contest dataset: CNN using HSI data only (top) and fused HSI and LiDAR data (middle). Google map view (bottom).

Figure 5.9 displays the color-coded classes for MUUFL Gulfport dataset scene and presents the comparison of classification done by the CNN model using HSI data, Fused HSI LiDAR data and a Google map snapshot. Classes labeled Targets 1-4 correspond to the materials placed on the ground and additionally, smaller targets are also placed in the scene with their locations unknown. The presented model seems successful in locating a few of these targets, though not very clearly represented due to their relatively smaller sizes in the given spatial resolution of the scene.

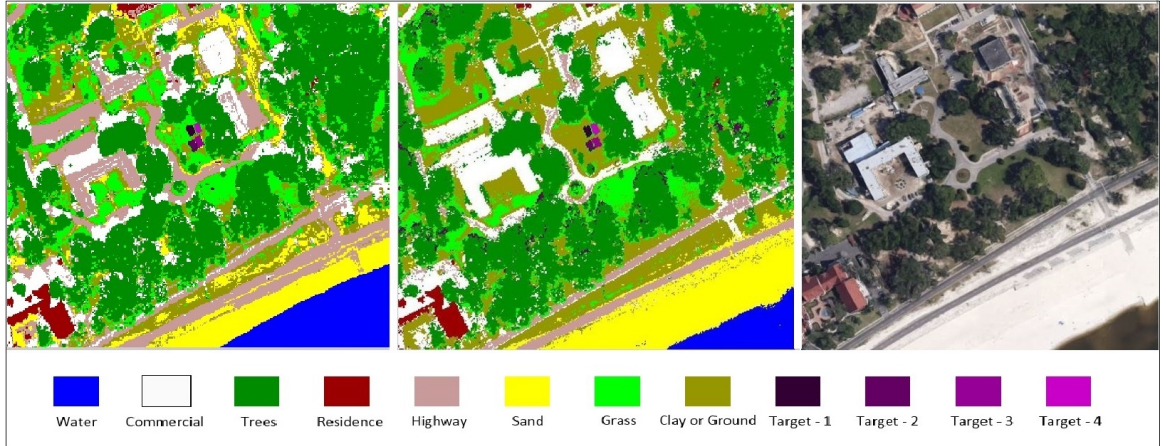


Figure 5.9: MUUFL Gulfport dataset: CNN using HSI data only (left) and fused HSI and LiDAR data (middle). Google map view (right).

5.3.3 Relevance of Fused LiDAR Data in Classification

Pixel-level fusion can introduce additional variability across HSI classes, as can be observed in Figure 5.10. The complementary nature of HSI and LiDAR enables this gain in variability across HSI classes. Notice how adding the LiDAR component significantly increases the distinction between commercial buildings and highways and between residential buildings and parking lot 1 spectral traces.

HSI Fused HSI LiDAR	Parking Lot 1	Residential	Commercial	Highway
Parking Lot 1	42.42% 52.27%	1.52% 0.00%	0.00% 5.30%	15.15% 3.03%
Residential	6.62% 0.00%	85.29% 89.71%	1.47% 4.41%	0.00% 0.00%
Commercial	3.05% 11.45%	0.76% 0.76%	54.20% 87.02%	2.29% 0.00%
Highway	16.79% 7.63%	0.00% 0.00%	0.00% 0.00%	62.60% 84.73%

Table 5.3: The classification accuracies of 4 classes with relatively similar spectral signatures in hyperspectral domain and high contrast in LiDAR values for 2013_IEEE_GRSS_DF_Contest dataset.

Table 5.3 bolsters the stated logic as the mis-classification accuracy declined from 6.62% to 0.00% for residential and from 2.29% to 0.00% for commercial when addi-

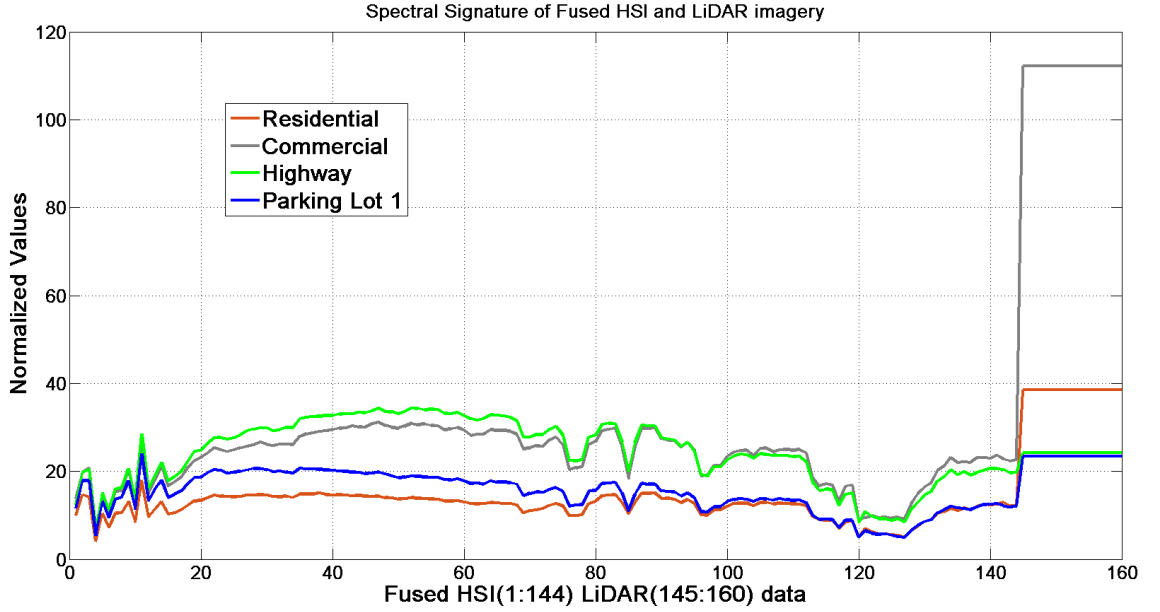


Figure 5.10: Fused data vectors for pixels with similar hyperspectral response.

tional LiDAR information is used. While for parking lot 1 and highway, where the LiDAR values are over-lapping, fused LiDAR has relatively more mis-classifications. Additional LiDAR information has significantly increased the overall accuracies for all the classes and largely reduced mis-classifications.

5.3.4 Relevance of Repetition of LiDAR Data in Fused Model

HSI Fused HSI LiDAR	Road	Highway	Parking Lot 1	Parking Lot 2
Road	51.88% 81.95%	36.84% 3.01%	3.01% 2.26%	0.75% 3.76%
Highway	13.74% 1.53%	62.60% 84.73%	16.79% 7.63%	0.00% 0.00%
Parking Lot 1	31.82% 31.82%	15.15% 3.03%	42.42% 52.27%	0.00% 6.82%
Parking Lot 2	16.94% 7.26%	4.84% 1.61%	11.29% 8.87%	45.97% 69.35%

Table 5.4: The classification accuracies of 4 classes with asphalt material and relatively similar spectral signatures from 2013_IEEE_GRSS_DF_Contest dataset.

Table 5.4 compares worst performing 4 classes based on classification accuracy when

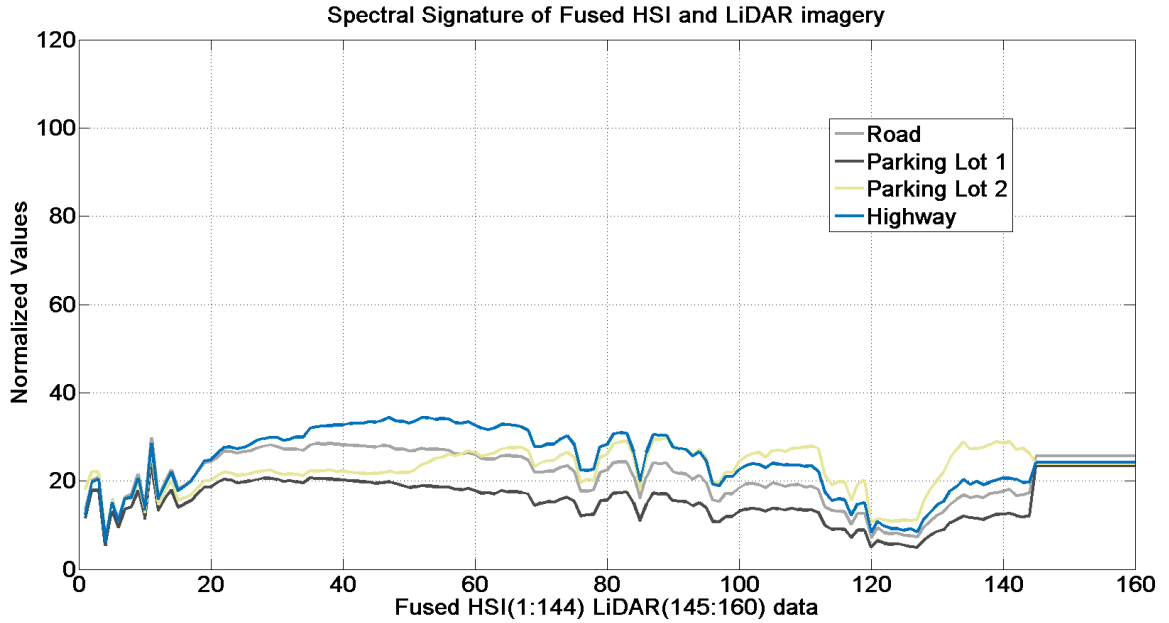


Figure 5.11: Fused data vectors for pixels with similar hyperspectral response and LiDAR values.

only hyperspectral data is used by CNN model for 2013_IEEE_GRSS_DF_Contest with classification accuracies when fused HSI LiDAR data is used. Figure 5.11 shows the spectral signature for the same classes with fused HSI LiDAR data. All these classes (Road, Highway, Parking Lot 1 and 2) have asphalt as a constituting material, which makes the spectral signature corresponding to hyperspectral data (1:144) very similar. When LiDAR data is fused with hyperspectral, it improves the classification for all the 4 classes indicated by substantial increase in precision and recall, and decline in miss-rate and false-discovery rate. In spite of very less relative differences in LiDAR values of these 4 classes, the fused LiDAR model has given very good improvement, which essentially seems to be an outcome of the repetition of LiDAR data.

Chapter 6: Conclusions and Future Works

The experimental results suggest that pixel-level data fusion can be an effective approach to improving accuracy of convolutional neural network classifiers. It is remarkable that data from disparate sources and in inherently different units can be concatenated (with appropriate replication and weighting) and used to improve the performance of a CNN classifier. The classification is improved by supplementing hyperspectral information with LiDAR, which compensates for the closeness of the spectral signatures in HSI domain by adding variance from the LiDAR data. Classes with similar hyperspectral spectral signatures and relatively less differences in LiDAR data, can be treated effectively by repeating LiDAR data in the fusion model i.e. giving more weight to that ‘less difference’.

It appears that the success of the reported results may be attributed in part to the inherent robustness of the neural network classifiers. This robustness is also evident when the performance of the CNN for most of the classes remained consistent on introduction of uncertainties in the training data. The CNN classifier for the fused hyperspectral LiDAR model performed equally well for 10 out of 15 classes, even on introduction of up to 30% error, while for the other 5 classes it gave accuracies similar to the CNN model using just hyperspectral data.

Future work includes exploitation of the natural correlation among nearby pixels. Nearby pixels may be expected to be highly correlated in their spectral signatures, their LiDAR elevations, and in their LiDAR intensities. Spatial correlation can be exploited for the classification in a two-step process in which the CNN model uses spectral domain for predicting classes, and then another classifier can improve these predicted classes by utilizing the spatial information. Alternatively, the CNN model can be extended to use 3-D convolution to take into account spectral information and spatial correlation simultaneously, and further

improve classification accuracy.

Additionally, a hierarchical classification model using a set of CNNs might be explored for this fusion model, which may further enhance the classification performance. The possibility of employing a hierarchical model utilizing same feature maps can also be explored, which necessarily reduces the memory requirements compared to a set of CNNs, and could give better classification performances.

Bibliography

- [1] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pizurica, S. Gautama, et al. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2405–2418, 2014.
- [2] M. Khodadadzadeh, J. Li, S. Prasad, and A. Plaza. Fusion of hyperspectral and lidar remote sensing data using multiple feature learning. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 8(6):2971–2983, 2015.
- [3] D. Nikic, J. Wu, V.P. Pauca, R. Plemmons, and Q. Zhang. A novel approach to environment reconstruction in lidar and hsi datasets. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, volume 1, page 81, 2012.
- [4] Q. Zhang, V. P. Pauca, R. J. Plemmons, and D. Nikic. Detecting objects under shadows by fusion of hyperspectral and lidar data: A physical model approach. In *Proc. 5th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens*, pages 1–4, 2013.
- [5] M. Dalponte, L. Bruzzone, and D. Gianelle. Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(5):1416–1427, 2008.
- [6] X. Ma, J. Geng, and H. Wang. Hyperspectral image classification via contextual deep learning. *EURASIP Journal on Image and Video Processing*, 2015(1):1–12, 2015.
- [7] F. P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj. Multiview deep learning for land-use classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2448–2452, Dec 2015.
- [8] Y. Chen, X. Zhao, and X. Jia. Spectral and spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, June 2015.
- [9] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013.
- [10] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2094–2107, 2014.

- [11] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, Vol. 2015, 2015.
- [12] K. Makantasis, K. Karantzas, A. Doulamis, and N. Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4959–4962, July 2015.
- [13] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, March 2016.
- [14] C. Pohl and J. L. Van Genderen. Review article multisensor image fusion in remote sensing: concepts, methods and applications. *International journal of remote sensing*, 19(5):823–854, 1998.
- [15] M. Mangolini. *Apport de la fusion d’images satellitaires multicateurs au niveau pixel en télédétection et photo-interprétation*. PhD thesis, Université de Nice Sophia-Antipolis, 1994.
- [16] M. T. Eismann. *Hyperspectral Remote Sensing*, volume PM210. SPIE.
- [17] Dr. N. M. Short. Graphic representation of hyperspectral data. NASA <http://rst.gsfc.nasa.gov/>, June 2007.
- [18] A. Picon, O. Ghita, P.F. Whelan, and P. Iriondo. Spectral and spatial feature integration for classification of non-ferrous materials in hyper-spectral data, Nov. 2009.
- [19] National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. Lidar 101: An introduction to lidar technology, data, and application, 2011.
- [20] Environmental Systems Research Institute (ESRI). Arcmap 10.3. <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-intensity-data-.htm>.
- [21] IEEE Geoscience and Remote Sensing Society 2013 Data Fusion Contest. <http://www.grss-ieee.org/community/technical-committees/data-fusion>.
- [22] J.R. Schott. *Remote sensing: the image chain approach*, volume 2. Oxford University Press New York, 2007.
- [23] A. Ross and A. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115 – 2125, 2003. Audio- and Video-based Biometric Person Authentication (AVBPA 2001).

- [24] J. R. Kaufman, M. T. Eismann, and M. Celenk. Assessment of spatial x2013;spectral feature-level fusion for hyperspectral target detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2534–2544, June 2015.
- [25] L. Dabbiru, S. Samiappan, R. A. A. Nobrega, J. A. Aanstoos, N. H. Younan, and R. J. Moorhead. Fusion of synthetic aperture radar and hyperspectral imagery to detect impacts of oil spill in gulf of mexico. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1901–1904, July 2015.
- [26] Q. Man, P. Dong, and H. Guo. Pixel-and feature-level fusion of hyperspectral and lidar data for urban land-use classification. *Int. J. Remote Sens.*, 36(6):1618–1644, March 2015.
- [27] D. Liu, H. Pu, D. Sun, L. Wang, and X. Zeng. Combination of spectra and texture data of hyperspectral imaging for prediction of ph in salted meat. *Food Chemistry*, 160:330 – 337, 2014.
- [28] L. He, Q. Pan, Y. Zhao, and W. Di. An automatic target detection algorithm for hyperspectral imagery based on feature-level fusion, 2005.
- [29] J. A. Benediktsson, J. A. Palmason, J. R. Sveinsson, and J. Chanussot. Decision level fusion in classification of hyperspectral data from urban areas. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International*, volume 1, page 76, Sept 2004.
- [30] Z. Ye, S. Prasad, W. Li, J. E. Fowler, and M. He. Classification based on 3-d dwt and decision fusion for hyperspectral image analysis. *IEEE Geoscience and Remote Sensing Letters*, 11(1):173–177, Jan 2014.
- [31] Y. Nanyam, R. Choudhary, L. Gupta, and J. Paliwal. A decision-fusion strategy for fruit quality inspection using hyperspectral imaging. *Biosystems Engineering*, 111(1):118 – 125, 2012.
- [32] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1847–1871, 2013.
- [33] E. W. Weisstein. "hyperbolic tangent." from mathworld—a wolfram web resource. <http://mathworld.wolfram.com/HyperbolicTangent.html>.
- [34] P. Gader, A. Zare, R. Close, and G. Tuell. Co-registered hyperspectral and lidar long beach, mississippi data collection. 2010. University of Florida, University of Missouri, and Optech International.

- [35] P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell. MUUFL gulfport hyperspectral and LiDAR airborne data set. Tech. rep. REP-2013-570, University of Florida, Oct. 2013.
- [36] R. B. Palm. Prediction as a candidate for learning deep hierarchical models of data. Master's thesis, Technical University of Denmark, DTU Informatics, 2012.

Vita

Personal Details

Name : Saurabh Morchhale
Address : Department of Computer Science, Wake Forest University,
Winston Salem, NC-27106, US
Phone : 336.830.6944
Email : saurabh.morchhale@gmail.com

Education

May 2016 Master of Science in COMPUTER SCIENCE, Wake Forest University
Area of Research: Machine Learning and Computer Vision
Thesis: “Deep Convolutional Neural Networks for Classification of
Hyperspectral and LiDAR data”
Advisor: Dr. V. Paúl Pauca
GPA: 3.54 / 4.0

June 2011 Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING,
Rajiv Gandhi Proudhyogiki Vishwavidyalaya
Advisor: Dr. Ujjwal Nigam
Percentage: 74.59 %

Work Experience

August 2015 - May 2016 Teaching Assistant for the Department of Computer Science
at Wake Forest University, NC

May 2015 - August 2015 Research Assistant for the Department of Computer Science
at Wake Forest University, NC

August 2014 - May 2015 Research Assistant for the Wake Forest Institute for
Regenerative Medicine, NC

August 2011 - July 2014 Software Engineer for Computer Sciences Corporation
(CSC) India Pvt. Ltd. , India