On the Importance of 3D Surface Information for Remote Sensing Classification Tasks

JAN PETRICH [ID]

RYAN SANDER [ID]

ELIZA BRADLEY [ID]

ADAM DAWOOD

SHAWN HOUGH

*Author affiliations can be found in the back matter of this article

]u[ubiquity press

ABSTRACT

There has been a surge in remote sensing machine learning applications that operate on data from active or passive sensors as well as multi-sensor combinations (Ma et al. (2019)). Despite this surge, however, there has been relatively little study on the comparative value of 3D surface information for machine learning classification tasks. Adding 3D surface information to RGB imagery can provide crucial geometric information for semantic classes such as buildings, and can thus improve out-of-sample predictive performance. In this paper, we examine in-sample and out-of-sample classification performance of Fully Convolutional Neural Networks (FCNNs) and Support Vector Machines (SVMs) trained with and without 3D normalized digital surface model (nDSM) information. We assess classification performance using multispectral imagery from the International Society for Photogrammetry and Remote Sensing (ISPRS) 2D Semantic Labeling contest and the United States Special Operations Command (USSOCOM) Urban 3D Challenge. We find that providing RGB classifiers with additional 3D nDSM information results in little increase in in-sample classification performance, suggesting that spectral information alone may be sufficient for the given classification tasks. However, we observe that providing these RGB classifiers with additional nDSM information leads to significant gains in out-of-sample predictive performance. Specifically, we observe an average improvement in out-of-sample all-class accuracy of 14.4% on the ISPRS dataset and an average improvement in out-of-sample F1 score of 8.6% on the USSOCOM dataset. In addition, the experiments establish that nDSM information is critical in machine learning and classification settings that face training sample scarcity.

CORRESPONDING AUTHOR:
Jan Petrich

Pennsylvania State University Applied Research Laboratory, United States

jup37@arl.psu.edu

1. INTRODUCTION

Significant resources are required to establish robust classification performance for remote sensing applications which can range from composite distributed land uses, for example urban versus rural development, to feature-specific mapping, such as roads, building and vehicles. Technical challenges naturally arise from the sheer variety of appearances and viewing angles for objects of interest, as well as the lack of annotations or labels relative to the overall volume of data.

Machine learning and artificial intelligence have made great strides in recent years due to the advent of hardware accelerated computing and the development of symbolic math libraries. However, a unified framework for remote sensing classification applications is still beyond reach (Ball, Anderson, and Chan, 2017; Zhu et al., 2017). The diversity of data collection methods, e.g. space-based or airborne, and sensor modalities, e.g. lidar RGB, and hyperspectral imagery, necessitate continuous adaptation of the underlying classification framework based on classification task, desired accuracy, computational bandwidth, and available datasets.

A range of machine learning methodologies are available for executing object classification from remotely sensed data including convolutional neural networks (CNNs) (Maturana and Scherer, 2015; R. Zhao, Pang, and J. Wang, 2018), decision trees (Blaha et al., 2016), and Support Vector Machines (SVMs) (Lai and Fox, n.d.; D. Wang, Zhang, and Y. Zhao, 2007). Although inherently different, all of those methods have been shown to be effective and successful given specific datasets and class labels for training and cross-validation. Interestingly, many algorithms originate from outside the remote sensing community, e.g. from biomedical image segmentation (Ronneberger, Fischer, and Brox, 2015), and have been modified to ingest mission-specific sensor modalities in order to classify different objects of interest.

The feasibility of applying machine learning concepts has been further studied, and showcased through various challenges, such as the USSOCOM Urban 3D challenge (H. Goldberg, Brown, and S. Wang, 2017; H.R. Goldberg et al., 2018) or the ISPRS 2D and 3D Semantic Labeling contests (Gerke et al., 2014). Several submitted algorithms yielded high accuracy, even without using 3D surface information (Audebert, Le Saux, and Lefèvre, 2018). However, given the recent success, it remains important to formally assess information content of available sensor modalities, and to infer generalizability of the resulting classifier if presented with inherently different, previously unseen test samples. In other words, given a specific object recognition task, we seek to answer the following questions. First, what are the preferred sensor modalities given the quantity and quality of training samples? After all, adding sensor modalities usually incurs cost and logistical considerations. Second, is the resulting classifier useful when applied outside the location or domain in which training data was collected?

This paper expands upon the findings in (Chen, Fu, et al., 2018), in which the authors acknowledge that the majority of ongoing work is focused on developing classification strategies with little regard to the information value provided by the sensor modality of the input data. In (Chen, Fu, et al., 2018), multi-spectral imagery is fused with DSM information in order to study the impact on the resulting classification performance. However, DSM information is pre-processed to extract specific features based on the 3D structure tensor (Weinmann, n.d.) such as linearity, planarity, and sphericity, as well as the Normalized Difference Vegetation Index (NDVI) from (Tucker et al., 2001). This paper expands upon the work in (Chen, Fu, et al., 2018) by considering multiple datasets and by examining training and test performance separately. Additionally, cross-city validation is added to formally assess classification performance when extrapolating the classifier from one city to another.

To partially close this gap between formally assessing information content and achieving high classification performance, this paper examines the importance of 3D surface information when generalizing trained classifiers using two distinct machine learning frameworks, and two different publicly available datasets as test cases. In order to derive results that are agnostic to the chosen classification methodology, we specifically select one classification scheme of high complexity, i.e. a Fully Convolutional Neural Network (FCNN) architecture, and one classification scheme of low complexity, i.e. a Support Vector Machine (SVM).

The novelties and technical contributions of this paper are two-fold. First, we examine generalizability of classification frameworks when trained and tested on different geographic

locations by formally evaluating out-of-sample performance of the classifier when trained with and without 3D DSM information. Different geographic locations often imply deviations in the underlying urban infrastructure, architecture, and thus the appearance and structure of objects of interest such as buildings and roads. Second, we assess the same generalizability when facing scarce training data. Therefore, we will formally evaluate out-of-sample performance when trained with and without 3D DSM information while the number of training samples is gradually reduced.

As acknowledged in (Chen, Weinmann, et al., 2018), the majority of research focuses on improving classification tasks, while "only little attention has been paid to the input data itself". Specifically, information content is often overlooked in light of achievable performance metrics for object classification and segmentation. In addition, the work in (Chen, Weinmann, et al., 2018) explores the utility of handcrafted, geometric features over machine-derived ones. The information content of specific feature types is studied in (Gevaert et al., 2016), in which the authors distinguish between point-based and segment-based features. Dependent on the available sensor modalities, only some feature types may be available in any given scenario. Their relative importance for object classification can then be formally assessed using an SVM classifier for different scenarios and different combinations of feature types. Although classification is tested via cross-validation, training and testing samples are extracted from the same geographical location, which may limit the diversity of building footprints the classifier is exposed to. This paper follows a similar approach by formally evaluating not only classification performance, but also generalizability using two distinct classification frameworks. Therefore, we attempt to establish the notion of information content independent of the underling classification strategy. In addition, our study focuses on entire sensor modalities instead of individual features that are derived from those modalities.

This paper is organized as follows: Section 2 outlines the technical approach. Experimental validation is presented in Section 3, while results are split into cross-city validation in Section 3.1 and validation for varying training sample proportions in Section 3.2. Section 4 concludes the study.

2. TECHNICAL APPROACH

This section describes the underlying classification approach and performance assessment. Section 2.1 provides an overview of binary and multi-class classification in a semantic segmentation setting. Section 2.2 discusses class balancing for imbalanced classification tasks. Section 2.3 outlines the two classification architectures used for this study, i.e. a Support Vector Machine (SVM) and a Fully Convolutional Neural Network (FCNN). The two publicly available datasets, including objects of interest used for training and testing, are described in Section 2.4. Section 2.5 summarizes the performance assessment strategies as well as the two different validation settings: (1) cross-city validation and (2) reduction of sample proportion for training.

2.1. BINARY AND MULTI-CLASS CLASSIFICATION

This work aims to (i) develop robust binary and multi-class semantic segmentation classifiers for remote sensing, and (ii) test the overall generalizability of these classifiers as a function of sensor modality. Semantic segmentation can be conceptualized as a classification problem in which we predict a class label for each pixel. We consider each of these pixel-level classification problems in binary and multi-class classification settings (Janocha and Czarnecki, 2017). One common optimization objective for finding optimal parameter sets in binary and multi-class classification tasks is two-dimensional cross-entropy, which we utilize in this work for our FCNN models. Below, we discuss the objective functions for both of these classification schemes.

2.1.1. Binary Classification

For binary classification, the classifier learns by minimizing a loss function over a set of parameters θ_b^*, in this case pixel-wise Negative Log Likelihood (NLL) with a single class.

(2.1) $$\theta_b^* = \arg \min_{\theta \in \Theta} \left(-\sum_{i=1}^{m} \sum_{j=1}^{n} Y_{i,j} \log(P(X;\theta)_{i,j}) + (1 - Y_{i,j}) \log(1 - P(X;\theta)_{i,j}) \right)$$

Where $Y_{i,j} \in \{0, 1\}$ and $P(X;\theta)_{i,j} \in [0, 1]$ denote the ground truth and predicted labels, respectively, at pixel (i, j) for each training image tile, and X denotes the input features (1, 3, or 4-channel image). Both X and Y are indexed as 2D arrays with (height, width) given as (m, n). This classification methodology is utilized for the USSOCOM Urban3D dataset for building footprint classification.

2.1.2 Multiclass Classification

A multiclass classifier minimizes a similar objective to find parameters θ_m^*, except rather than considering a single class, e.g. a building, this classifier makes pixel-wise predictions over multiple classes. This classifier learns by minimizing the following objective, which corresponds to pixel-wise Negative Log Likelihood (NLL) over N classes.

$$(2.2) \qquad \theta_m^* = \arg\min_{\theta \in \Theta} \left(-\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{N} Y_{i,j,k} \log(P(X;\theta)_{i,j,k}) \right)$$

Where $Y_{i,j,k} \in \{0, 1\}$ and $P(X, \theta)_{i,j,k} \in [0, 1]$ denote the ground truth and predicted labels, respectively, at pixel (i, j) for class $k \in \{0, 1, ..., N\}$ for each training tile X, which is indexed as a 2D array of shape (m, n) as above.

To make optimal semantic segmentation predictions, we use gradient descent methods (Ruder, 2016) to iteratively solve for a set of weights θ that minimize the objectives presented above.

2.2. CLASS IMBALANCE CORRECTION VIA CLASS WEIGHTING

In addition to computing element-wise cross-entropy loss for each pixel in the image, we weight each pixel using its inverse class frequency, i.e. if class i occurs in a dataset with fraction $f_i \in (0, 1]$, then our weighting corresponds to the inverse of this fraction: $w_i = \frac{1}{f_i}$. This weighting scheme is used to correct for class imbalance in our datasets, and corresponds to a weighted Negative Log Likelihood (wNLL) or weighted Cross-Entropy objective (Aurelio et al., 2019). Using a weighted cost function allows for learning a classifier that is not biased toward the most common class. With the incorporation of these weights, our objectives become:

Weighted Binary Classification:

$$(2.3) \qquad \theta_{b,w}^* = \arg\min_{\theta \in \Theta} \left(-\sum_{i=1}^{m} \sum_{j=1}^{n} w(Y_{i,j})(Y_{i,j} \log(P(X;\theta)_{i,j}) + (1 - Y_{i,j}) \log(1 - P(X;\theta)_{i,j})) \right)$$

Where $w(Y_{i,j}) \in \mathbb{R}^+$ denotes the weight assigned to pixel (i,j) based off of its ground truth class, $Y_{i,j}$. Note that $Y_{i,j} \in \{0, 1\}$.

Weighted Multi-Class Classification:

$$(2.4) \qquad \theta_{m,w}^* = \arg\min_{\theta \in \Theta} \left(-\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{N} w(c_{i,j}) Y_{i,j,k} \log(P(X;\theta)_{i,j,k}) \right)$$

Where $w(c_{i,j}) \in \mathbb{R}^+$ denotes the weight assigned to pixel (i,j) based off of its ground truth class $c_{i,j}$. Note that here we use $c_{i,j} \in \{0, 1, ..., N\}$ to denote the class, rather than $Y_{i,j,k} \in \{0, 1\}$.

Our FCNN classifiers, SegNet and SegNet Lite, which are described in detail below, each leverage these weighted objectives, and our classification results are reported as balanced/class frequency adjusted.

2.3 CLASSIFICATION ARCHITECTURES

To fully evaluate the importance of 3D nDSM information independent of the underlying classification algorithm, two distinct classification architectures have been selected for experimentation. First, we consider a relatively simple, well-known SVM framework. Second, we apply an FCNN architecture. These two classifers are outlined in detail below.

2.3.1 Architecture 1 – Support Vector Machine (SVM)

SVM is a discriminative classifier that aims to separate sets of feature vectors based on the assigned labels by finding the maximum margin hyperplane (Cortes and Vapnik, 1995). Separation can be accomplished through linear hyperplanes or nonlinear hypersurfaces, which are constructed in a feature space through numerical optimization with kernel functions. *Figure 1* illustrates the application of a 5 × 5 neighborhood in the imagery to construct n-dimensional feature vectors $fi \in \mathbb{R}^n$, one for each pixel (or data sample) i. For case 1 (one channel), we extract features only from the DSM information to obtain a feature vector $f_{1,i} \in \mathbb{R}^{25}$. Similarly, for case 2 (three channels), we utilize only RGB information to obtain $f_{2,i} \in \mathbb{R}^{75}$. For case 3 (four channels), we find $f_{3,i} \in \mathbb{R}^{100}$ by concatenating $f_{1,i}$ and $f_{2,i}$, thereby constructing the full feature vector fusing DSM and RGB information. Therefore, classification will be carried out in 25, 75 or 100-dimensional feature spaces.
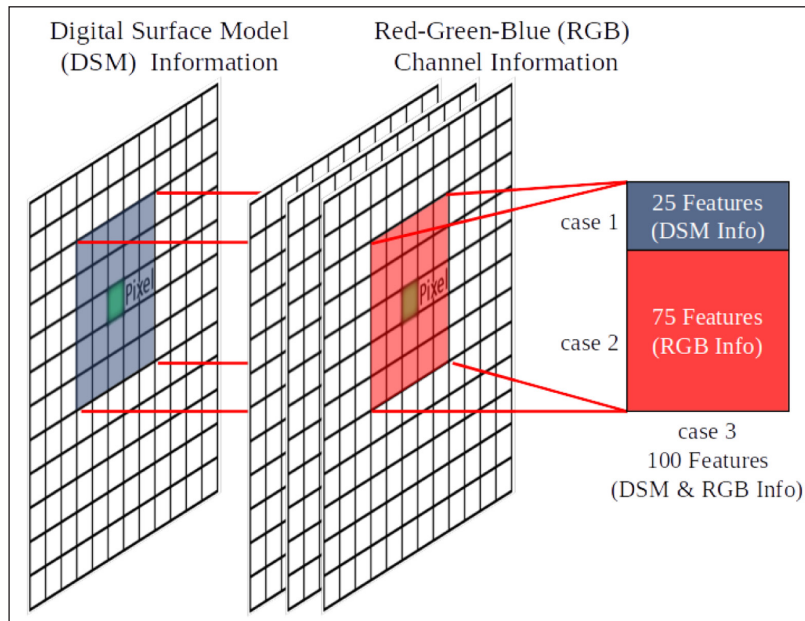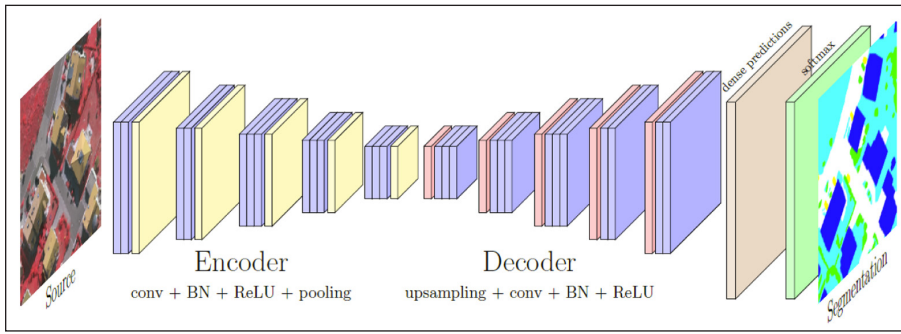
Based on ground truth information and training data available, labels $y_i \in L$ can be assigned to each feature vector f_i. Here, L denotes the set of all labels with cardinality $|L| = N$, where N is the number of unique labels contained in the dataset. Once feature vectors have been constructed and labels have been assigned, supervised training via SVM can be accomplished using any available machine learning library. For this study, nonlinear hypersurfaces are constructed using a Gaussian/Radial Basis Function (RBF) kernel, implemented with MATLAB's `fitcecoc` module (MathWorks, 2018).

2.3.2 Architecture 2 – Fully Convolutional Neural Network (FCNN)

Although originally developed for image segmentation tasks, the Segmentation Network, or SegNet, architecture presented in (Badrinarayanan, Kendall, and Cipolla, 2017) and depicted in *Figure 2* has recently gained increased attention for object classification in remote sensing applications (Audebert, Le Saux, and Lefèvre, 2018). SegNet is characterized by a Fully Convolutional Neural Network (FCNN) architecture with an encoder-decoder structure. Image information is initially down-sampled throughout the five encoder blocks (left side of *Figure 2*) using convolution operations, batch normalization, nonlinear activation functions, and pooling. These encoding blocks create a latent representation of the input image, characterized by spatial features extracted from the convolutional encoding layers. Throughout the five decoding blocks (right side of *Figure 2*), segmentation information is reconstructed from this latent representation to the full image resolution using convolution operations, batch normalization, nonlinear activation functions, and nonlinear up-sampling blocks. To perform nonlinear up-sampling, the SegNet decoder leverages pooling indices computed in the encoder layers of the network and connected from the encoder to the decoder via skip connections (Badrinarayanan, Kendall, and Cipolla, 2017). The input layer can be modified to ingest one channel (DSM only), three channel (RGB-only), or four channel (DSM & RGB) data.

Each of the five encoder blocks and five decoder blocks consists of two to three convolutional layers. For an input window size of 256 × 256 pixels, it can be shown that the original SegNet structure from (Badrinarayanan, Kendall, and Cipolla, 2017) consists of roughly 30 million weights. By limiting the number of convolutional layers per block to two and reducing the output dimensions (or channels) for each layer by 75 percent, we construct a similar, yet lightweight SegNet architecture consisting of only 1.2 million weights, reducing the total number of weights by 96%. For experiments carried out in Section 3, we will refer to the original SegNet architecture as *SegNet* (`SegNet`) and the lightweight SegNet structure with a reduced number of weights as *SegNet Lite* (`SegNet Lite`). *Figure 3* compares our `SegNet` (red) and `SegNet Lite` (blue) architectures, listing the number of weights per layer and the total number of weights. *Table 1* provides a consolidated view of these two architectures.

	LAYER	INDEX	BLOCK	IN_CHANNELS		OUT_CHANNELS		KERNEL SIZE	Number of Weights (4D Tensor)	
				SegNet Lite	SegNet	SegNet Lite	SegNet		SegNet Lite	SegNet
Encoder incl. pooling	1	1	Conv2d	3	3	16	64	3	432	1,728
		2	Conv2d	16	64	16	64	3	2,304	36,864
	2	1	Conv2d	16	64	32	128	3	4,608	73,728
		2	Conv2d	32	128	32	128	3	9,216	147,456
	3	1	Conv2d	32	128	64	256	3	18,432	294,912
		2	Conv2d	64	256	64	256	3	36,864	589,824
		3	Conv2d	removed	256	removed	256	3	-	589,824
	4	1	Conv2d	64	256	128	512	3	73,728	1,179,648
		2	Conv2d	128	512	128	512	3	147,456	2,359,296
		3	Conv2d	removed	512	removed	512	3	-	2,359,296
	5	1	Conv2d	128	512	128	512	3	147,456	2,359,296
		2	Conv2d	128	512	128	512	3	147,456	2,359,296
		3	Conv2d	removed	512	removed	512	3	-	2,359,296
Decoder incl. unpooling	5	1	Conv2d	128	512	128	512	3	147,456	2,359,296
		2	Conv2d	128	512	128	512	3	147,456	2,359,296
		3	Conv2d	removed	512	removed	512	3	-	2,359,296
	4	1	Conv2d	128	512	128	512	3	147,456	2,359,296
		2	Conv2d	128	512	64	512	3	73,728	2,359,296
		3	Conv2d	removed	512	removed	256	3	-	1,179,648
	3	1	Conv2d	64	256	64	256	3	36,864	589,824
		2	Conv2d	64	256	32	256	3	18,432	589,824
		3	Conv2d	removed	256	removed	128	3	-	294,912
	2	1	Conv2d	32	128	32	128	3	9,216	147,456
		2	Conv2d	32	128	16	64	3	4,608	73,728
	1	1	Conv2d	16	64	16	64	3	2,304	36,864
		2	Conv2d	16	64	6	6	3	864	3,456
						Total Number of Weights			1,176,336	29,422,656

| NEURAL ARCHITECTURE	TOTAL PARAMETERS	CHANNELS (RELATIVE TO SEGNET)	KERNEL SIZE
SegNet	29,422,656	1.0×	3
SegNet Lite	1,176,336	0.25×	3

Both of these SegNet neural architecture variants were implemented using PyTorch (Paszke et al., 2019), which supports streamlined GPU acceleration. Source code was provided though a git repository[1] shared by the authors of (Audebert, Le Saux, and Lefèvre, 2018). Some modifications were made (i) to ingest composite images combining spectral and DSM information, and (ii) to transform the SegNet (`SegNet`) architecture into the SegNet Lite (`SegNet Lite`) architecture.

---

1   *https://github.com/nshaud/DeepNetsForEO*.

2.4. DATASETS AND OBJECTS OF INTEREST

This section describes the two datasets used for performance assessment, including objects of interest. Although there exists a wide variety of available datasets for examining remote sensing applications, this work focuses on (i) high-resolution satellite imagery provided through the USSOCOM Urban 3D Challenge (H. Goldberg, Brown, and S. Wang, 2017; H.R. Goldberg et al., 2018) and (ii) aerial imagery released by the International Society for Photogrammetry and Remote Sensing (ISPRS) in support of their 2D Semantic Labeling Contest (Gerke et al., 2014). In this paper, we refer to these datasets by the originator (e.g. USSOCOM for the Urban 3D Challenge and ISPRS for the ISPRS 2D Semantic Labeling Contest). To establish the core results of this study, we apply both the SVM and SegNet classifiers to the ISPRS and USSOCOM datasets.

2.4.1. USSOCOM Urban3D Dataset

The USSOCOM dataset contains orthorectified red-green-blue (RGB) imagery of three US cities: (i) Jacksonville, FL, (ii) Tampa, FL, and (iii) Richmond, VA with a resolution of 50-centimeters ground sample distance (GSD). The data was collected via commercial satellites and additionally provides coincident 3D Digital Surface Models (DSM) as well as Digital Terrain Models (DTM). These DSMs and DTMs are derived from multi-view, EO satellite imagery at 50-centimeter resolution, rather than through lidar (Light Detection and Ranging) sensors. DSM and DTM information is used to generate normalized DSM (nDSM) information, i.e. nDSM = DSM – DTM. All imagery products were created using the Vricon (now Maxar Technologies) production pipeline using 50-centimeter DigitalGlobe satellite imagery. Buildings are the only objects of interest, i.e. $L = \{0, 1\}$ and $|L| = 2$, with roughly 157,000 annotated building footprints contained in the data. These ground truth building footprints are generated through the use of a semi-automated feature extraction tool in tandem with the HSIP 133 cities dataset (H. Goldberg, Brown, and S. Wang, 2017). *Figure 4* shows one of the 144 Jacksonville tiles from the USSOCOM dataset with RGB imagery on the left, nDSM information (i.e. the difference between surface and terrain) in the center, and annotated ground truth for building footprints on the right (buildings in yellow with background in blue). Tiles are 2048 × 2048 pixels and span an area of roughly 1km². *Figure 5* depicts an nDSM of one of the Jacksonville tiles from the USSOCOM dataset (H. Goldberg, Brown, and S. Wang, 2017; H.R. Goldberg et al., 2018) shown from a different point of view. More information about the USSOCOM Urban3D dataset can be found in (H. Goldberg, Brown, and S. Wang, 2017) and (H.R. Goldberg et al., 2018).
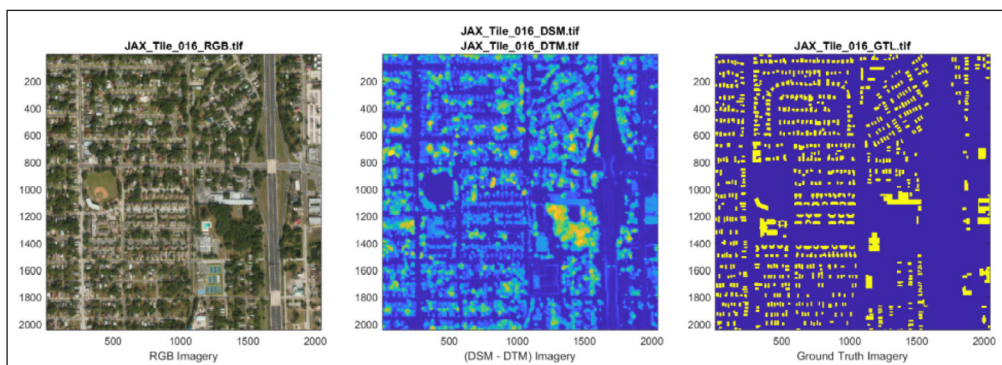

Figure 4 Sample tile from USSOCOM Urban 3D Challenge dataset for Jacksonville, FL showing RGB imagery (left), nDSM info (center), and annotated ground truth for building footprints (right).
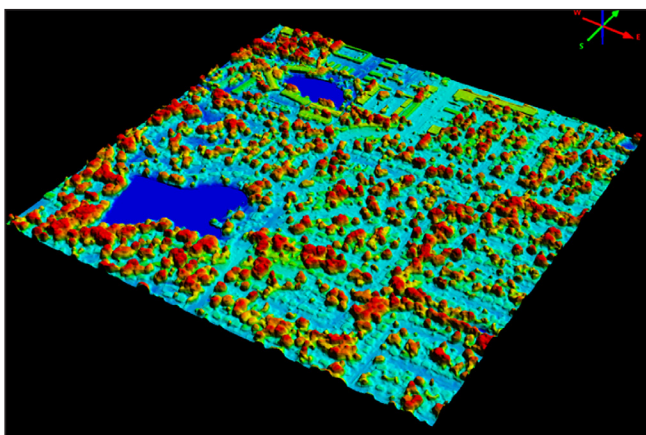
Figure 5 Another view of a sample nDSM (Jacksonville Tile 23) from the USSOCOM dataset.

2.4.2 ISPRS Semantic Labeling Dataset

The ISPRS dataset contains infrared, red, and green (IRRG) bands for two locations: (i) Vaihingen, Germany, and (ii) Potsdam, Germany. The GSDs are 9 centimeters and 5 centimeters, respectively. DSM and nDSM information is generated via dense image matching using Trimble INPHO 5.3 software. In order to avoid areas without data ('holes') in the True Orthophoto (TOP) and DSM, dataset patches were selected only from the central region of the TOP mosaic, i.e. not at the boundaries. Any remaining (very small) holes in the TOP and the DSM were interpolated. nDSM imagery is produced using a fully automatic filtering workflow without any manual quality control. 32-bit grey levels are used to encode heights for the DSMs and nDSMs in the TIFF format (Gerke et al., 2014). Ground truth annotations for objects of interest are provided for $|L| = 6$ classes, i.e. impervious surfaces (i.e. roads), buildings, low vegetation, trees, cars, and clutter. *Figure 6* presents a sample of the ISPRS dataset with IRRG imagery on the left, nDSM information in the center, and ground truth information on the right. Ground truth is color-coded for roads (white), buildings (blue), low vegetation (cyan), trees (green), cars (yellow) and clutter (red).
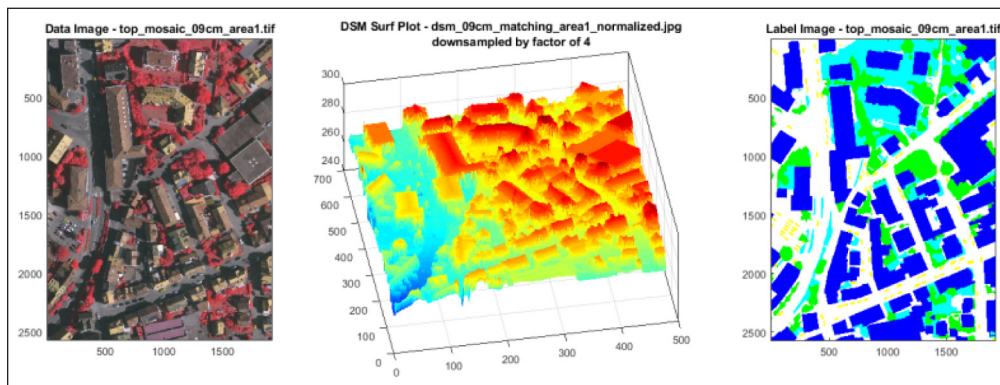
Figure 6 Sample tile from the ISPRS dataset for Vaihingen, Germany showing IRRG imagery (left), nDSM information (center), and color-coded ground truth for six object classes of interest (right).

For nDSM imagery produced in both the ISPRS and USSOCOM datasets, it is unclear if the authors leveraged any techniques to mitigate occlusion due to objects such as trees, which can have a substantial, seasonally-dependent effect on the ground-truth accuracy of the semantic labels (Park and Guldmann, 2019) present in both of these datasets. For the USSOCOM dataset, the authors also note that producing DTM imagery products from overhead imagery remains an open research question (H. Goldberg, Brown, and S. Wang, 2017).

2.5 VALIDATION SETTINGS

This study provides experimental validation for two scenarios demonstrating the importance of 3D surface information for remote sensing classification tasks.

First, we seek to establish performance metrics for cross-city validation when using classifiers that were trained with and without nDSM information. Previous work in this domain concluded that in-sample performance drops only slightly when depriving the classifier (Segnet in (Audebert, Le Saux, and Lefèvre, 2018)) of nDSM information. However, the impact of nDSM information on out-of-sample performance, i.e. cross-city performance, has not been studied and formally assessed yet, and is one of the major contributions of this work.

In addition to cross-city validation, we study the impact of nDSM information when training the classifier using scarce data. Therefore, we will reduce the number of training samples while assessing overall out-of-sample performance of the resulting classifier when trained both with and without nDSM information.

Table 2 summarizes the cross-city training and testing methodologies for the USSOCOM dataset, while *Table 3* summarizes the cross-city training and testing methodologies for the ISPRS dataset. As noted, we establish the core results of this study by evaluating the importance of DSM information for classification. For this, we use both the SVM and SegNet classifiers for both the USSOCOM and ISPRS datasets.

	CLASSIFIER ARCHITECTURE		
TYPE OF DATASET	SVM	SEGNET	
Training	Jacksonville, FL	Tampa, FL	Tampa, FL
In-Sample Testing	Jacksonville, FL	–	–
Out-of-Sample Testing	Tampa, FL	Richmond, VA	Jacksonville, FL

Table 2 USSOCOM training and testing (in-sample and out-of-sample) procedures for SVM and SegNet. For evaluating the SegNet classifier on the USSOCOM dataset, we only test out-of-sample performance.

	CLASSIFIER ARCHITECTURE		
TYPE OF DATASET	SVM	SEGNET LITE	SEGNET
Training	Vaihingen tiles 1–12	Vaihingen tiles 1–12	Vaihingen tiles 1–12
In-Sample Testing	Vaihingen tiles 13–16	Vaihingen tiles 13–16	Vaihingen tiles 13–16
Out-of-Sample Testing	Potsdam*	Potsdam*	Potsdam*

Table 3 ISPRS training and testing (in-sample and out-of-sample) procedures for our classification architectures: SVM, SegNet Lite, and SegNet.

*It is important to note that the ISPRS – Potsdam data (5 centimeters GSD) will be down-sampled by a ratio of 9:5 to achieve a GSD of 9 centimeters, which is used for training using the ISPRS – Vaihingen data.

In addition to what is summarized in *Tables 2* and *3*, these training and evaluation procedures also allow for formal comparisons between the SVM and SegNet classifiers on both the USSOCOM and ISPRS datasets.

3. EXPERIMENTAL VALIDATION

This section summarizes the experimental validation including performance assessment results. The section is divided with Section 3.1 addressing cross-city validation and Section 3.2 describing the impact of reducing available training samples. Classification performance with and without DSM information is the main focus of this study.

3.1 CROSS-CITY VALIDATION

Our first experimental study applies SVM and SegNet classifiers to the ISPRS dataset. Specifically, both networks were trained using three cases: (i) IRRG information only, (ii) nDSM information only, and (iii) nDSM & IRRG information combined. Training was conducted using 12 out of 16 tiles from the ISPRS Vaihingen dataset. Training times were approximately 20 hours for the SegNet and 3 hours and 20 minutes for SegNet Lite.

In our second experimental study, we apply SVM and SegNet classifiers to the USSOCOM dataset. For SVM, we first train the SVM classifier from Section 2.3.1 on the USSOCOM dataset using all of the available 144 Jacksonville tiles. Training samples were down-selected randomly by a factor of 1,000 in order to circumvent memory limitations, reduce the number of resulting support vectors, and to allow for adequate training and inference times. SVM training then yields three classifiers, one for training with RGB & nDSM information, one for training with RGB information only, and one for training with nDSM information only. For SegNet, we follow a similar training procedure: we train the SegNet classifier from Section 2.3.2 on all 144 available Tampa tiles. Down-selection is not performed for the SegNet classifier, i.e. the classifier is trained on all pixels present in all training tiles. SegNet training similarly yields three classifiers, one for training with RGB & nDSM information, one for training with RGB information only, and one for training with nDSM information only.

Results from these two aforementioned experimental studies are discussed below.

3.1.1 ISPRS Dataset Results

The results outlined in *Tables 4* and *5* replicate the findings from (Audebert, Le Saux, and Lefèvre, 2018). These tables summarize resulting classification accuracy for the SegNet and SegNet Lite architectures, respectively. Individual rows are associated with different

objects of interest, while columns cite the corresponding in-sample (Vaihingen) and out-of-sample (Potsdam) class-balanced accuracies for the three training cases. Note that random guessing would yield a total accuracy of $1 : 6 \approx 0.1667$ for a six class classification problem. Our results indicate that for SegNet, in-sample classification performance is not impacted significantly when depriving the classifier of nDSM information. In fact, accuracy drops less than 0.5% for either SegNet classifier between the nDSM & IRRG case and the IRRG-only case. For SVM, we observe a more significant drop in in-sample accuracy of 9% between the nDSM & IRRG case and the IRRG-only case. However, unlike in-sample performance, total accuracy drops for out-of-sample validation by 25%, from 65% to 40% for the SegNet, by 5%, from 50% to 45% for SegNet Lite, and 13%, from 54% to 41% for SVM, when excluding nDSM information from training. Performance losses are noticeable across all objects of interest. Although the nDSM-only classifier performs worst in-sample, for SegNet, it outperforms the IIRG-only classifier by 8% out-of-sample, and for SegNet Lite, it outperforms the IRRG-only classfier by 5% out-of-sample. For comparison, *Table 6* lists the performance metrics when using the SVM classifier for the ISPRS datasets. As expected, overall performance drops significantly.

OBJECTS OF INTEREST	SEGNET (ISPRS)					
	VAIHINGEN			POTSDAM (9 CM)		
	NDSM	IRRG	NDSM & IRRG	NDSM	IRRG	NDSM & IRRG
Impervious surfaces	0.8727	0.9520	**0.9531**	0.7127	0.7502	**0.8374**
Buildings	0.9549	0.9738	**0.9722**	0.6828	0.4571	**0.7886**
Low vegetation	0.8486	0.9299	**0.9243**	0.7320	0.7829	**0.8589**
Trees	0.9159	**0.9488**	0.9473	**0.8846**	0.8568	0.8643
Cars	0.9922	**0.9969**	0.9959	0.9865	0.9879	**0.9912**
Clutter	0.9995	0.9993	**0.9996**	0.9518	**0.9598**	0.9522
Total	0.7919	**0.9003**	0.8962	0.4752	0.3974	**0.6463**

Table 4 SegNet – Classification performance by object type (accuracy only) for ISPRS in-sample (Vaihingen) and out-of-sample (Potsdam) validation using three training cases.

OBJECTS OF INTEREST	SEGNET LITE (ISPRS)					
	VAIHINGEN			POTSDAM (9 CM)		
	NDSM	IRRG	NDSM & IRRG	NDSM	IRRG	NDSM & IRRG
Impervious surfaces	0.8706	0.9519	**0.9559**	0.7123	**0.7950**	0.7827
Buildings	0.9539	0.9726	**0.9735**	**0.8559**	0.5554	0.6016
Low vegetation	0.8417	**0.9322**	0.9276	0.6077	0.7651	**0.8182**
Trees	0.9162	0.9490	**0.9486**	**0.8687**	0.8384	0.8669
Cars	0.9922	**0.9969**	0.9959	0.9864	**0.9887**	0.9871
Clutter	0.9992	0.9992	**0.9996**	0.9522	**0.9551**	0.9495
Total	0.7869	**0.9009**	0.9006	0.4916	0.4488	**0.5030**

Table 5 SegNet Lite – Classification performance by object (accuracy only) for ISPRS in-sample (Vaihingen) and out-of-sample (Potsdam) validation using three training cases.

Figure 7 shows qualitative results for the SegNet architecture when generating predictions using the three training cases. Ground truth is annotated using color-coding for roads (white), buildings (blue), low vegetation (cyan), trees (green), cars (yellow) and clutter (red). Again, without nDSM information, misclassifications occur between buildings, low vegetation, trees and roads. *Figure 8* presents the corresponding qualitative results for the SegNet Lite architecture.

OBJECTS OF INTEREST	5 × 5 SVM CLASSIFIER (ISPRS)					
	VAIHINGEN			POTSDAM (9 CM)		
	NDSM	IRRG	NDSM & IRRG	NDSM	IRRG	NDSM & IRRG
Impervious surfaces	0.7812	0.8733	**0.9320**	0.6847	0.7665	**0.8352**
Buildings	0.7931	0.8914	**0.9567**	**0.7550**	0.5257	0.6913
Low vegetation	0.8309	0.8715	**0.8978**	0.7246	0.7768	**0.8147**
Trees	0.7537	0.9101	**0.9317**	0.7464	**0.8325**	0.8214
Cars	0.9688	0.9915	**0.9928**	0.8530	**0.9862**	0.9832
Clutter	0.9922	**0.9997**	**0.9997**	0.9412	**0.9436**	0.9429
Total	0.5600	0.7687	**0.8553**	0.3266	0.4157	**0.5444**

Table 6 SVM – Classification performance by object (accuracy only) for in-sample (Vaihingen) and out-of-sample (Potsdam) validation using three training cases.
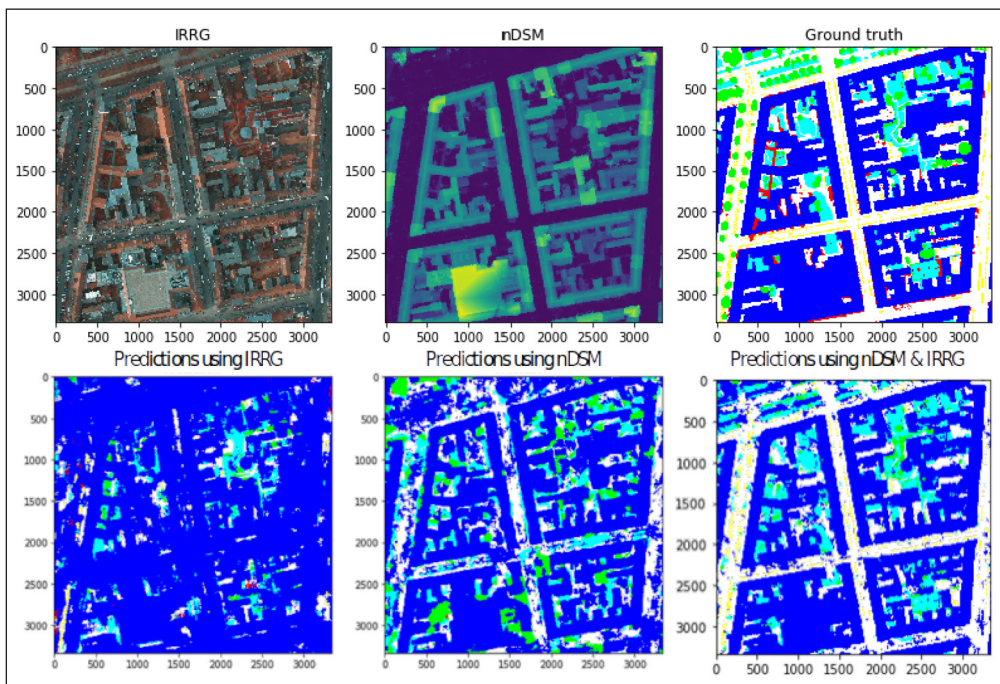
Figure 7 Qualitative out-of-sample classification performance for SegNet classifier applied to ISPRS Potsdam data. From left to right, the top row shows IRRG imagery, nDSM information, color-coded ground truth annotations. From left to right, bottom row display predictions when trained with (i) IRRG info only, (ii) nDSM info only, and (iii) combined IIRG & nDSM info.
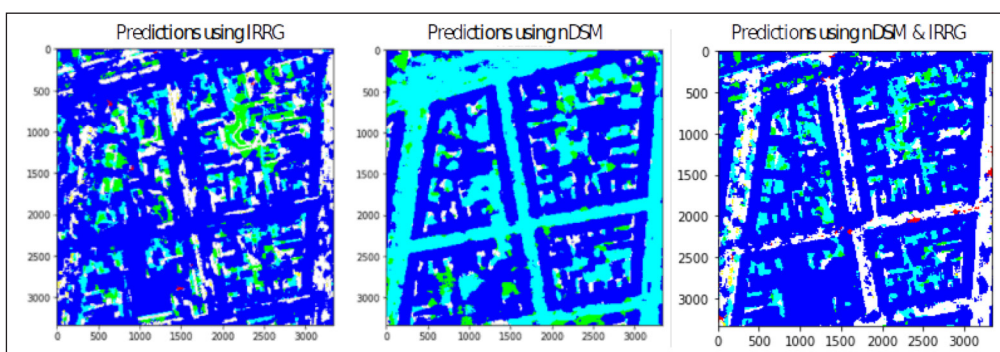

Figure 8 Qualitative out-of-sample classification performance for SegNet Lite classifier for the same tile as used in Figure 7 from the ISPRS Potsdam data, display predictions when trained with (i) IRRG info only, (ii) nDSM info only, and (iii) combined IIRG & nDSM info.

3.1.2 USSOCOM Dataset Results

Figures 9 and *10* summarize the resulting performance for our SegNet and SVM models using quantitative binary classification metrics such as accuracy, precision, recall, F1-score, and false-negative and false-positive rates. Classifiers are color-coded as follows: nDSM & RGB in blue, RGB-only in green, and nDSM-only in yellow.

For SVM (*Figure 10*), the left three bars show in-sample performance, i.e. testing was performed on the same 144 tiles that the classifiers were trained on, while the right three bars represent out-of-sample performance, i.e. testing was performed using 144 unseen Tampa tiles. For SegNet (*Figure 9*), the left three bars show out-of-sample performance for procedure 1 (testing
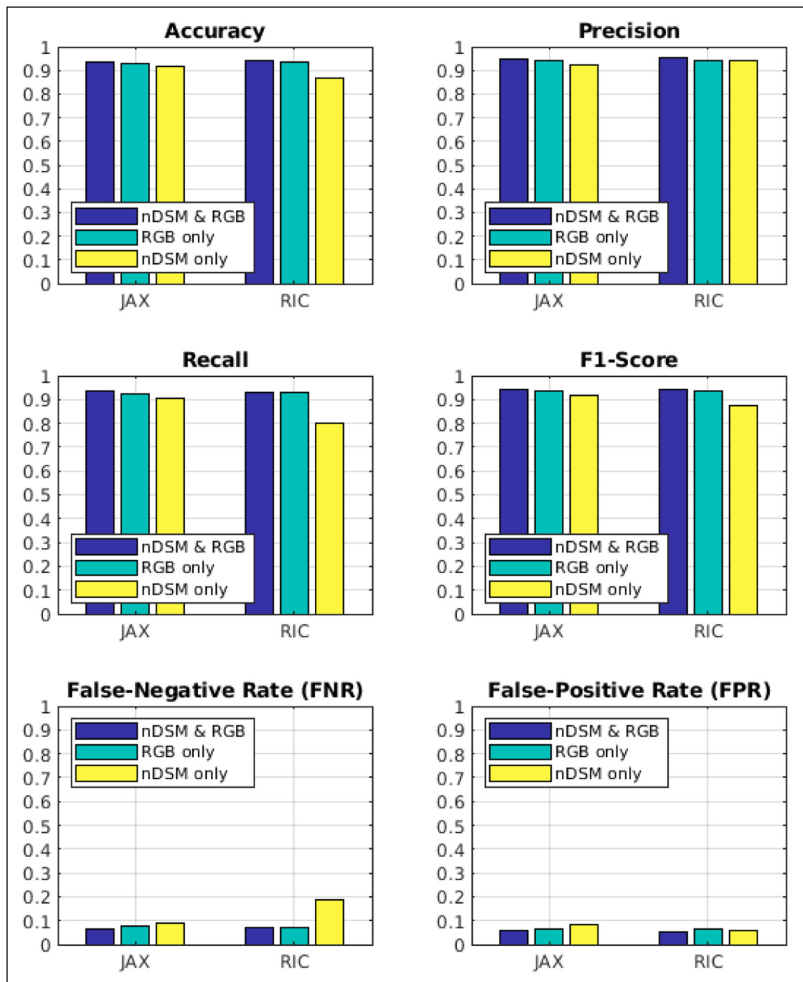
Figure 9 Cross-city building classification performance for the USSOCOM dataset using SegNet classifiers. Classifiers are color-coded: nDSM & RGB in blue, RGB-only in green, and nDSM-only in yellow. Note that JAX corresponds to out-of-sample testing with tiles from Jacksonville, and RIC corresponds to out-of-sample testing with tiles from Richmond.
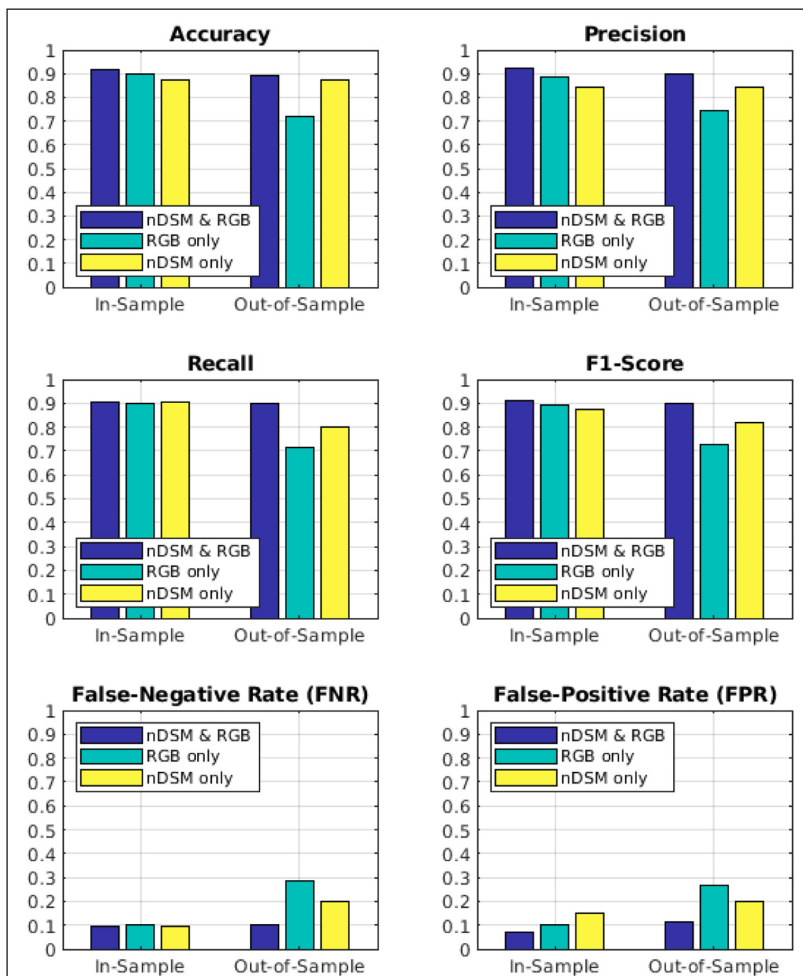

Figure 10 In-sample and out-of-sample building classification performance for the USSOCOM dataset using SVM classifiers. Classifiers are color-coded: nDSM & RGB in blue, RGB-only in green, and nDSM-only in yellow.

on 144 tiles over Jacksonville), while the right three bars represent out-of-sample performance for procedure 2 (testing on 144 tiles over Richmond).

Figure 10 indicates that in-sample performance (left most three bars in all six subplots) decreases only slightly when using RGB (green) or nDSM (yellow) information only, as compared to the combined nDSM & RGB classifier (blue). Note the RGB classifier slightly outperforms the (nDSM) classifier in in-sample performance. However, *Figures 9* and *10* indicate that performance differs significantly when testing the trained classifiers on a previously unseen test dataset, here USSOCOM Jacksonville or Richmond tiles (SegNet) and USSOCOM Tampa tiles (SVM). In addition to the overall performance discrepancies between the three classifiers for both SegNet and SVM, it becomes evident that accuracy drops only 10% when using only nDSM data as compared to 15% when using only RGB information (SVM; see upper left plot in *Figure 10*). For the SegNet classifiers, we observe that classifiers trained with RGB & nDSM information exhibit an average 0.6% higher out-of-sample accuracy than classifiers trained on RGB information alone. These results support the hypothesis that the nDSM information facilitates greater classifier generalizability, as compared to RGB information alone for building classification tasks.

Figure 11 presents the qualitative out-of-sample performance for all three SVM classifiers. From left to right, the upper row shows the training data RGB imagery, nDSM information, and ground truth, i.e. annotated building footprints, for Tampa tile #014. From left to right, the lower row shows predicted building footprints when training on (i) RGB information only, (ii) nDSM information only, and (iii) combined RGB & nDSM information. It is clear that the RGB & nDSM classifier on the lower right provides the best correlation with the actual ground truth (upper right). However, specific misclassifications occur for the other two cases. For example, when using nDSM information only, taller non-building objects such as trees are associated with a higher misclassification rate. In contrast, when using RGB information only, objects such as roads are often misclassified as buildings. However, when combining RGB and nDSM information, the number of misclassifications (both Type I and Type II) is significantly reduced.
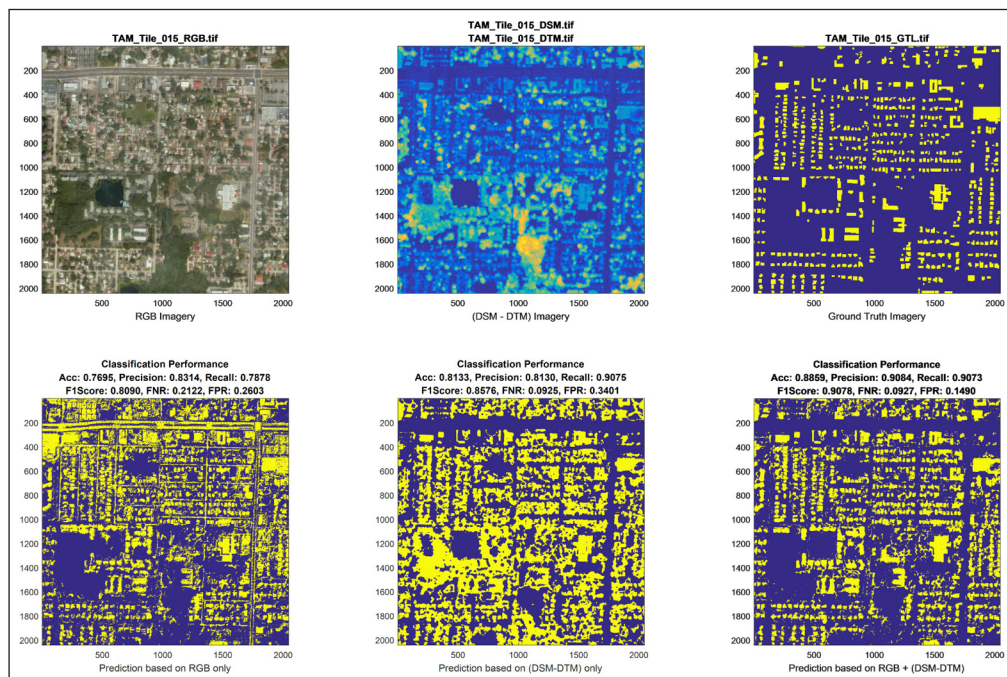
Figure 11 Qualitative out-of-sample classification performance for SVM classifiers applied to USSOCOM data. From left to right, the upper row shows RGB imagery, nDSM (DSM-DTM) information, and ground truth, i.e. annotated building footprints, for Tampa tile #014. From left to right, the lower row shows predicted building footprints when training on (i) nDSM information only, (ii) RGB imagery only, and (iii) combined RGB & nDSM information.

Table 7 captures our results from applying SegNet to the USSOCOM dataset with the training procedures specified in *Table 2*. Similarly, *Table 8* captures our results from applying our SVM classifier to the same dataset.

3.2. VALIDATION USING SMALL SAMPLE PROPORTION

In this section, the importance of 3D surface information is further tested using classification scenarios with scarce training samples. Sufficient data with adequate representation and viewing angles for all objects of interest may not always be assumed, particularly for remote

CLASSIFICATION METRICS	SEGNET (US SOCOM)					
	TRAIN TAM TEST JAX			TRAIN TAM TEST RIC		
	NDSM	RGB	NDSM & RGB	NDSM	RGB	NDSM & RGB
Accuracy	0.9164	0.9298	**0.9367**	0.8690	0.9339	**0.9386**
Precision	0.9245	0.9412	**0.9451**	0.9425	0.9416	**0.9512**
Recall	0.9105	0.9245	**0.9341**	0.8122	**0.9307**	0.9298
F1 Score	0.9175	0.9328	**0.9396**	0.8725	0.9361	**0.9404**
False Negative Rate	0.0895	0.0755	**0.0659**	0.1878	**0.0693**	0.0702
False Positive Rate	0.0829	0.0643	**0.0604**	0.0610	0.0626	**0.0518**

Table 7 SegNet – Balanced building classification performance metrics for cross-city (out-of-sample) validation following procedures 1 and 2 in table 2. In procedure 1 (left three columns), SegNet was trained on tiles from Tampa, Florida, and tested on tiles from Jacksonville, Florida. In procedure 2 (right three columns), SegNet was trained on tiles from Tampa, Florida, and tested on tiles from Richmond, Virginia.

CLASSIFICATION METRICS	5 × 5 SVM CLASSIFIER (USSOCOM)					
	IN-SAMPLE TESTING			OUT-OF-SAMPLE TESTING		
	NDSM	RGB	NDSM & RGB	NDSM	RGB	NDSM & RGB
Accuracy	0.8763	0.8978	**0.9178**	0.8763	0.7212	**0.8931**
Precision	0.8438	0.8850	**0.9214**	0.8438	0.7467	**0.9003**
Recall	**0.9047**	0.8996	0.9023	0.8000	0.7126	**0.8963**
F1 Score	0.8732	0.8922	**0.9117**	0.8200	0.7292	**0.8983**
False Negative Rate	**0.0953**	0.1004	0.0977	0.2000	0.2874	**0.1037**
False Positive Rate	0.1488	0.1039	**0.0684**	0.2000	0.2693	**0.1105**

Table 8 SVM – Balanced building classification performance metrics for in-sample and out-of-sample testing on the USSOCOM dataset.

sensing applications. Therefore, we train and test the two classification architectures from Section 2.3, while successively decreasing the number of training samples.
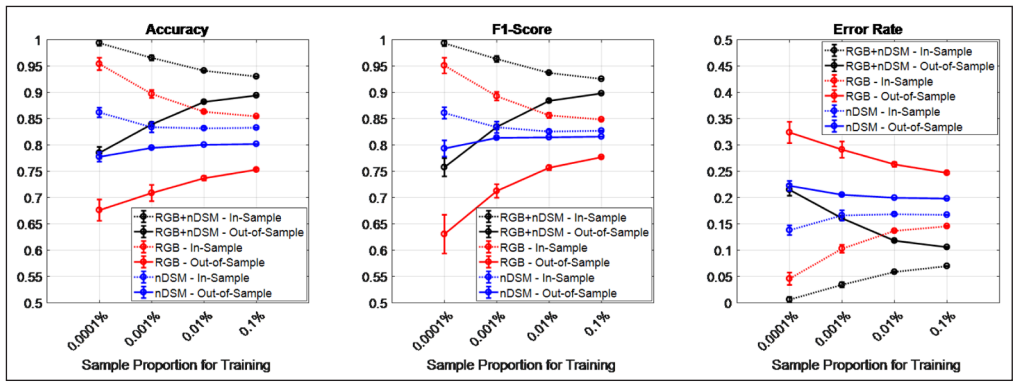
SVM classification in Section 3.1 was carried out using all 144 Jacksonville tiles from the ISPRS dataset. The 600 million training samples, i.e. annotated pixels, were randomly down-selected by a factor of 1,000 to 600,000 training samples, which corresponds to a sample proportion for training of 0.1%. For the following analysis, we further decrease the sample proportion to 0.01%, 0.001% and 0.0001%, thereby reducing the total number of training samples to 60,000, 6,000, and 600, respectively.

Table 9 presents the resulting average training times for all three SVM classifiers, highlighting the orders of magnitude between the different test cases. Clearly, the underlying numerical optimization can be completed significantly faster if fewer training samples need to be classified.

SAMPLE PROPORTION FOR TRAINING	0.0001%	0.001%	0.01%	0.1%
SVM Training Times (sec)	0.1	1.5	180	24,000

Table 9 Average training times (in seconds) for SVM classifiers when using smaller sample proportions for training on USSOCOM data.

Figure 12 displays the resulting in-sample and out-of-sample classification performance for the three SVM classifiers: RGB-only (red), nDSM-only (blue), and RGB & nDSM (black) as a function of sample proportion for training. Here, performance is measured in accuracy (left plot), F1-score (center plot) and error rate (right plot). All metrics assume class balancing. In-sample performance is plotted as dotted lines, while out-of-sample performance is plotted as solid lines. As training and test samples are selected randomly, five trials were conducted for each test case studied. In *Figure 12*, vertical bars are added to indicate the standard deviation for the particular performance metric over those five trials.

As discussed in the previous section, the RGB & nDSM classifier provides the best in-sample performance at 93% accuracy when using 0.1% of all training data. In-sample performance for the RGB-only and nDSM-only classifiers is 85% and 83%, respectively. In-sample accuracy (dotted lines) increases for all three classifiers as the sample proportion for training decreases. This is due to the fact that fewer training samples have to be classified. However, out-of-sample performance for all classifiers decreases with decreasing sample proportion for training, indicating that the resulting classifiers lose their generalizability when trained on smaller training sets due to overfitting. For out-of-sample performance, the nDSM-only classifier outperforms the RGB-only classifier, which further affirms the findings from Section 3.1. Interestingly, nDSM-only even outperforms the RGB & nDSM in the 0.0001% case. This result may relate to the curse of dimensionality (Keogh and Mueen, 2017), as the nDSM classifier operates in a reduced feature space of 25 dimensions (see Section 2.3.1), while the combined RGB & nDSM classifier operates in 100 dimensions. In general, if training data is scarce, a reduced feature space can improve generalizability by avoiding overfitting.

In addition to the SVM classifiers, we conduct the same validation analysis for small sample proportion for training the two SegNet architectures from Section 2.3.2. *Figure 13* displays the results when training the SegNet and SegNet Lite classifiers with 15%, 25%, 50% and 100% of the data. The method used for obtaining a subset of the data is to select a random point from each image and take a width and height equal to the desired fraction of the full image as the cropping region, which is then used for training. As before, training was carried out using 12 ISPRS Vaihingen tiles. Testing was then performed for three cases: (i) in-sample/in-city (using the 12 Vaihingen tiles that were used for training), (ii) out-of-sample/in-city (using the remaining 4 Vaihingen tiles not used from training), and (iii) out-of-sample/out-of-city (using all ISPRS Potsdam tiles). Out-of-sample/cross-city accuracy across the SegNet and Segnet Lite models with and without nDSM generally indicate a mild positive correlation between portion of data and accuracy, suggesting that 50% of the data for a given city might be sufficient for the full city classification.
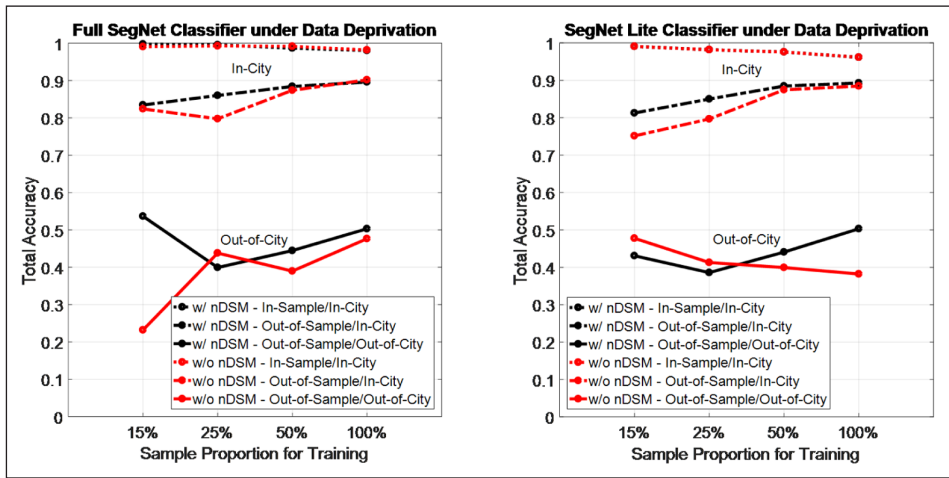
In-sample/in-city accuracy across the SegNet and Segnet Lite models with and without nDSM exhibits a negative correlation between portion of dataset and accuracy. As with the SVM classifier, this can be attributed to the network having less samples to classify, and therefore being able to overfit to the scarce training set. Lastly, the non-nDSM trained SegNet model has a negative correlation between accuracy and training proportion in regards to cross-city

testing. This correlation may indicate that, in addition to the cross-validation results of RGB and nDSM, RGB information alone can lead to overfitting and therefore hinder generalizability of a classification model to other cities.

4. CONCLUSION

This paper evaluated the importance of 3D surface information for remote sensing using several classification tasks. Two distinct classifiers, i.e. SVM and FCNN architectures, were introduced and assessed for performance when trained with and without nDSM information. Two publicly available datasets, i.e. the USSOCOM Urban 3D challenge (H. Goldberg, Brown, and S. Wang, 2017; H.R. Goldberg et al., 2018) and the ISPRS 2D Semantic Labeling contests (Gerke et al., 2014), were utilized for training and rigorous in-sample and out-of-sample performance assessment. In all cases, the study demonstrated that high in-sample classification performance can be maintained even when depriving the classifier of nDSM information. However, out-of-sample performance, i.e. when testing the classifier on previously unseen data from a different city, drops significantly for both SVM and FCNN classifiers trained without nDSM information. We conclude that nDSM information is vital for accurately generalizing classification methods to datasets not included in training.

An additional study revealed that nDSM information is also critical when training a classifier with relatively few training samples. Again, in-sample performance remains high with and without nDSM information, but generalizability decreases substantially when nDSM information is excluded from training.

Together, these validation experiments demonstrate the importance of including nDSM information to ensure generalizable out-of-sample predictive performance for remote sensing classification tasks.

DATA ACCESSIBILITY STATEMENT

Please find the pre-print version of the article here: *https://arxiv.org/abs/2104.13969*.

ACKNOWLEDGEMENTS

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Jan Petrich ⓘ *orcid.org/0000-0003-4457-1537*
Pennsylvania State University Applied Research Laboratory, United States

Ryan Sander ⓘ *orcid.org/0000-0002-2747-253X*
National Geospatial-Intelligence Agency, United States; Massachusetts Institute of Technology, United States

Eliza Bradley ⓘ *orcid.org/0000-0002-8071-1560*
Pennsylvania State University Applied Research Laboratory, United States

Adam Dawood
Pennsylvania State University Applied Research Laboratory, United States

Shawn Hough
Pennsylvania State University Applied Research Laboratory, United States

REFERENCES

Audebert, N, Le Saux, B and **Lefèvre, S.** 2018. Beyond rgb: very high resolution urban remote sensing with multimodal deep networks. *ISPRS journal of photogrammetry and remote sensing* [Online], 140: 20–32. DOI: *https://doi.org/10.1016/j.isprsjprs.2017.11.011*

Aurelio, YS, de Almeida, GM, de Castro, CL and **Braga, AP.** 2019. Learning from imbalanced data sets with weighted cross-entropy function. *Neural processing letters* [Online], 50(2): 1937–1949. DOI: *https://doi.org/10.1007/s11063-018-09977-1*

Badrinarayanan, V, Kendall, A and **Cipolla, R.** 2017. Segnet: a deep convolutional encoderdecoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* [Online], 39(12): 2481–2495. DOI: *https://doi.org/10.1109/TPAMI.2016.2644615*

Ball, JE, Anderson, DT and **Chan, CS.** 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of applied remote sensing* [Online], 11(4): 042609. DOI: *https://doi.org/10.1117/1.JRS.11.042609*

Blaha, M, Vogel, C, Richard, A, Wegner, JD, Pock, T and **Schindler, K.** 2016. Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling. *Proceedings of the IEEE conference on computer vision and pattern recognition* [Online], pp. 3176–3184. DOI: *https://doi.org/10.1109/CVPR.2016.346*

Chen, K, Fu, K, Sun, X, Weinmann, M, Hinz, S, Jutzi, B and **Weinmann, M.** 2018. Deep semantic segmentation of aerial imagery based on multi-modal data. *Igarss 2018–2018 IEEE international geoscience and remote sensing symposium* [Online], pp. 6219–6222. DOI: *https://doi.org/10.1109/IGARSS.2018.8519225*

Chen, K, Weinmann, M, Gao, X, Yan, M, Hinz, S, Jutzi, B and **Weinmann, M.** 2018. Residual shuffling convolutional neural networks for deep semantic image segmentation using multi-modal data. *ISPRS annals of photogrammetry, remote sensing & spatial information sciences* [Online], 4(2). DOI: *https://doi.org/10.5194/isprs-annals-IV-2-65-2018*

Cortes, C and **Vapnik, V.** 1995. Support-vector networks. *Machine learning* [Online], 20(3): 273–297. DOI: *https://doi.org/10.1023/A:1022627411411*

Gerke, M, Rottensteiner, F, Wegner, J and **Sohn, G.** 2014. ISPRS semantic labeling contest [Online]. DOI: *https://doi.org/10.13140/2.1.3570.9445*

Gevaert, C, Persello, C, Sliuzas, R and **Vosselman, G.** 2016. Classification of informal settlements through the integration of 2d and 3d features extracted from uav data. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences* [Online], 3: 317. DOI: *https://doi.org/10.5194/isprs-annals-III-3-317-2016*

Goldberg, H, Brown, M and **Wang, S.** 2017. A benchmark for building footprint classification using orthorectified rgb imagery and digital surface models from commercial satellites. *Proceedings of IEEE applied imagery pattern recognition workshop 2017* [Online]. DOI: *https://doi.org/10.1109/AIPR.2017.8457973*

Goldberg, HR, Wang, S, Christie, GA and **Brown, MZ.** 2018. Urban 3d challenge: building footprint detection using orthorectified imagery and digital surface models from commercial satellites. *Geospatial informatics, motion imagery, and network analytics viii* [Online]. Vol. 10645. International Society for Optics and Photonics, p. 1064503. DOI: *https://doi.org/10.1117/12.2304682*

Janocha, K and **Czarnecki, WM.** 2017. On loss functions for deep neural networks in classification. *Arxiv preprint arxiv:1702.05659* [Online]. DOI: *https://doi.org/10.4467/20838476SI.16.004.6185*

Keogh, E and **Mueen, A.** 2017. Curse of dimensionality. In: Sammut, C and Webb, GI (eds.), *Encyclopedia of machine learning and data mining* [Online]. Boston, MA: Springer US, pp. 314–315. DOI: *https://doi.org/10.1007/978-1-4899-7687-1_192*

Lai, K and **Fox, D.** n.d. 3d laser scan classification using web data and domain adaptation. [Online]. DOI: *https://doi.org/10.15607/RSS.2009.V.022*

Ma, L, Liu, Y, Zhang, X, Ye, Y, Yin, G and **Johnson, BA.** 2019. Deep learning in remote sensing applications: a meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing* [Online], 152: 166–177. DOI: *https://doi.org/10.1016/j.isprsjprs.2019.04.015*

MathWorks. 2018. *Fitcecoc – fit multiclass models for support vector machines or other classifiers. https://www.mathworks.com/help/stats/fitcecoc.html*.

Maturana, D and **Scherer, S.** 2015. 3d convolutional neural networks for landing zone detection from lidar. *2015 IEEE international conference on robotics and automation (icra)* [Online]. *IEEE*, pp. 3471–3478. DOI: *https://doi.org/10.1109/ICRA.2015.7139679*

Park, Y and **Guldmann, J-M.** 2019. Creating 3d city models with building footprints and lidar point cloud classification: a machine learning approach. *Computers, environment and urban systems* [Online], 75: 76–89. DOI: *https://doi.org/10.1016/j.compenvurbsys.2019.01.004*

Paszke, A, Gross, S, Massa, F, Lerer, A, Bradbury, J, Chanan, G, Killeen, T, Lin, Z, Gimelshein, N, Antiga, L, Desmaison, A, Kopf, A, Yang, E, DeVito, Z, Raison, M, Tejani, A, Chilamkurthy, S, Steiner, B, Fang, L, Bai, J and **Chintala, S.** 2019. Pytorch: an imperative style, high-performance deep learning library.

In: Wallach, H, Larochelle, H, Beygelzimer, A, d'Alché-Buc, F, Fox, E and Garnett, R (eds.), *Advances in neural information processing systems 32* [Online]. Curran Associates, Inc., pp. 8024–8035. Available from: *http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf*.

Ronneberger, O, Fischer, P and **Brox, T.** 2015. U-net: convolutional networks for biomedical image segmentation. *International conference on medical image computing and computer-assisted intervention* [Online]. Springer, pp. 234–241. DOI: *https://doi.org/10.1007/978-3-319-24574-4_28*

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *Corr* [Online], abs/1609.04747. arXiv: 1609.04747. Available from: *http://arxiv.org/abs/1609.04747*.

Tucker, CJ, Slayback, DA, Pinzon, JE, Los, SO, Myneni, RB and **Taylor, MG.** 2001. Higher northern latitude normalized difference vegetation index and growing season trends from 1982 to 1999. *International journal of biometeorology* [Online], 45(4): 184–190. DOI: *https://doi.org/10.1007/s00484-001-0109-8*

Wang, D, Zhang, Y and **Zhao, Y.** 2007. Support vector machines for photometric redshift estimation from broadband photometry. *Data science journal* [Online], 6: S474–S480. DOI: *https://doi.org/10.2481/dsj.6.S474*

Weinmann, M. n.d. *Reconstruction and analysis of 3d scenes* [Online]. Springer. DOI: *https://doi.org/10.1007/978-3-319-29246-5*

Zhao, R, Pang, M and **Wang, J.** 2018. Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network. *International journal of geographical information science* [Online], 32(5): 960–979. DOI: *https://doi.org/10.3390/rs11131540*

Zhu, XX, Tuia, D, Mou, L, Xia, G-S, Zhang, L, Xu, F and **Fraundorfer, F.** 2017. Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE geoscience and remote sensing magazine* [Online], 5(4): 8–36. DOI: *https://doi.org/10.1109/MGRS.2017.2762307*