A Semi-automated Probabilistic Framework for Tree Cover Delineation from 1-m NAIP Imagery Using a High Performance Computing Architecture

Saikat Basu, *Student Member, IEEE,* Sangram Ganguly, *Member, IEEE,* Ramakrishna R. Nemani, Supratik Mukhopadhyay, *Member, IEEE,* Gong Zhang, Cristina Milesi, Andrew Michaelis, Petr Votava, Ralph Dubayah, Laura Duncanson, Bruce Cook, Yifan Yu, Sassan Saatchi, Robert DiBiano, Manohar Karki, Edward Boyda, Uttam Kumar, *Member, IEEE* and Shuang Li

***Abstract*—Accurate tree cover estimates are useful to derive Above Ground Biomass (AGB) density estimates from Very High Resolution (VHR) satellite imagery data. Numerous algorithms have been designed to perform tree cover delineation in high to coarse resolution satellite imagery, but most of them do not scale to terabytes of data, typical in these VHR datasets. In this paper, we present an automated probabilistic framework for the segmentation and classification of 1-m VHR data as obtained from the National Agriculture Imagery Program (NAIP) for deriving tree cover estimates for the whole of Continental United States, using a High Performance Computing Architecture. The results from the classification and segmentation algorithms are then consolidated into a structured prediction framework using a discriminative undirected probabilistic graphical model based on Conditional Random Field (CRF), which helps in capturing the higher order contextual dependencies between neighboring pixels. Once the final probability maps are generated, the framework is updated and re-trained by incorporating expert knowledge through the relabeling of misclassified image patches. This leads to a significant improvement in the true positive rates and reduction in false positive rates. The tree cover maps were generated for the state of California, which covers a total of 11,095 NAIP tiles and spans a total geographical area of 163,696 sq. miles. Our framework produced correct detection rates of around 88% for fragmented forests and 74% for urban tree cover areas, with false positive rates lower than 2% for both regions. Comparative studies with the National Land Cover Data (NLCD) algorithm and the LiDAR high-resolution canopy height model showed the effectiveness of our algorithm for generating accurate high-resolution tree-cover maps.**

***Index Terms*—NAIP, Aerial Imagery, Machine Learning, Neural Network, Statistical Region Merging, Conditional Random Field, High Performance Computing.**

S. Basu, S. Mukhopadhyay, M. Karki and R. DiBiano are with the Department of Computer Science, Louisiana State University, Baton Rouge, LA, 70803, USA, e-mail: (sbasu8@lsu.edu).

S. Ganguly and G. Zhang are with the Bay Area Environmental Research Institute (BAERI)/NASA Ames Research Center, Moffett Field, CA 94035, USA.

R. Nemani is with the NASA Advanced Supercomputing Division, NASA Ames Research Center, Moffett Field, CA 94035, USA.

C. Milesi is with the Biospheric Science Branch, NASA Ames Research Center, Moffett Field, CA 94035, USA.

A. Michaelis, P. Votava and S. Li are with the NASA Ames Research Center & University Corporation Monterey, Moffett Field, CA 94035, USA.

R. Dubayah and L. Duncanson are with the Department of Geographical Sciences, University of Maryland, College Park, MD 20742, USA.

B. Cook is with the NASA Goddard Space Flight Center, Biospheric Sciences Laboratory, Greenbelt, MD 20771, USA.

Y. Yu is with the Department of Atmospheric and Ocean Sciences, University of California at Los Angeles, CA 90095, USA.

S. Saatchi is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA.

E. Boyda is with the Department of Physics and Astronomy, Saint Mary's College of California, Moraga, CA 94575, USA.

U. Kumar is with the Oak Ridge Associated Universities/NASA Ames Research Center, Moffett Field, CA 94035, USA.

I. Introduction

An unresolved issue with medium-to-coarse resolution satellite-derived forest cover maps is their inaccuracy, particularly over heterogeneous landscapes, and the high degree of uncertainty they introduce when they are used for forest carbon mapping applications. Previous efforts have acknowledged the issues pertaining to misclassification errors in coarser resolution satellite-derived land cover products, however, limited studies are in place that demonstrate how very high resolution (VHR) land cover products at 1-m spatial resolution could improve regional estimations of Above Ground Biomass (AGB). This paper develops techniques and algorithms designed to improve the accuracy of current satellite-based AGB maps as well as provide a reference layer for more accurately estimating regional AGB densities from the Forest Inventory and Analysis (FIA). The VHR tree-cover map can be used to compute tree-cover estimates at any medium-to-coarse resolution spatial grid, reducing the uncertainties in estimating AGB density and mitigating the present shortcomings of medium-to-coarse resolution land-cover maps.

The principal challenges in computing VHR estimates of tree cover at 1-m are associated with (a) the high variability in land cover types as recognizable from satellite imagery, (b) data quality affected by conditions during acquisition and pre-processing, and (c) corruption of data due to atmospheric contamination and associated filtering techniques. Land cover class identification is difficult even through visual interpretation owing to high variance in atmospheric and lighting conditions, and manual delineation of tree cover from millions of imagery acquisitions is neither feasible nor cost-effective. Tree cover delineation can be mapped to an object recognition problem ([1], [2], [3], [4], [5]), which can be framed in two ways: a boundary delineation problem that can be solved by perceptual grouping or a bounding box extraction problem that is addressed using a classification framework that performs a binary/multi-class classification on the bounding

box. Perceptual grouping employs a segmentation module that clusters contextually related objects/object parts into a single unified region ([6], [7], [8], [9]). On the other hand, a classification framework uses a variety of learning algorithms, such as boosting ([10], [11]), random forests ([12], [13], [14]), Support Vector Machines [15] and various others for performing both supervised and unsupervised classification of image patches based on visual and spectral characteristics. Our work combines both of these approaches into a unified framework that performs a classification for individual pixels using feature descriptors extracted from a neighborhood (defined on a window centered at the pixel of interest) and then performs a perceptual grouping of pixels sharing similar visual and spectral signatures.

Present classification algorithms used for Moderate-resolution Imaging Spectroradiometer (MO-DIS) [16] or Landsat-based land cover maps, such as National Land Cover Data (NLCD) [17], produce accuracies of 75% and 78%, respectively. The MODIS algorithm works on 500-m resolution imagery; the NLCD works at 30-m resolution. The accuracy assessment is performed on a per-pixel basis and the relatively lower resolution of the dataset makes it difficult to analyze the performance of these algorithms for 1-m imagery. Thus, there is a pressing need for creating high resolution forest cover maps at a resolution of 1 m to improve accuracy in land cover maps and to improve several prognostic and diagnostic models that require land cover maps as input. An automated approach for tree crown delineation was proposed in [18], based on the identification of tree apexes and the maximum rate of change in spectral reflectance along transect extending outward from the tree center. The algorithm was applied to sub-meter resolution imagery (at most up to 30 cm) but its accuracy decreased consistently and non-linearly with increasing pixel spacing or decreasing sampling resolution. Other approaches for tree crown delineation based on the distribution of pixel brightness are proposed in [19] and [20]. [19] proposed evaluating the brightness distribution within the radius of a circle centered on each tree, with values near the center of crown being larger than at the periphery showing a test for a 150m by 150m IKONOS image. [20] applies a similar concept with the valley forming approach of [21], which treats variation in brightness in the imagery as topography, where bright pixels are peaks (the crowns) interspersed by valleys (the darker inter-tree spaces). Also here results are reported for a small test-area of 620×550 meter and hence it is unknown how the algorithm would perform on a larger test area with higher variability. Other novel classification algorithms based on Deep Neural Networks have been used in ([22], [23]). The framework in [22] is used for the recognition of roads in aerial images. Detecting trees is a much harder problem considering the significantly higher variability in tree-cover – trees can have various color and texture characteristics while roads have little variation in color or texture and belong to a fixed set of classes, such as concrete, mud, gravel, sand, etc. Another important feature in road detection is the incorporation of contextual information that improves accuracy of the classifier. On the other hand, a tree can be present beside another tree, a road, a building or even a water body. Thus, incorporating inter-class contextual information into our framework does not lead to significant improvements of the classification. [22] use a 64x64 detection window, which is a very large context for a tree-delineation problem in which an image patch might contain multiple classes, such as bare ground, roads, rooftops etc. and hence not suitable for the tree-classification problem. A method based on object detection using a Bayes framework and a subsequent clustering of the objects into a hierarchical model using Latent Dirichlet Allocation was proposed by [23], but accurate delineation of tree-cover areas demands the use of a different approach because of the need for higher accuracy and lack of useful contextual information (for e.g., detecting a parking lot can use the presence of multiple cars and their orientation as a useful feature, but, a tree-delineation problem lacks the presence of such contextual information encoded in neighboring objects of interest). Classification and/or Segmentation of 1-m or sub-meter resolution imagery is possible with commercial packages (ENVI, PCI Geomatica, etc.), but these tools are not scalable across millions of scenes in an automated manner. The algorithm proposed by [24] is similar to our approach, which uses a segmentation module and a Random Forest based classification module to assess tree cover in the National Agriculture Imagery Program (NAIP) data [25]. The algorithm demonstrates a viable operational tool for the classification of 1-m NAIP imagery and produces an overall accuracy of 84.8%. However, the analysis is based on the software Definiens Developer Professional [26], which affects the scalability and cost-effectiveness of the implementation to terabytes of data. Additionally, the authors limited the testing of the methodology to Pembina County in North Dakota, which covers an area of only 1,122 sq. miles as opposed to the 163,696 sq. miles in our implementation.

In this paper, we present an automated probabilistic framework for the segmentation and classification of 1-m VHR NAIP data to derive accurate large-scale estimates of tree cover. The results from the classification and segmentation algorithms are consolidated using a discriminative undirected probabilistic graphical model that performs structured prediction and helps in capturing the higher order contextual dependencies between neighboring pixels. A detailed description of the dataset is given in Section II. A comprehensive summary of the proposed framework and the High Performance Computing (HPC) implementation details are provided in Section III. Section IV discusses the results and performance analysis for our pilot demonstration of the algorithm over California.

II. DATASET

The NAIP dataset consists of a total of 330,000 scenes spanning the whole of the Continental United States (CONUS). We used the uncompressed Digital Ortho Quarter Quad tiles (DOQQs) which are GeoTIFF images with an area corresponding to the United States Geological Survey (USGS) topographic quadrangles. The average image tiles are $\sim$6000 pixels in width and $\sim$7000 pixels in height, and are approximately 200 megabytes each. The entire NAIP dataset for the Continental Unites State is $\sim$65 terabytes. The imagery was acquired at a 1-m ground sample distance (GSD) with a horizontal accuracy

that lies within six meters of photo-identifiable ground control points [27]. The images consist of 4 bands – red, green, blue and Near Infrared (NIR). We performed the preliminary test of our algorithm and obtained tree-cover maps for the entire state of California, a total of 11,095 image tiles in the NAIP dataset. Figure 1 shows some sample image patches from the NAIP dataset containing tree and non-tree areas.

The tree cover maps generated by our algorithm were validated against two high-resolution airborne LiDAR data footprints. The first set of LiDAR data (henceforth referred to as Area 1) was collected over the Teakettle Experimental Forest in the western Sierra Nevada mountain range, California. The LiDAR was flown in the summer of 2008 with the University of Florida OPTECH GEMINI ALSM unit, operating at 100-125 kHz with a maximum 25° scanning angle. Data were flown 600-750 m above ground, with 50%-75% swath overlap yielding an average return density of approximately 18 pts/m^2. LiDAR processing was conducted at the University of Maryland following [28]. A Digital Elevation Model (DEM) was fit to the lowest returns from the raw LiDAR returns, and smoothed to represent local topography. The elevation of the corresponding DEM pixel was subtracted from each raw LiDAR return. The maximum LiDAR height in each pixel was used to produce a Canopy Height Model (CHM) at a resolution of 0.5 m. For the purpose of validation, the LiDAR data were resampled to 1 m spatial resolution.

The second set of LiDAR data (henceforth referred to as Area 2) was obtained in the Chester area in California, using the NASA Goddard's LiDAR, Hyperspectral and Thermal (G-LiHT) Airborne Imager [29]. NASA's Cessna 206 was used for acquiring the G-LiHT data. The Cessna was fitted with the VQ-480 (Riegl USA, Orlando, FL, USA) airborne laser scanning (ALS) instrument and was flown at an altitude of 335 m. The data acquired had a swath width of 387 m and a Field of View of 60°. The sampling density was 6 pulses/m^2. The spatial resolution of the final LiDAR data was 1 m.

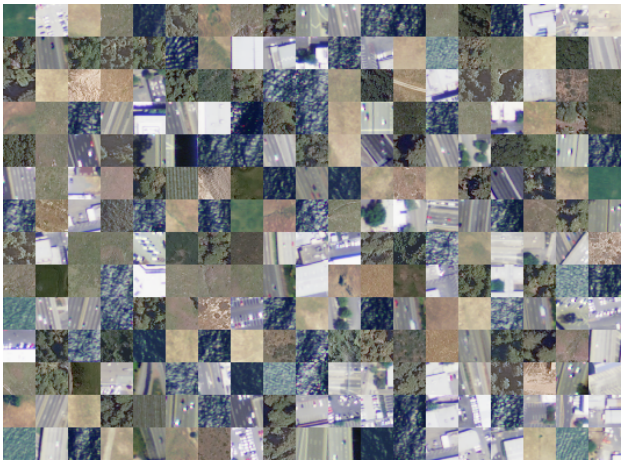


Fig. 1: A sample of image patches from the NAIP dataset showing tree and non-tree areas.

III. Methodology

We have designed and implemented a scalable semi-automated probabilistic framework for the classification and segmentation of millions of scenes using a HPC architecture. The framework is robust to account for variability in land cover data as well as atmospheric and lighting conditions. Our framework consists of the following modules: (1) Segmentation, (2) Feature Extraction, (3) Classification, and (4) Labeling.

A. Unsupervised Segmentation

A segment can be considered to be any region having pixels with uniform spectral characteristics. The aim of segmentation is to find regions with uniform values for the different spectral bands representing a particular land cover class. Segmentation is performed using the Statistical Region Merging (SRM) Algorithm [30]. We use a generalized SRM algorithm that incorporates values from all four bands. The SRM algorithm initially considers each pixel as a region and merges them to form larger regions based on a merging criterion. The merging criterion that we use in this case is as follows: Given the differences in red, green, blue and NIR values of neighboring pixels that correspond to dR, dG, dB and dNIR, respectively, merge two regions if (dR<threshold & dG<threshold & dB<threshold & dNIR<threshold). The merging criterion can be formalized as a merging predicate that is evaluated as true if two regions are merged and false otherwise. The generalized version of the merging predicate (adopted from [30]) can be formally written as follows:

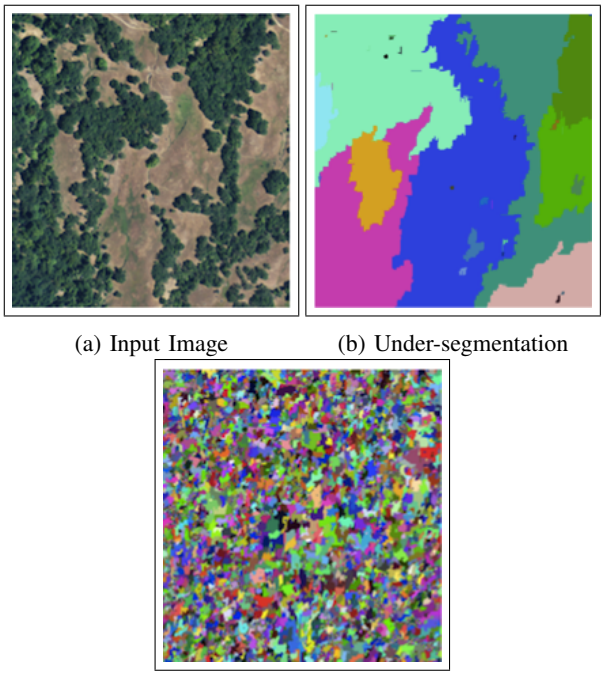
$$P(R,R') = \begin{cases} true, & \text{if } \forall c \in \{R,G,B,NIR\} \\ & |\bar{R}'_c - \bar{R}_c| \leq \sqrt{b^2(R) + b^2(R')} \\ false & \text{otherwise.} \end{cases} \tag{1}$$

where $\bar{R}_c$ and $\bar{R'}_c$ denote the mean value of the color channel c for regions R and R' respectively. b is a function defined as follows:

$$b(R) = g\sqrt{\frac{1}{2Q|R|} ln\left(\frac{|R_{|R|}|}{\delta}\right)} \tag{2}$$

where g is the number of possible values for each color channel (256 in our case). $|R|$ denotes the cardinality of a segment, i.e., the number of pixels within the boundaries of an image region R. $R_{|R|}$ represents the set of all regions that have the same cardinality as R. δ is a parameter that is inversely proportional to the image size. Q is the quantization parameter that controls the coarseness of the segmentation. A careful analysis of Equation 1 and Equation 2 shows that a higher value of Q results in a lower threshold thereby reducing the probability of two segments getting merged into a bigger segment, thus giving a finer segmentation. A lower value of Q results in a higher threshold and a coarser segmentation. The algorithm calculates the differences between neighboring pixels and sorts the pairs using radix sort. If the merging criterion is met, then it merges corresponding segments into one. We set a low threshold (or a higher Q value of 2^{15}) in

order to perform over-segmentation. Each class (e.g. forest, grass, etc.) might be divided into multiple segments, but one segment would ideally not contain more than one class. This is useful for eliminating the possibility of inter-class overlap within a segment. Figure 2 shows an under-segmented and an over-segmented image. As can be seen in the under-segmented version, the same segment may contain both vegetated and non-vegetated areas.



(a) Input Image (b) Under-segmentation

(c) Over-segmentation

Fig. 2: Example NAIP input image and under-segmented and over-segmented outputs from the Statistical Region Merging Algorithm. Under-segmentation creates interclass overlap within a segment while in over-segmentation, each segment ideally contains regions belonging to a single class and there is no inter-class overlap.

In the case of an over-segmented image, areas within large homogeneous patches of vegetated pixels are split into multiple segments in the presence of spectral variability induced by factors such as shadows cast by tree/non-tree regions or the presence of dry brown patches within grassy areas, improving overall classification accuracy. SRM is more efficient compared to other segmentation algorithms, such as k-means clustering [31]. The lists of merging tests can be sorted using radix sort with color difference as the keys and hence has a time complexity of $O(|I|\log(g))$ which is linear in $|I|$. Here, $|I|$ is the cardinality or size of the input image. SRM segments a 512×512 image in about one second on an Intel Pentium 4 2.4G processor and hence is well suited for the current application involving terabytes of data. However, SRM has high memory requirements, around 3 Gigabytes per 6000×7000 image. This is mitigated by splitting the input image into 256×256 windows. This architectural implementation is detailed in Section III-G.

B. Feature Extraction

Prior to the classification process, the feature extraction phase computes 150 features from the input imagery. The key features are mean, standard deviation, variance, 2nd moment, direct cosine transforms, correlation, co-variance, autocorrelation, energy, entropy, homogeneity, contrast, maximum probability and sum of variance of the hue, saturation, intensity, and NIR channels as well as those of the color co-occurrence matrices. These features were shown to be useful descriptors for classification of satellite imagery in previous studies ([32], [33], [34]). The Red band already provides a useful feature for delineating forests and non-forests based on chlorophyll reflectance, however, we also use derived features (vegetation indices derived from spectral band combinations) that are more representative of vegetation greenness, such as the Enhanced Vegetation Index (EVI) [35], Normalized Difference Vegetation Index (NDVI) ([36], [37]) and Atmospherically Resistant Vegetation Index (ARVI) [38].

These indices are expressed as :

$$EVI = G \times \frac{NIR - Red}{NIR + c_{red} \times Red - c_{blue} \times Blue + L} \quad (3)$$

Here, the coefficients G, c_{red}, c_{blue} and L are chosen to be 2.5, 6, 7.5 and 1, following those adopted in the MODIS EVI algorithm [27].

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (4)$$

$$ARVI = \frac{NIR - (2 \times Red - Blue)}{NIR + (2 \times Red + Blue)} \quad (5)$$

The performance of our machine learning-based approach depends to a large extent on the selected features. Some features contribute more than others towards optimal classification. The 150 features extracted are narrowed down to 22 using a feature-ranking algorithm based on Distribution Separability Criterion [39]. Some example image features are shown in Figure 3.

Sample Dataset	Dist. between Means	Standard Deviations
Raw Images	0.2163	0.1337
Extracted Features	**0.6712**	**0.0751**

TABLE I: Distance between Means and Standard Deviations for raw image values and the Extracted feature vectors for a sample set of 5000 randomly selected labeled image patches from the NAIP dataset for the state of California.

1) Feature Ranking: Improving classification accuracy can be viewed as maximizing the separability between the class-conditional distributions. Following the analysis presented in [39], we can view the problem of maximizing distribution separability as maximizing the distance between distribution means and minimizing their standard deviations. In order to quantify the statistical distribution properties of the NAIP dataset and to compare them to those of the extracted feature vectors, we randomly selected 5000 image patches from the NAIP tiles from the state of California and manually labeled

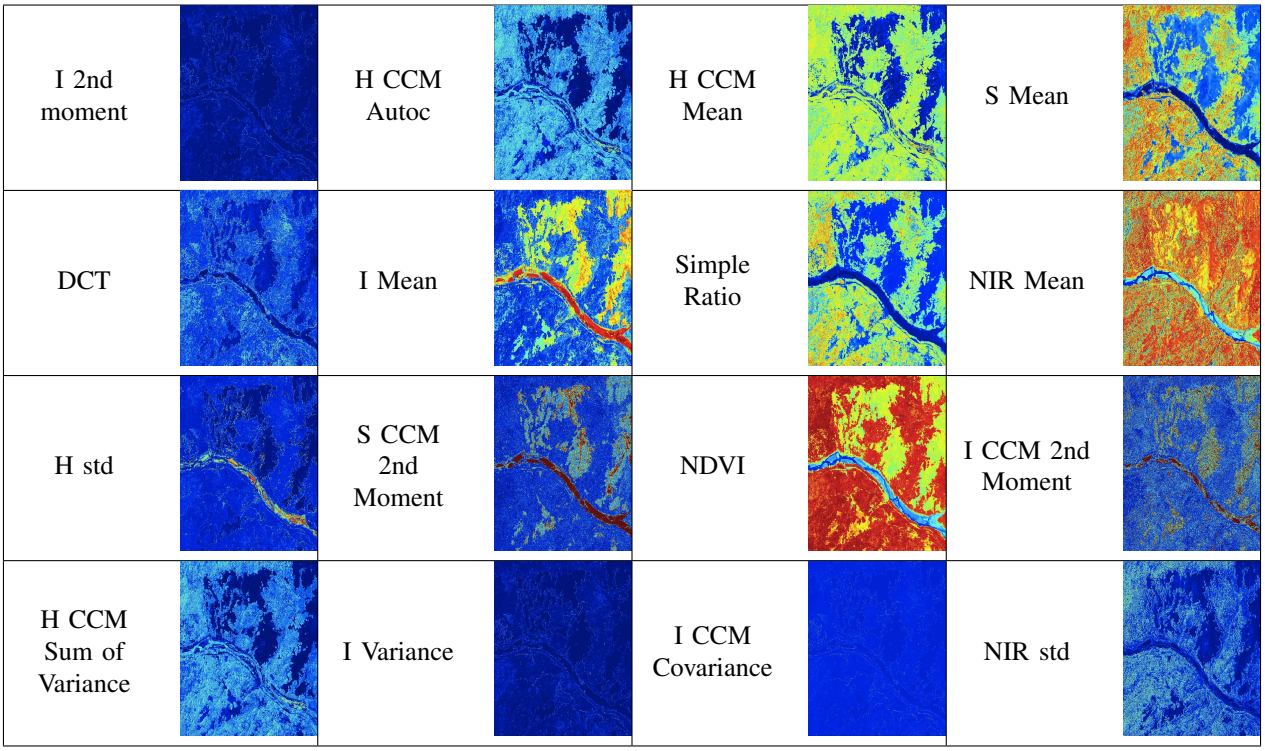


Fig. 3: Example Features extracted from a sample NAIP tile.

as tree/non-tree. The labeling was done in an unbiased way, i.e., $\sim 50\%$ of the samples are chosen from each class. Then we measured the distance between the means of the class conditional distributions and the standard deviations for both the raw pixel values as well as the features extracted in our framework. As illustrated in Table I, the extracted features have a higher distance between means and a lower standard deviation as compared to the original image distributions, thereby ensuring better class separability. We can derive a metric for the Distribution Separability Criterion as follows:

$$D_s = \frac{\overline{\|\delta_{mean}\|}}{\overline{\delta_\sigma}} \tag{6}$$

where $\overline{\|\delta_{mean}\|}$ indicates the mean of distance between means and $\overline{\delta_\sigma}$ indicates the mean of standard deviations of the class conditional distributions. Maximizing D_s over the feature space, a feature ranking can be obtained. Table II shows the ranking of the various features used in our framework along with the values of the corresponding distance between means $\overline{\|\delta_{mean}\|}$, standard deviation $\overline{\delta_\sigma}$ and Distribution Separability Criterion D_s.

Rank	Feature	$\overline{\|\delta_{mean}\|}$	$\overline{\delta_\sigma}$	D_s
1	I CCM mean	0.4031	0.1371	2.9403
2	H CCM sosvh	0.2359	0.0928	2.5413
3	H CCM autoc	0.2334	0.1090	2.1417
4	S CCM mean	0.0952	0.0675	1.4099
5	H CCM mean	0.0629	0.0560	1.1237
6	SR	0.0403	0.0428	0.9424
7	S CCM 2nd moment	0.0260	0.0312	0.8354
8	I CCM 2nd moment	0.0260	0.0312	0.8354
9	I 2nd moment	0.0260	0.0312	0.8345
10	I variance	0.0260	0.0312	0.8345
11	NIR std	0.0251	0.0315	0.7980
12	I std	0.0251	0.0314	0.7968
13	H std	0.0252	0.0317	0.7956
14	H mean	0.0240	0.0314	0.7632
15	I mean	0.0254	0.0336	0.7541
16	S mean	0.0232	0.0319	0.7268
17	I CCM covariance	0.0378	0.0522	0.7228
18	NIR mean	0.0246	0.0351	0.6997
19	ARVI	0.0229	0.0345	0.6622
20	NDVI	0.0215	0.0326	0.6594
21	DCT	0.0344	0.0594	0.5792
22	EVI	0.0144	0.0450	0.3207

TABLE II: Ranking of features based on Distribution Separability Criterion for the sample dataset.

C. Classification

Classification is performed for each image pixel using feature descriptors defined on its neighborhood. A neighborhood system for a pixel p is a set $\prod_p$ defined as

$$\prod_p = \bigcup_{r_L - p_L \leq \tau} r \tag{7}$$

Here, r_L and p_L are the locations i.e., the ordered tuple (x, y) for the pixels r and p respectively, where, x is the X-coordinate (along the horizontal axis) and y is the Y-coordinate (along the vertical axis).

$$r_L - p_L = \delta \text{ if } r_L \text{ lies on a } \delta \times \delta \text{ window centered at } p_L \tag{8}$$

The neighborhood system for the pixel p_L in shown in Figure 4.

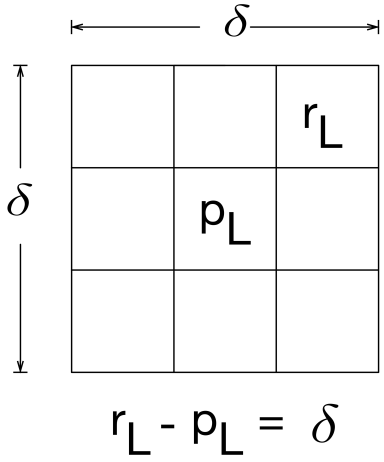


Fig. 4: The neighborhood system for the pixel p_L where $r_L - p_L = \delta$

τ is the parameter that controls the extent of the neighborhood. τ was chosen to be 4 by experiment and a Receiver Operating Characteristic (ROC) Curve analysis as detailed in the results section. The ROC Curve analysis was used to select the optimum value for the parameter τ that resulted in the highest True Positive Rates for the detection window.

Our classification module consists of a probabilistic Neural Network framework that generates the posterior probability maps of the tree-cover estimates in the imagery data. The proposed network takes the form of a fully connected feed-forward backpropagation Neural Network. In order to choose the optimal network architecture for the Neural Network classifier, we experimented with various network configurations along with the full set of 150 features as well as the set of 22 features selected in the feature selection stage highlighted in Section III-B. The results are reported in Table III. Interestingly, it can be observed from the table that the networks with 3 hidden layers produce lower classification accuracy than the networks with 2 hidden layers. This can be attributed to the limited amount of labeled training samples with respect to the increased model complexity of the deeper architectures (for instance, 22700 free parameters for the 100-100-100 neural network) and hence resulting in over-fitting. In order to prevent over-fitting of the deeper networks, we employ two techniques - 1) L_2-norm regularization [40] and 2) Dropout [41]. For L_2 regularization we used a weight decay penalty of 10^{-4} and for the Dropout, we used a dropout fraction of 0.5. The results of the validation error of the 100-100 and 100-100-100 neural networks with varying epochs of the learning algorithm are presented in Figure 18. It can be seen that the 100-100-100 network with L_2 norm regularization and Dropout perform on an equal scale and produce lower validation error than the non-regularized version. However, it is interesting to see that the smaller non-regularized network with 2 hidden layers and 100 neurons in each layer outperforms the network with 3 hidden layers and 100 neurons in each layer even with regularization. So, it can be concluded from the experiments that both Dropout and L_2 norm regularization can act equally well as regularization techniques for Deeper Neural Networks, however, for a limited number of training samples, the shallow network with 2 hidden layers still produces lower classification errors on the held-out validation set. Following the results from Table III, the best network was chosen as one with 2 hidden layers each having 50 neurons and one output layer having one neuron. The activation function is tansigmoid (tanhyperbolic) for hidden layers and linear for output layer:

$$\sigma(t) = \tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \tag{9}$$

The weights and biases are initialized using the Nguyen-Widrow Randomization algorithm [10]. We use the mean squared error (MSE) as the performance function. In the training phase around 100,000 training samples are chosen from each class. They are chosen randomly from a multitude of scenes ranging from densely forested areas to urban landscapes. An automated image labeling tool based on interactive segmentation developed as part of this study displays images randomly to a human expert who then labels the image patches as a tree cover or non-tree area, which are in turn saved to the training database along with proper labeling. We provide details of the image labeling tool in Section III-F.

The neural network gives an estimate of the posterior probabilities of the class labels, given the input vectors - the input vectors are the feature vectors extracted from the input image. As illustrated in [42], the outputs of a neural network trained by minimizing the mean squared error function approximates the conditional averages of the target data as

$$y_k(x) = \langle t_k|x\rangle = \int t_k p(t_k|x) dt_k \tag{10}$$

where t_k are the set of target values that represent the class membership of the input vector x_k and $p(t_k|x)$ is the probability that the input vector x attains the target value t_k. Thus, dt_k defines the differential over all target values t_k. To map the outputs of the neural network to the posterior probabilities of the labeling, we use a single output y and a target coding that sets $t^n = 1$ if x^n is from class C_1 and $t^n = 0$ if x^n is from class C_2. The target distribution can then be represented as

$$p(t_k|x) = \delta(t-1)P(C_1|x) + \delta(t)P(C_2|x) \tag{11}$$

Here, δ denotes the Dirac delta function which has the properties $\delta(x) = 0$ if $x \neq 0$ and

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \tag{12}$$

From (7) and (8), we get

$$y(x) = P(C_1|x) \tag{13}$$

The network output $y(x)$ represents the posterior probability of the input vector x having the class membership C_1 and the probability of the class membership C_2 is given by $P(C_2|x) = 1 - y(x)$.

D. Conditional Random Field

A Conditional Random Field (CRF) [43] has been used in the pattern recognition literature for performing structured prediction [44]. In structured prediction the labeling of a pixel depends not only on the feature values of that particular pixel but also on the values assumed by "neighboring" pixels. The word "neighboring" here can either mean a 4-connected or 8-connected neighborhood or some custom metric defining the notion of neighborhood. The concept of neighborhood is useful in encoding contextual information. The final labeling of a pixel as a vegetated pixel depends not only on whether that pixel is classified as a tree, but also on the classification of neighboring pixels. For example, if a pixel has been classified as a tree pixel by the classifier and all the neighboring pixels have been classified as non-tree pixels, then, it is safe to assume with a high probability that the result of the classifier is due to random classification noise. A Conditional Random Field (CRF) is a type of discriminative undirected probabilistic graphical model that encodes contextual information using an undirected graph [43]. The probability distributions are defined using a random variable X over a set of observations and another random variable Y over corresponding label sequences. Y is indexed by the vertices of an undirected graph $G = (V, E)$ such that $Y = (Y_v)_{v \in V}$. The tuple (X, Y) is known as a Conditional Random Field if the random variable Y conditioned on X exhibits the Markov property with respect to the graphical model, i.e., $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w{\sim}v)$, where, $w{\sim}v$ means that w and v are neighbors in G. Following the conventions defined in [45], the random variable X is defined over a lattice $V = 1, 2, ..., n$ and a neighborhood system N. It is to be noted that this neighborhood system N should not be confused with the neighborhood system $\prod_p$ defined in Equation 7. $\prod_p$ indicates the neighborhood for the classification algorithm that determines the bounds of the decision boundaries for the classifier outcome for a particular pixel p, whereas, N denotes the system characterized by uniform probability distributions owing to similar visual and spectral characteristics, which takes the form of a segment in this case. A CRF defines a set of random variables X_C conditionally dependent on each other as a clique c. A probability distribution associated with any random variable X_i of a clique is conditionally dependent on the distributions of all other random variables in the clique. The objective function we use takes the form

$$E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) + \sum_{c \in S} \psi_c(x_c) \quad (14)$$

where, $\psi_i(x_i)$ is the unary potential, $\psi_{ij}(x_i, x_j)$ is the pairwise potential and $\psi_c(x_c)$ is the function associated with higher order region consistency potential defined over a segment S.

The unary potential term is defined as

$$\psi_i(x_i) = \theta_N \psi_N(x_i) + \theta_{band} \psi_{band}(x_i) \quad (15)$$

where, $\theta_N \psi_N(x_i)$ denotes the potential due to the output produced by the neural network classifier described in Section III-C and $\theta_{band} \psi_{band}(x_i)$ is the potential from the band values from the NAIP images.

We can define the potential $\psi_N(x_i)$ derived from the classifier output as

$$\psi_N(x_i) = -\log P(C_i|x) = -\log y_i \quad (16)$$

Here, $P(C_i|x)$ denotes the normalized distribution generated by the classifier and y_i denotes the output distribution from the classifier.

Similarly, the pairwise term $\psi_{ij}(x_i, x_j)$ is updated to encode the band information as:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0, & \text{if } x_i = x_j \\ \theta_P + \theta_V exp(-\theta_\beta ||B_i - B_j||^2), & \text{otherwise.} \end{cases} \quad (17)$$

Here, B_i and B_j are the band vectors for pixels i and j respectively. The model parameters θ_N, θ_{band}, θ_P, θ_V and θ_β are learnt from the training data. The term $\psi_c(x_c)$ denotes the region consistency potential as defined in [45] and is given by:

$$\psi_c(x_c) = \begin{cases} 0, & \text{if } x_i = l_k \\ \theta_R |c|^{\theta_\alpha}, & \text{otherwise.} \end{cases} \quad (18)$$

Here, $|c|$ is the number of pixels in the segment and l_k denotes the label assigned to the segment c. $\theta_R |c|^{\theta_\alpha}$ denotes the cost associated with labelings that do not confirm with the labeling of the other pixels in the segment. This term ensures that the labels assigned to the pixels belonging to the same segment are consistent with one another, i.e., pixels belonging to the same segment are likely to belong to the same object/class. As illustrated in [45], this is useful in obtaining object segmentations with fine boundaries and particularly helpful for accurate delineation of tree cover areas in aerial images, where a single pixel denotes an area of $1m^2$. The CRF output with the unary, pairwise and the region consistency terms $\psi_i(x_i)$, $\psi_{ij}(x_i, x_j)$ and $\psi_c(x_c)$ are shown in Figure 5. It can be seen that the pairwise term improves the classification accuracy of the unary term by reducing the probability values associated with the false positives as evident from the fact that the probability values of most of the yellow (high probability) pixels appearing among the barren patch of land in figure (a) for unary potential are effectively reduced (denoted by blue/purple pixels) by the pairwise term in figure (b). Similarly, the region consistency term (or the segmentation term) improves upon the unary and pairwise terms by reducing the false positives further – most of the blue/purple pixels in figure (b) for the pairwise term are cleaned using the segmentation term/ the region consistency term in figure (c).

1) The CRF Learning Algorithm: The energy minimization in CRF is done using the α-expansion and $\alpha\beta$-swap algorithms [46]. In the α-expansion algorithm, for a given label α, an arbitrary set of pixels are assigned to this class label. For the $\alpha\beta$-swap algorithm, given a set of pixels with labeling α and another set of pixels with labeling β, the algorithm swaps the class labels for these set of pixels until the energy cannot be minimized any further. Details of the algorithms

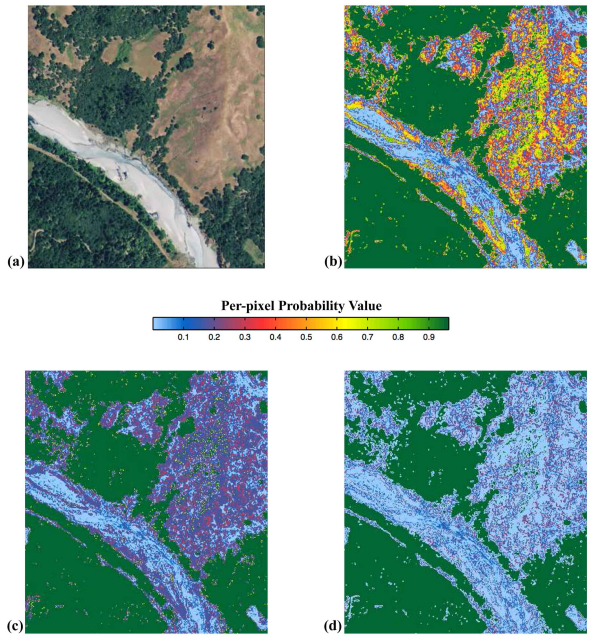


Fig. 5: (a) A sample NAIP tile and the CRF output probability maps with (b) the unary term $\psi_i(x_i)$, (c) the combination of the unary term $\psi_i(x_i)$ and the pairwise term $\psi_{ij}(x_i, x_j)$ and (d) the combination of the unary term $\psi_i(x_i)$, the pairwise term $\psi_{ij}(x_i, x_j)$ and the region consistency term $\psi_c(x_c)$.

are provided in Algorithm 1 and Algorithm 2. The key step in both algorithms is Step 5 where $\hat{y}$ is computed using graph cuts [46].

Algorithm 1 Alpha Expansion Algorithm

1: **procedure** ALPHAEXPANSION
2: Assign an arbitrary labeling y to the pixels of the image.
3: *done* $\leftarrow 0$
4: **for** `each label` $\alpha \in L$ **do**
5: find $\hat{y} \leftarrow$ argminE(y') among y' where y' lies within one α-expansion of y
6: **if** $E(\hat{y}) < E(y)$ **then**
7: y $\leftarrow \hat{y}$
8: *done* $\leftarrow 1$
9: **if** *done* = 1 **then**
10: **goto 3**.
11: **return** y.

2) Learning the Model Parameters: The optimal values of the model parameters were learnt by minimizing the cross validation error of the final pixel labeling assigned to the validation image set. Multiple rounds of cross-validation was used by selecting different subsets of the images for training and validation. The combined space of the parameter values θ_N, θ_{band}, θ_P, θ_V, θ_β and θ_α of the CRF is exponential in the number of individual parameter values and to determine the optimal values by an exhaustive search over the parameter

Algorithm 2 Alpha-Beta Swap Algorithm

1: **procedure** ALPHABETASWAP
2: Assign an arbitrary labeling y to the pixels of the image.
3: *done* $\leftarrow 0$
4: **for** `each pair of labels` $\alpha, \beta \in L$ **do**
5: find $\hat{y} \leftarrow$ argminE(y') among y' where y' lies within one $\alpha\beta$-swap of y
6: **if** $E(\hat{y}) < E(y)$ **then**
7: y $\leftarrow \hat{y}$
8: *done* $\leftarrow 1$
9: **if** *done* = 1 **then**
10: **goto 3**.
11: **return** y.

space is computationally intractable. A heuristic approximation technique was used by first optimizing the unary model parameters θ_N and θ_{band} followed by the parameters θ_P, θ_V and θ_β for the pairwise potential terms and finally the higher order parameters θ_R and θ_α.

E. Online Update of the Training Database

Once the final results are obtained, the training database is updated online with incorrectly labeled examples using expert knowledge on the fly. It should be noted that "expert knowledge" here means using humans with domain knowledge to hand-label image patches related to various landcover classes. This is done as follows – After the generation of tree-cover maps from a certain number NAIP tiles (100, here), 10 (10% in general) maps are chosen at random and a reference to the NAIP tiles corresponding to these maps are saved to a database. An automated image-rendering tool (developed as part of our framework) allows experts to re-label misclassified image patches. The details of this image relabeling tool are provided in Section III-F. These re-labeled patches are then saved to the training database with the correct labeling. This improves the quality of results produced by the classifier in subsequent iterations. Choosing 10% of the image tiles randomly after the generation of every 100 tiles helps in maintaining the homogeneity of candidate selection for relabeling among the generated probability maps. 100 consecutively processed NAIP tiles cover a relatively small geographical area. Hence, a random 10% of the tiles represent a uniform selection of tiles from every spatial window and choosing every 100 images ensures a uniform selection from the entire mapped region. Every time the training dataset is updated, automated online training is done. This online update stage is very similar to a supervised post-processing of the classifier just like the boosting algorithm in machine learning [47] which recursively updates a strong learner by higher re-weighting of misclassifications by weak learners. The online update phase helps in reducing the False Positive Rate and at the same time, significantly increases the True Positive Rate. Figure 6 lists the variation of the User's and Producer's accuracy (omission and commission errors) with changing epochs of the online update algorithm. It can be seen that both

the User's and Producer's accuracy improve with the number of re-training epochs for the online update algorithm until 8 to 10 iterations. After that both accuracies remain stable up to about 14 epochs, after which they start dropping. This can be attributed to overfitting of the classification algorithm due to an excessive number of training samples fed into the supervised learner.

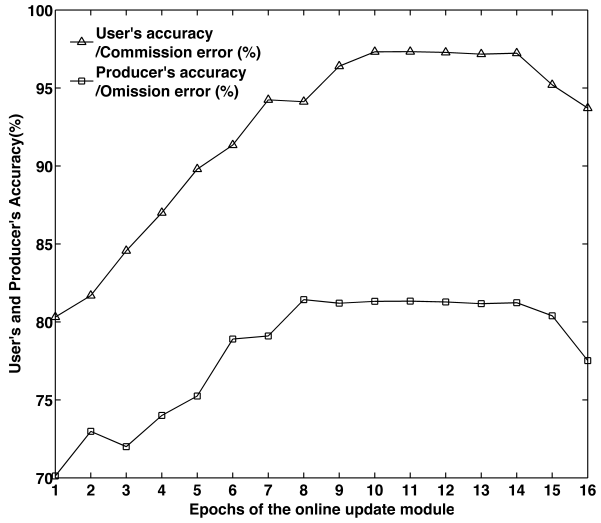
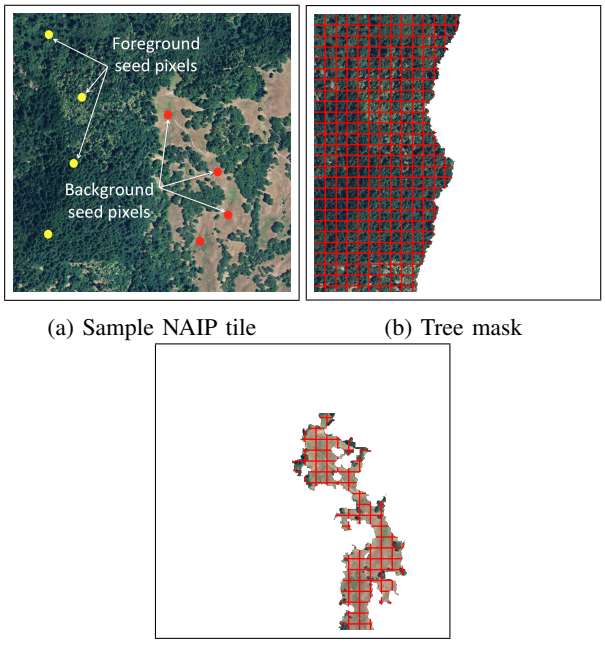


Fig. 6: Variation of Omission and Commision Errors with changing epochs of the online update algorithm.

F. Image Labeling and Re-labeling using Interactive Segmentation

We use an interactive image segmentation tool to extract and label the training samples as well as to re-label misclassified image patches in the online update phase described above. The interactive segmentation module uses a Random Walk based image segmentation algorithm first presented in [48]. In this method, at first a certain number of pixels are labeled as foreground and background pixels. These act as seeds for the segmentation algorithm. For any given unlabeled pixel in the original image, a random walk is initialized at the pixel. It is possible to determine the probability that the random walker starting at each unlabeled pixel will first reach one of the pre-labeled seed pixels. For k seed pixels, we get a $k\times1$ probability vector for each unlabeled pixel, each element of which represents the probability that the random walker starting at that pixel will reach the corresponding seed pixel first. Then we can assign a class label to each unlabeled pixel based on which element in the probability vector has the highest value. Figure 7 shows a sample NAIP tile with tree and non-tree masks generated by the Random walk based segmentation algorithm by selecting a certain number of foreground and background seed pixels corresponding to tree and non-tree areas. The foreground and background seed pixels are marked with yellow and red circles, respectively. The red boxes in Figure 7b and Figure 7c represent the training samples extracted from the image which are in turn saved to the training database with the correct label. Note that only complete boxes representing 4×4 training images are saved to the database while the rest are discarded. It should be noted that the masks shown in Figure 7 are generated by manually selecting a certain set of seed pixels. Choosing a different set of seeds can create a different segmentation mask.



(a) Sample NAIP tile (b) Tree mask

(c) Non-tree mask

Fig. 7: A sample NAIP tile with tree and non-tree cover masks generated by the Random Walker Segmentation module by selecting a certain set of foreground and background seed pixels. The small red squares indicate the training samples extracted from the image which are in turn saved to the training database with the correct label. Note that only complete squares representing 4×4 training images are saved to the database while the rest are discarded.

G. Implementation details and the High Performance Computing Architecture

We have deployed the abovementioned modules as stand alone on the NASA Earth Exchange (NEX) supercomputing cluster. The deployment was done through QSub routines and the Message Passing Interface (MPI). The data was accessed through a MySQL database. The NAIP tiles were processed in parallel in the cores of the NASA Earth Exchange High Performance Computing (NEX HPC) platform. Each node in the cluster having Harpertown CPUs consists of 8 gigabytes of memory and 8 cores with 3GHz processors per node [49]. In order to process 8 tiles in parallel, one tile per core, the memory requirement per core has to be kept lower than 1 Gigabyte. However, the problem arises with the use of the Statistical Region Merging (SRM) algorithm illustrated in Section III-A. Despite being fast, the algorithm has to store all the indices in memory while sorting them using radix sort as it makes decisions about region boundaries using global scene level image descriptors. This has space complexity of the order

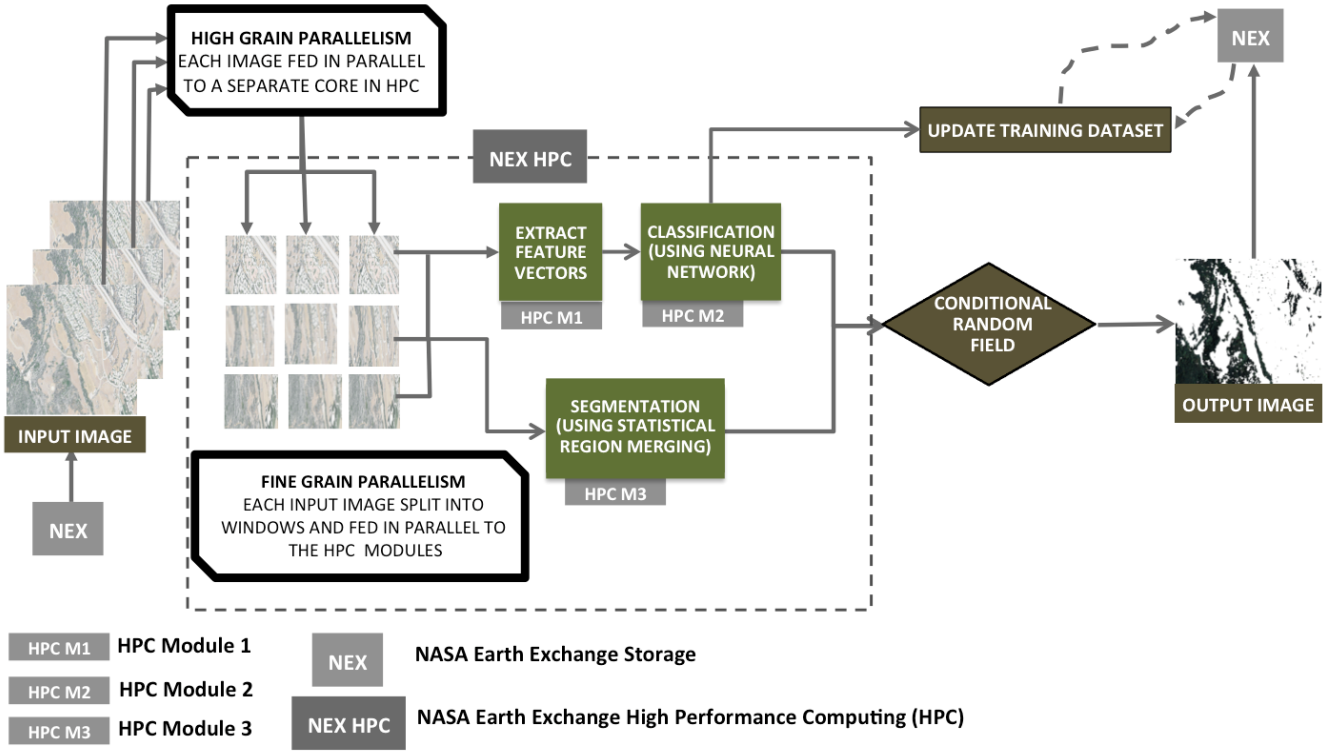


Fig. 8: High Performance Computing Architecture of our approach.

of $O(n^2)$, which indicates all image gradients in a $n \times n$ image, which is of the order of $\sim$3 Gigabytes for a typical NAIP tile. In order to address this memory-performance tradeoff, each image was split into $\lambda \times \lambda$ windows and then fed in a pipeline to each core in the HPC node. λ was chosen to be 256 for our experiments, because, higher values led to a higher memory requirement while lower values resulted in a substantial increase of processing time. The current architecture takes a maximum of approximately 4 hours to process each NAIP tile. The details of the architecture are illustrated in Figure 8.

Network Arch. Neurons/layer [Layers]	Classifier Accuracy 150 features(%)	Classifier Accuracy 22 features(%)
10 [2]	88.81	90.59
20 [2]	88.9	91.63
50 [2]	90.16	92.24
100 [2]	89.34	89.93
10 [3]	84.53	87.97
20 [3]	83.72	85.23
50 [3]	89.4	86.42
100 [3]	88.85	76.015

TABLE III: Classification Accuracy of the classifier with various network architectures using the entire set of 150 features and the set of 22 features derived using the feature selection method presented in Section III-B

	Densely Forested	Fragmented Forests	Urban Forests	Overall
Total Samples	12000	12000	12000	36000
Tree Samples	6000	6000	6000	18000
Non-tree Samples	6000	6000	6000	18000
True Positive Rate	85.87	88.26	73.65	82.59
False Positive Rate	2.21	0.99	1.98	1.73

TABLE IV: Preliminary classification accuracy assessment.

	Actual Class			
Predicted class	**Tree**	**Non-tree**	**Total pixels**	**User's accuracy**
Tree	14832	317	15149	97.9%
Non-tree	3168	17683	20851	84.8%
Total pixels	18000	18000	36000	
Producer's accuracy	82.4%	98.23%		90.31%

TABLE V: Confusion Matrix

IV. Results and Discussion

A rudimentary implementation of our framework produced encouraging results. We chose 1500 image tiles at random covering the following three types of landscapes (1) Densely forested, (2) Fragmented forests, and (3) Urban forested areas. A total of 36000 sampling points were chosen at random from the test images - 12000 samples for each land-cover type. The sample validation points are shown in Figure 19. The classifier accuracy was measured and averaged over 100 iterations. The results are tabulated in Table IV, which shows that our framework produces true positive rates higher than

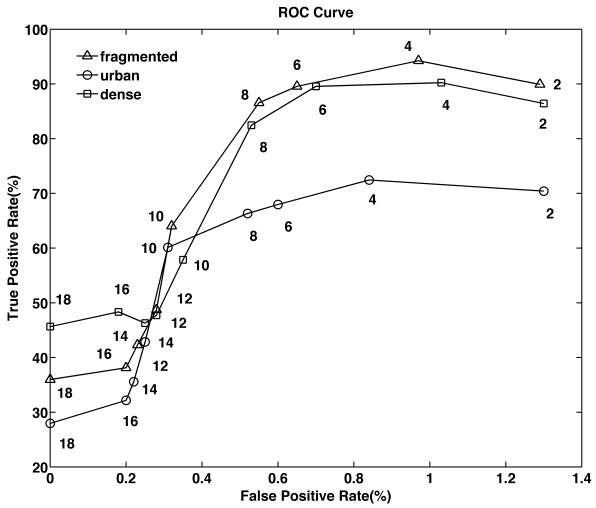


Fig. 9: ROC Curves for the three types of landscapes considered – fragmented, urban and densely forested (The numbers indicate the corresponding window sizes). One representative NAIP tile was chosen for each landscape – a densely forested tile from the Gasquet region in Northwestern California (7750m × 6380m), a tile with fragmented forests from the Susanville region in Northeastern California (7610m × 6000m), and an urban tile from a region in San Jose, California (7620m×6240m). 5000 points were chosen randomly from each image tile, labels for these representative points were assigned by a human expert and the tree cover maps were validated against these ground truth data to generate the true positive and false positive rate for the ROC Curves.

	NLCD-30m	NAIP-1m
Total samples	1000	1000
Tree samples	500	500
Non-tree samples	500	500
True Positive Rate(%)	72.31	87.13
False Positive Rate(%)	50.8	1.9
	NLCD-30m	**NAIP-1m**
Total samples	1000	1000
Tree samples	500	500
Non-tree samples	500	500
True Positive Rate(%)	2.88	79.64
False Positive Rate(%)	3.23	1.68

TABLE VI: Comparative results with NLCD for fragmented forests (top) and urban forested areas (bottom).

85% for both densely forested and fragmented areas. However, the results degraded for urban areas where we achieved correct detection rates of about 74%. This can be attributed primarily to the presence of trees in urban regions with canopies having dimensions of less than 4m in any direction (the value of our neighborhood parameter τ). However, experimenting with τ values less than 4 did not improve the performance of the framework as evident from the ROC Curve analysis presented in Figure 9. The ROC curves represent the change in True Positive Rate with the change in False Positive Rate by varying a certain adjustable parameter of a model (window size τ here). The degradation in performance of the framework after the adjustable parameter τ is increased beyond a certain

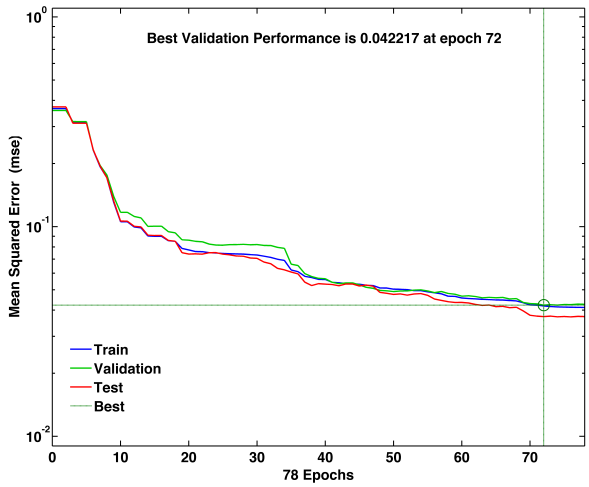
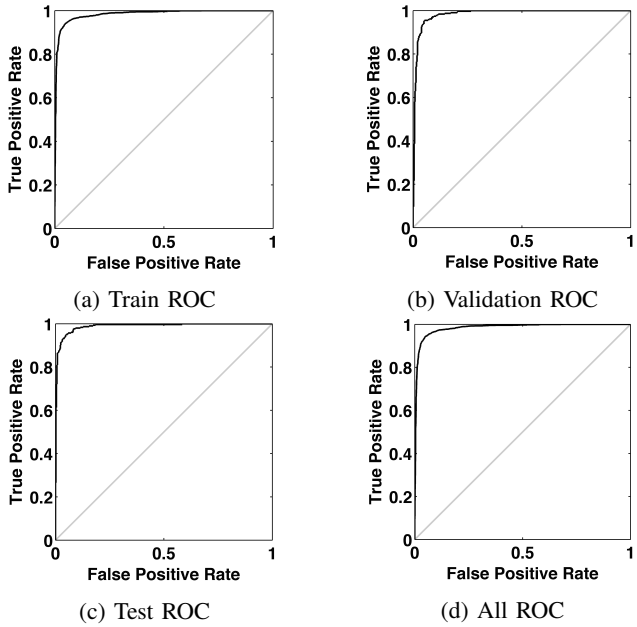


Fig. 10: Performance of the Neural Network training algorithm for a set of (randomly chosen) 3500 training samples, 750 validation samples and 750 test samples from a NAIP tile from Blocksburg, California (7610m × 6000m). The X-axis marks the iterations/epochs of the training algorithm, while the mean-squared error is noted along the Y-axis. The blue line indicates the mean squared error at various epochs during the training phase of the Neural Network, the green line indicates the mean squared validation error and the red line indicates the mean squared test error. The best performance is attained at iteration 72.

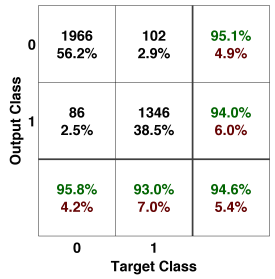


(a) Train ROC (b) Validation ROC

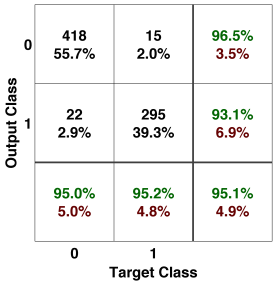
(c) Test ROC (d) All ROC

Fig. 11: ROC Curves for the Neural Network Training algorithm for the same dataset used in Figure 10 – ROC curve generated for training, validation, test sets and taking the mean of the true positive rate and false positive rate of the training, validation and test dataset respectively.

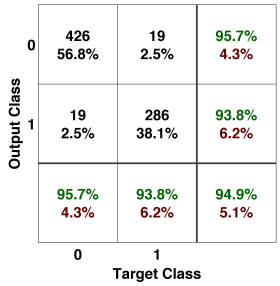
value (4, here) can be attributed to a flat response from the



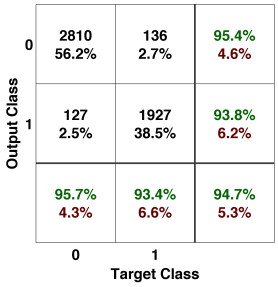
(a) Training Confusion Matrix



(b) Validation Confusion Matrix



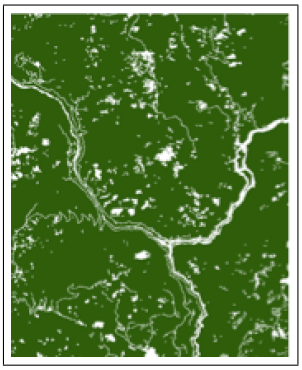
(c) Test Confusion Matrix



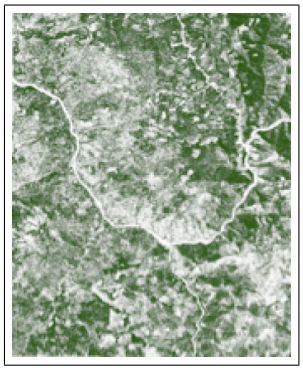
(d) All Confusion Matrix

Fig. 12: Confusion Matrix for the Neural Network training algorithm for the dataset used in Figure 10 and Figure 11. A total of 3500 samples are used for training, 750 samples for validation and 750 samples for testing.

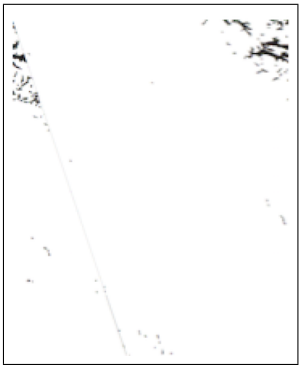
classification module due to reduced discriminative power of the class separability criterion. Table V shows the confusion matrix, where, the columns represent the instances of the actual class while the rows represent the instances of the predicted class. It can be seen from the table that our framework produces an overall accuracy of 90.31%. ROC Curves generated by varying the neighborhood parameter τ (defined in Section III-C) for various land-cover types (fragmented forests, densely forested and urban areas) is shown in Figure 9. τ was chosen to be 4 by doing a ROC Curve analysis as illustrated in Figure 9, such that the True Positive Rate is maximized. The error minimization of the Neural Network training algorithm is shown in Figure 10. It can be seen that the best validation performance with a mean-squared error of 0.14191 is attained at epoch 30 of the Neural Network training algorithm. Figure 11 and Figure 12 show the ROC Curves and confusion matrix for the Neural Network training algorithm. Comparative studies with the 2001 National Land Cover Data (NLCD) [50] 30-m are enumerated in Table VI. It can be seen that results from our probabilistic framework outperforms NLCD by nearly 15% for fragmented forests and nearly 77% for urban areas. Figure 13 shows the sample output for two tiles - one for a fragmented area in Hoopa, California, north of the Klamath River and another for an urban area in San Jose, California using NLCD and NAIP. The comparative studies with NLCD clearly show that our algorithm outperforms the NLCD approach for the classes of land-cover types (tree-cover areas) considered in this study. This can be primarily attributed to the higher resolution of the NAIP dataset as compared to LANDSAT imagery which has 30 times lower resolution than



(a) NLCD output for Hoopa, California



(b) NAIP output for Hoopa, California



(c) NLCD output for San Jose, California

(d) NAIP output for San Jose, California

Fig. 13: Results for an image tile with fragmented trees in Hoopa, California, north of the Klamath River and an urban area in San Jose, California for NLCD and NAIP.

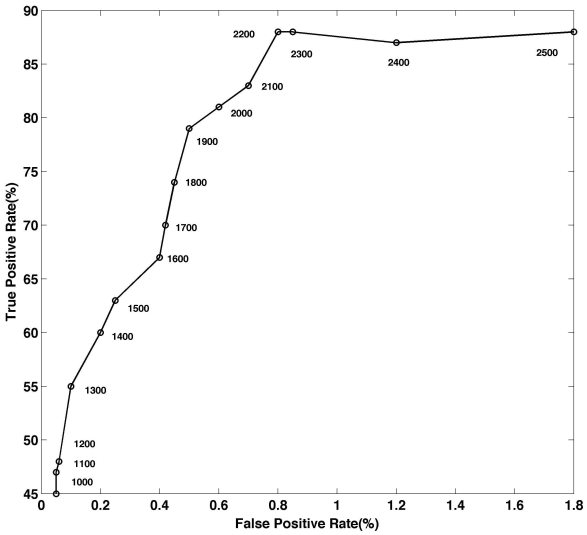


Fig. 14: ROC Curve generated by varying the sample size of the training data. The region of study was the same as that of Figure 10 – an area with fragmented forests in the Blocksburg region in northwestern California (7610 m $\times$ 6000 m). The numbers denote the number of training samples.

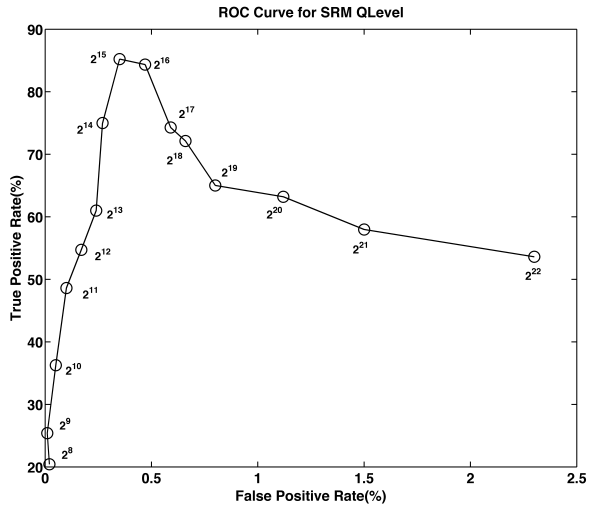


Fig. 15: ROC Curve generated by varying the Quantization Level in the SRM algorithm for the same area as considered in Figure 14. The training sample size is 5000 - consisting of 2500 tree and 2500 non-tree samples chosen at random from the NAIP tile. The numbers denote the corresponding Qlevel values.

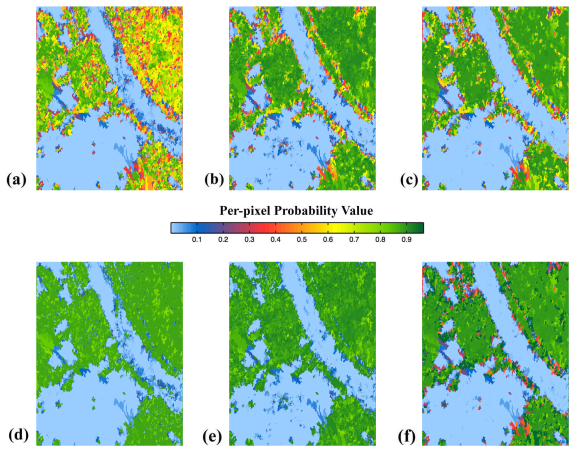


Fig. 16: Probability Maps for the probabilistic NN classifier results (a-c) and CRF output (d-f) for various training sample sizes (1300,1800 and 2500 samples per class from left to right) for a sample NAIP tile from Blocksburg, California. The color maps show the probabilities on a scale of 0 to 1. The probability maps for the NN represent the probability of a pixel being classified as a tree by the Neural Network and the probability maps for the CRF output represent the final probabilities assigned to the pixels by the CRF labeling algorithm. A pixel assuming a value of 1 in the probability map is marked as a tree and a pixel assuming a value of 0 is marked as a non-tree, with intermediate pixels values being marked as tree/non-tree according to the problem (here, we use a 50% threshold, i.e., a pixel is marked as tree if probability is greater than 0.5).

NAIP. Moreover, our structured prediction framework helps us in decreasing false positives by considering intra and inter-class votes towards the labeling algorithm. Figure 14 shows the ROC Curve generated by varying the sample size of the training dataset from 1000 samples per class to 2500 per class in steps of 100 for a particular tile in the NAIP dataset with fragmented forest cover. The flat response towards the end of the curve indicates that increasing the training sample size has minimal effect beyond a point, which is around 2200 training samples for this exercise. This shows the robustness of the algorithm and the fact that minimal amount of training samples is sufficient for training the classifier. Figure 15 shows a ROC Curve generated by varying the Quantization Level (Qlevel) in the Statistical Region Merging algorithm. The analysis of this curve helped us in selecting a Qlevel of 2^{15} for our framework as is evident from the maxima attained at $Q = 2^{15}$ in the figure. Figure 16 shows the probability maps generated by the classification algorithm and the Conditional Random Field. Figure 17 shows the final probability map generated for a sample tile from the NAIP database using our framework. Figure 27 shows a sample NAIP tile from the Blocksburg area in California and the corresponding binary tree-cover map generated by our framework. To generate this binary tree cover map, the output probability map from the CRF is filtered with a threshold τ to eliminate pixels with output probability less than the threshold. The threshold τ is set as 0.5. For the CRF output probability for a pixel x being $Pr(x)$, the final output map value for pixel x,

$$O(x) = \begin{cases} 1, & \text{if } Pr(x) \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

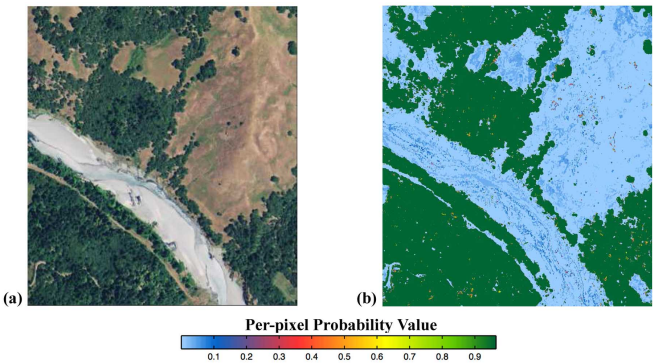


Fig. 17: (a) A sample image with (b) the final Probability Map generated by our framework for a region in Blocksburg, California. This final probability map is the same as the map generated by the CRF based labeling algorithm as shown in Figure 16. The CRF algorithm combines the probability values assumed by the classifier outputs for individual pixels and generates the final probability map as shown above.

Figure 28 shows the final tree-cover map generated by our framework for the whole of California covering 11,095 NAIP tiles.

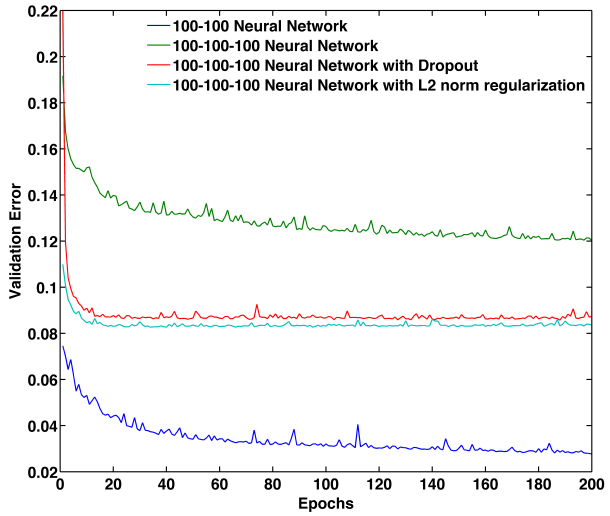


Fig. 18: The validation error rate for the same dataset used in Figure 10 for the 100-100 and 100-100-100 neural networks without regularization, and the 100-100-100 neural network with L2 norm regularization and Dropout.

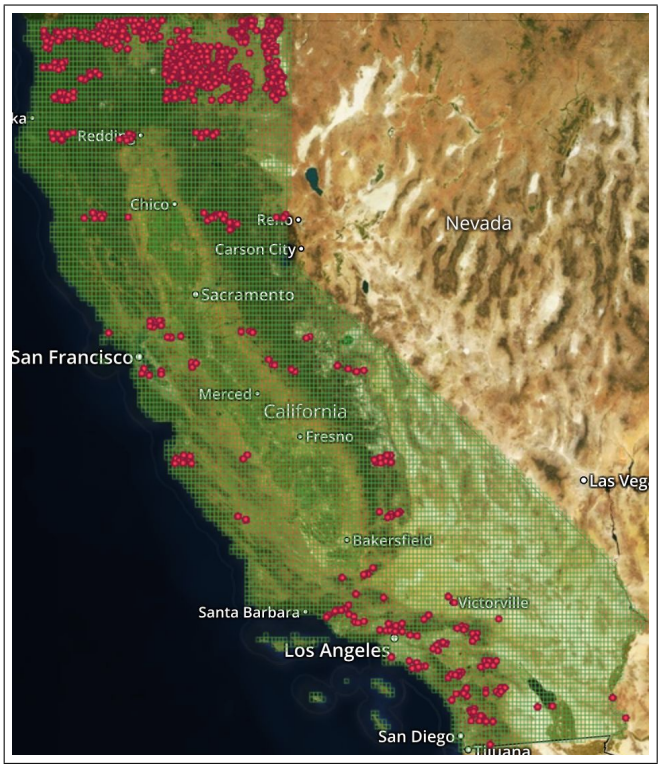


Fig. 19: A satellite image showing the validation points chosen for our experiments over California. The red circles denote the validation points. A total of 36000 sampling points were chosen to represent densely forested areas, fragmented forests and urban forested areas of California. The green grid represents the individual NAIP tiles. In order to display the locations from which the validation points were sampled, multiple points were clustered into subgroups and hence each red circle in the figure represents multiple validation points.

A. Validation with High Resolution Airborne LiDAR Canopy Height Model

The tree-cover maps generated by the Canopy Height Model (CHM) from the LiDAR data and the probabilistic framework for the NAIP dataset for both Area 1 and Area 2 are presented in Figure 20.

A Random Forest (RF) based classifier was independently trained on the same dataset used for the probabilistic Neural Network classifier. The RF classifier was implemented using a random forest package [51], available in the R interface [52]. The number of trees, node size and maximum number of terminal node trees in the forest were varied in iterations to achieve a stable solution with maximum accuracy. The number of trees was set to 250, node size was set to 5 and maximum number of terminal nodes was set to 500. Number of trees is selected in such a way so that every input row gets predicted at least few times. Node size is usually set as a small number as setting this number larger causes smaller trees to be grown although it may take less time. The number of terminal nodes decides the maximum possible size of the growth of the trees (also subject to limits by node size). The final values of the parameters were obtained empirically depending on the minimum execution time and memory requirements when further parameter variations did not increase the accuracy. A sliding window analysis (with a window size of 50 pixels) was performed on the two scenes and the percentage of tree-cover pixels and non-tree pixels are presented in Figure 21 and Figure 22. As can be seen from Figure 21, the tree-cover predictions generated by both the Probabilistic Neural Network (NN) framework and the Random Forest (RF) based framework have a high positive correlation with LiDAR, while NLCD produces significantly less accurate results, having error rate more than 40% on average with LiDAR tree cover estimates considered as ground truth, while the Neural Network classifier produces a mean error rate less than 5% with the same ground truth data. The Random Forest implementation has a higher error rate averaging around 15%. An evaluation of the True Positive Rates and False Positive Rates for the NN and RF algorithms for the NAIP data and NLCD algorithm for LANDSAT data for Area 1 is enumerated in Figure 23a and 23b. The significantly higher values of the True Positive Rates for NLCD can be attributed to the significant loss of resolution of the dataset. Therefore, a tree-cover region is classified in its entirety and approximated as either a tree-cover or non-tree region based on whether most of the pixels are tree or not. In other words, in NLCD, a single pixel represents an area of 30 sq. m. The highest resolution attainable for NLCD is of the order of at least 8-10 full-grown trees. This argument can be substantiated by the NLCD output shown in Figure 13. As can be seen in Figure 13, non-tree regions are approximated as tree-cover regions in a comparatively densely forested region while for urban areas, tree-cover regions are classified as non-tree regions owing to its presence amidst a sparsely forested landscape. Figure 13 explains the high

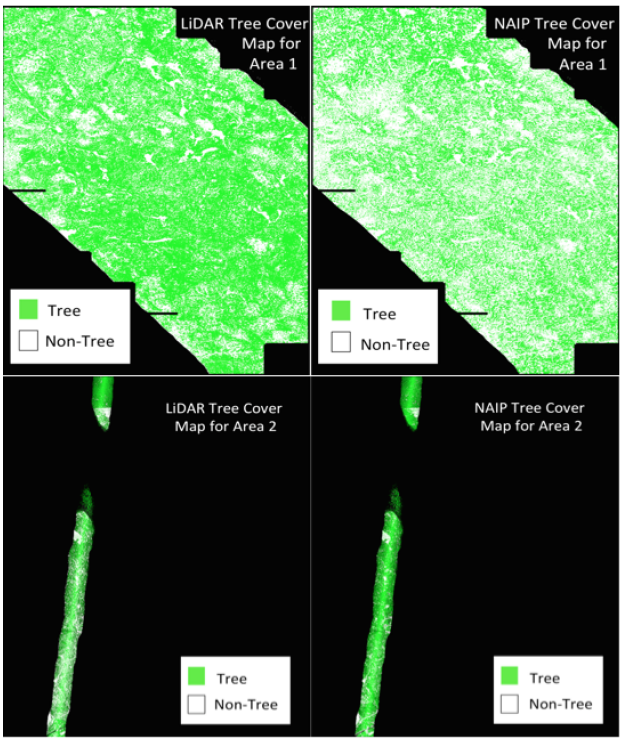


Fig. 20: The final tree-cover maps generated using LiDAR (left) and NAIP (right) for Area 1 (top) and Area 2 (bottom). The green regions represent tree cover areas, the white regions represent non-tree areas and the black regions represent the areas with null values in the LiDAR data (these black regions were masked out from the NAIP tree cover maps for comparative studies with the corresponding LiDAR maps).

False Positive Rate for NLCD as illustrated in Figure 23b. The same validation technique was applied to Area 2 and the results of the percentage of tree and non-tree pixels are presented in Figure 24 and Figure 25. As highlighted by both figures, our probabilistic Neural Network based framework produces near optimal results, which are highly correlated to the LiDAR output and outperforms the Random Forest based classification algorithm. This is substantiated by the True Positive and False Positive Rates enumerated in Figure 26a and 26b. It can be easily seen that the algorithm produces a significantly high accuracy with mean True Positive Rate as high as 97% and mean False Positive Rate around 8%. On the other hand, though NLCD produces True Positive Rate of around 92% (almost same as NN) for the area, but it produces False Positive Rate as high as 88%. It also outperforms the Random Forest classifier, which produces True Positive Rate of around 82%.

V. Conclusions and Future Directions

Our probabilistic framework has been proved to be a useful tool for analyzing 1-m NAIP imagery for large-scale tree-cover mapping. Preliminary results on the NAIP tiles for California have produced positive detection rates of ∼86% for densely forested areas and ∼74% for urban areas. Comparative studies with NLCD show the effectiveness of our approach towards

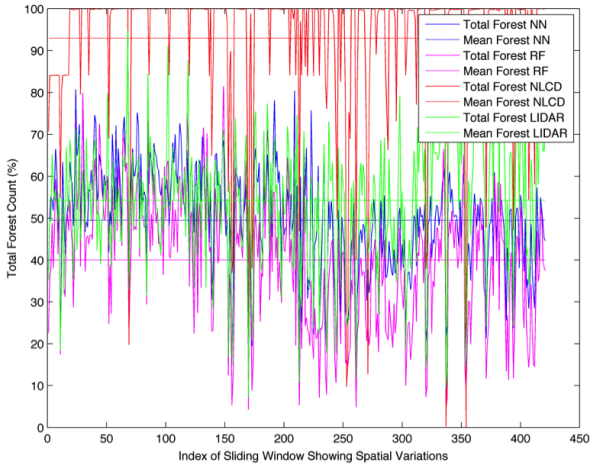


Fig. 21: Percentage of forest cover obtained using Neural Network and Random Forest (for NAIP), and NLCD and LiDAR in Area 1 (the Teakettle Experimental Forest in the western Sierra Nevada mountain range, California). A 50×50 sliding window was used to obtain the percentage of tree-cover pixels in both NAIP and NLCD with LiDAR as the ground truth.

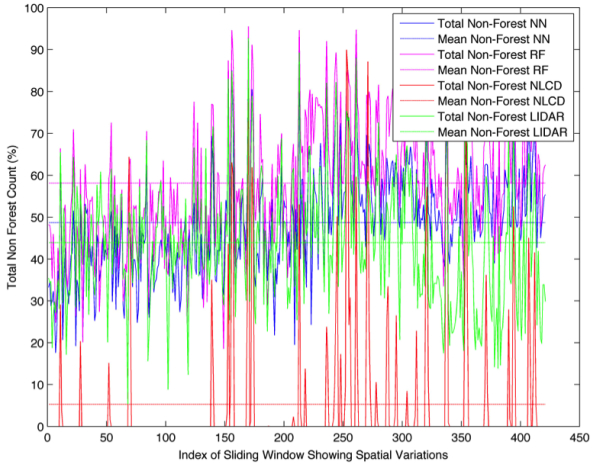
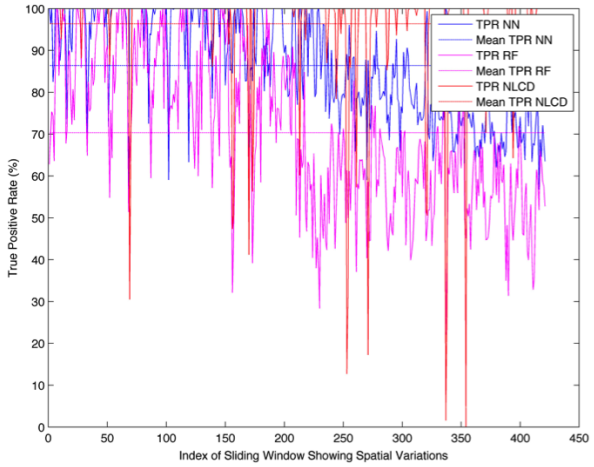
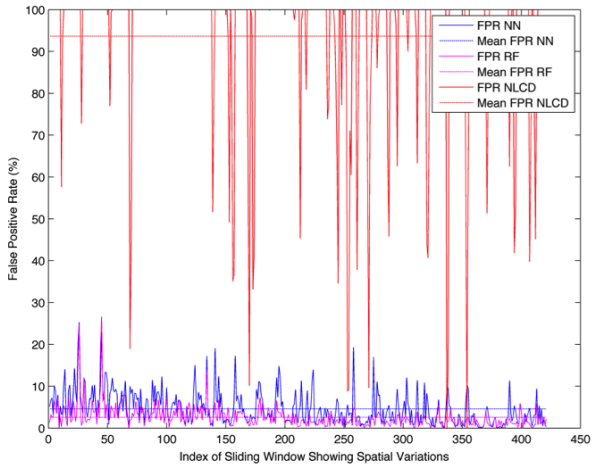


Fig. 22: Percentage of non-forest area obtained using Neural Network and Random Forest (for NAIP), and NLCD and LiDAR in Area 1 (same as the area in Figure 21). The sliding window size was 50×50.

creating the NAIP 1-m "Golden Dataset". Validation with high-resolution airborne LiDAR data shows average positive detection rates of around 83% and average false positive rate as low as 10%. This proves the effectiveness of our approach in generating high-resolution tree-cover maps for the entire country. The algorithm scales seamlessly to millions of scenes and can handle high variations, which is often the case for aerial imagery. The use of handcrafted features extracted from the Hue, Saturation, Intensity and NIR channels provides a useful framework for classifying NAIP imagery. The integration of



(a) True Positive Rate



(b) False Positive Rate

Fig. 23: True Positive Rate (TPR) and False Positive Rate (FPR) of Neural Network and Random Forest (for NAIP) and NLCD with LiDAR as ground truth for Area 1 (same as the area in Figure 21). The sliding window size was 50×50.

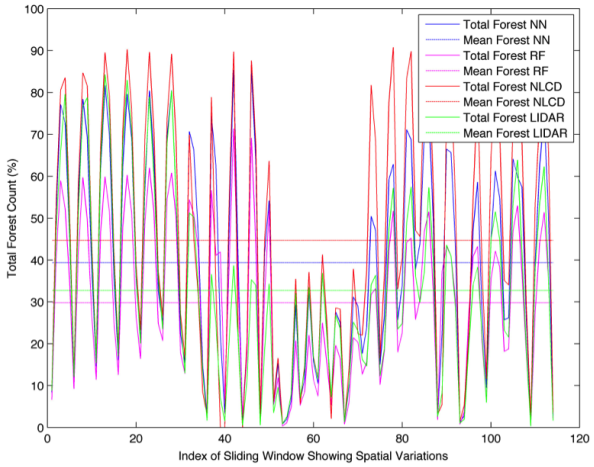


Fig. 24: Percentage of forest cover obtained using Neural Network and Random Forest (for NAIP), and NLCD and LiDAR for Area 2 (the Chester area in California). The sliding window size was kept as 50×50.

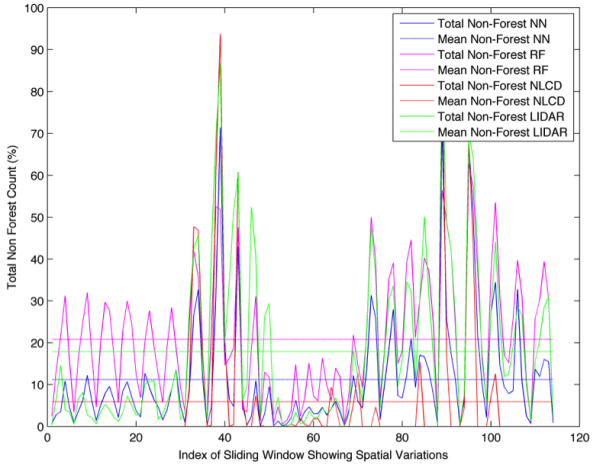


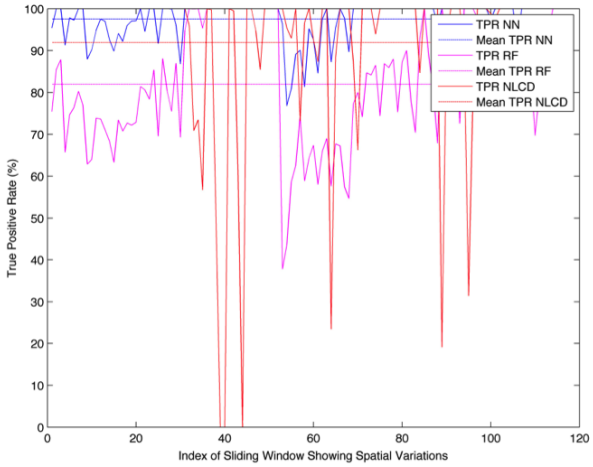
Fig. 25: Percentage of non-forest area obtained using Neural Network and Random Forest (for NAIP), and NLCD and LiDAR for Area 2 (same as Figure 24) with the sliding window size kept constant at 50×50.

the structured prediction framework based on Conditional Random Field helped to increase the True Positive Rates while reducing the False Positive Rate by incorporating classifier outputs from the neighboring pixels located within the same neighborhood system. The Near Infrared channel in the NAIP dataset was also useful in segregating regions with chlorophyll from others and proved to be a very useful discriminative feature for addressing the tree/non tree classification of the 1-m NAIP dataset. We plan to create tree-cover maps for the entire continental United States and incorporate and train the framework for more classes, such as different types of tree canopies, grasslands, croplands, etc., in the future. We also plan to integrate man-made classes like roads, rooftops, parking lots, etc. into the framework.

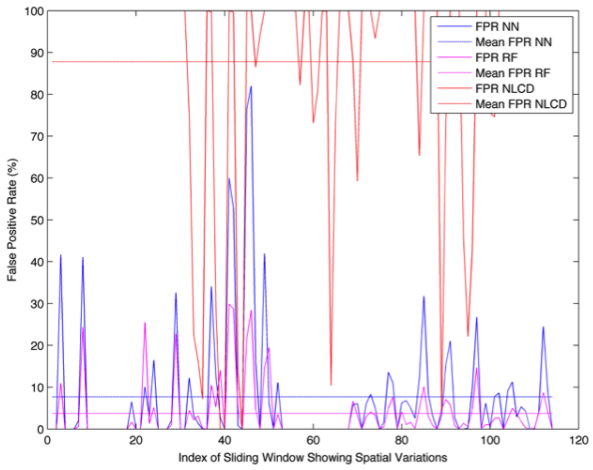
ACKNOWLEDGMENT

The project is supported by NASA Carbon Monitoring System through Grant #NNH14ZDA001-N-CMS. We are grateful to the United States Department of Agriculture for providing us the National Agriculture Imagery Program (NAIP) airborne imagery dataset for the Continental United States.

This research was partially supported by the Cooperative Agreement Number NASA-NNX12
AD05A, CFDA Number 43.001, for the project identified as "Ames Research Center Cooperative for Research in Earth Science and Technology (ARC-CREST)". Any opinions findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect



(a) True Positive Rate



(b) False Positive Rate

Fig. 26: True Positive Rate (TPR) and False Positive Rate (FPR) of the Neural Network and Random Forest (for NAIP) and NLCD with LiDAR as ground truth for Area 2 (same as the area in Figure 21). The sliding window size was 50×50.

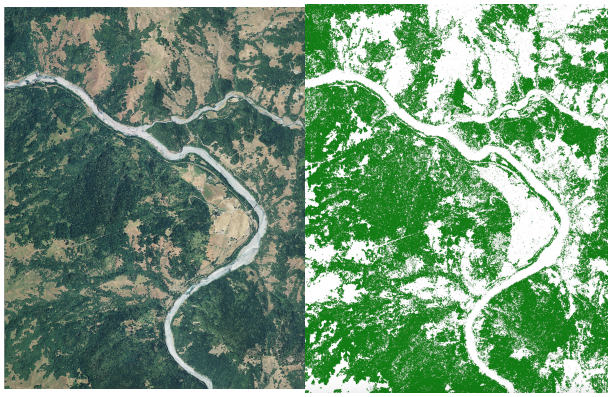


Fig. 27: A sample NAIP tile (left) and the corresponding binary tree cover mask (right). The white pixels denote non-tree areas while the green pixels denote the tree-cover areas.

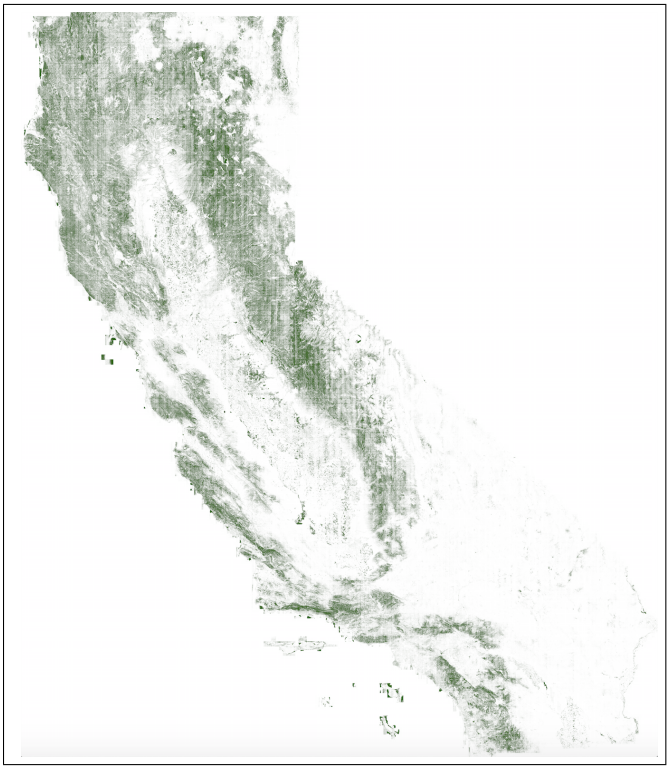


Fig. 28: The final tree cover map generated by our framework for the whole of California covering 11,095 NAIP tiles. The green pixels denote the tree-cover areas while the white pixels denote the non-tree areas.

that of NASA or the United States Government.

We are thankful to the reviewers for their constructive comments and suggestions which helped in improving this manuscript significantly.

REFERENCES

[1] H. G. Akcay and S. Aksoy, "Automatic detection of geospatial objects using multiple hierarchical segmentations." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2097–2111, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/tgrs/tgrs46.html#AkcayA08

[2] X. Sun, H. Wang, and K. Fu, "Automatic detection of geospatial objects using taxonomic semantics," *Geoscience and Remote Sensing Letters, IEEE*, vol. 7, no. 1, pp. 23–27, Jan 2010.

[3] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3706–3715, Dec 2006. [Online]. Available: http://vision.ece.ucsb.edu/publications/sitaram_TGRS06.pdf

[4] D. Guo, H. Xiong, V. Atluri, and N. Adam, "Object discovery in high-resolution remote sensing images: a semantic perspective," *Knowledge and Information Systems*, vol. 19, no. 2, pp. 211–233, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10115-008-0160-4

[5] A. C. Holt, E. Y. W. Seto, T. Rivard, and P. Gong, "Object-based detection and classification of vehicles from highresolution aerial photography," *Photogrammetric Engineering and Remote Sensing*, vol. 75, no. 7, pp. 871–880, 2009.

[6] J. Li, R. Nevatia, and S. Nornoha, "User assisted modeling of buildings from aerial images," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, vol. 2, 1999, pp. –279 Vol. 2.

[7] E. Sharon, A. Brandt, and R. Basri, "Segmentation and boundary detection using multiscale intensity measurements." in *CVPR (1)*. IEEE Computer Society, 2001, pp. 469–476. [Online]. Available:

http://dblp.uni-trier.de/db/conf/cvpr/cvpr2001-1.html#SharonBB01

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[9] Z. Tu, X. Chen, A. L. Yuille, and S. chun Zhu, "Image parsing: Unifying segmentation, detection, and recognition," in *International Journal of Computer Vision*, vol. 63, 2003, pp. 18–25.

[10] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing," in *Initial Values of the Adaptive Weights, International Joint Conference of Neural Networks*, 1990, pp. 21–26.

[11] C. Leistner, H. Grabner, and H. Bischof, "Semi-supervised boosting using visual similarity learning," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, June 2008, pp. 1–8.

[12] D. S. Chapman, A. Bonn, W. E. Kunin, and S. J. Cornell, "Random forest characterization of upland vegetation and management burning from aerial imagery," *Journal of Biogeography*, vol. 37, no. 1, pp. 37–46, 2010. [Online]. Available: http://dx.doi.org/10.1111/j.1365-2699.2009.02186.x

[13] P. Gislason, J. Benediktsson, and J. Sveinsson, "Random forest classification of multisource remote sensing and geographic data," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International*, vol. 2, Sept 2004, pp. 1049–1052 vol.2.

[14] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 294 – 300, 2006, pattern Recognition in Remote Sensing (PRRS 2004). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865505002242

[15] R. Archibald and G. Fann, "Feature selection and classification of hyperspectral images with support vector machines," *Geoscience and Remote Sensing Letters, IEEE*, vol. 4, no. 4, pp. 674–677, Oct 2007.

[16] M. A. Friedl, D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley, and X. Huang, "Modis collection 5 global land cover: Algorithm refinements and characterization of new datasets," *Remote Sensing of Environment*, vol. 114, pp. 168–182, 2009.

[17] J. D. Wickham, S. V. Stehman, L. Gass, J. Dewitz, J. A. Fry, and T. G. Wade, "Accuracy assessment of NLCD 2006 land cover and impervious surface," *Remote Sensing of Environment*, vol. 130, pp. 294–304, 2013.

[18] D. A. Pouliot, D. J. King, F. W. Bell, and D. G. Pitt, "Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration," *Remote Sens. Environ.*, vol. 82, no. 2-3, pp. 322–334, 2002.

[19] R. Komura, M. Kubo, and K.-i. Muramoto, "Delineation of tree crown in high resolution satellite image using circle expression and watershed algorithm," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International*, vol. 3, Sept 2004, pp. 1577–1580 vol.3.

[20] D. G. Leckie, F. A. Gougeon, N. Walsworth, and D. Paradine, "Stand delineation and composition estimation using semi-automated individual tree crown analysis," *Remote Sensing of Environment*, vol. 85, no. 3, pp. 355 – 369, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425703000130

[21] F. A. Gougeon, "A crown-following approach to the automatic delineation of individual tree crowns in high spatial resolution aerial images," *Canadian Journal of Remote Sensing*, vol. 21, no. 3, pp. 274–284, 1995. [Online]. Available: http://cat.inist.fr/?aModele=afficheN\&\#38;cpsidt=2918322

[22] V. Mnih and G. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, September 2010.

[23] C. Vaduva, I. Gavat, and M. Datcu, "Deep learning in very high resolution remote sensing image information mining communication concept," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, Aug 2012, pp. 2506–2510.

[24] G. Liknes, C. Perry, and D. Meneguzzo, "Assessing tree cover in agricultural landscapes using high resolution aerial imagery," *J. Terres. Obs*, vol. 2, pp. 38–55, 2010.

[25] WWW2, "Naip," http://www.fsa.usda.gov/Internet/FSA_File/naip_2009_info_final.pdf.

[26] D. AG., http://www.definiens.com, 2008.

[27] WWW3, "Modis," http://vip.arizona.edu/documents/MODIS/MODIS_VI_UsersGuide_01_2012.pdf.

[28] L. Duncanson, B. Cook, G. Hurtt, and R. Dubayah, "An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems," *Remote Sensing of Environment*, vol. 154, pp. 378–386, Nov 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425714000984

[29] B. D. Cook, L. A. Corp, R. F. Nelson, E. M. Middleton, D. C. Morton, J. T. McCorkel, J. G. Masek, K. J. Ranson, V. Ly, and P. M. Montesano, "NASA Goddards LiDAR, Hyperspectral and Thermal (G-LiHT) Airborne Imager," *Remote Sensing*, vol. 5, no. 8, pp. 4045–4066, 2013. [Online]. Available: http://www.mdpi.com/2072-4292/5/8/4045

[30] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1452–1458, 2004.

[31] T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, "Fast image segmentation based on k-means clustering with histograms in hsv color space." in *MMSP*. IEEE Signal Processing Society, 2008, pp. 322–325.

[32] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973. [Online]. Available: http://dx.doi.org/10.1109/tsmc.1973.4309314

[33] L. kiat Soh and C. Tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *Geoscience and Remote Sensing, IEEE Transactions on*, pp. 780–795, 1999.

[34] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Can. J. Remote Sensing*, vol. 28, no. 1, pp. 45–62, 2002.

[35] A. Huete, K. Didan, T. Miura, E. Rodriguez, X. Gao, and L. Ferreira, "Overview of the radiometric and biophysical performance of the {MODIS} vegetation indices," *Remote Sensing of Environment*, vol. 83, no. 12, pp. 195 – 213, 2002, the Moderate Resolution Imaging Spectroradiometer (MODIS): a new generation of Land Surface Monitoring. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425702000962

[36] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering, "Monitoring vegetation systems in the great plains with ERTS," *NASA Goddard Space Flight Center 3d ERTS-1 Symposium*, pp. 309–317, 1974.

[37] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation," *Remote Sensing of Environment*, vol. 8, no. 2, pp. 127 – 150, 1979. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0034425779900130

[38] Y. Kaufman and D. Tanre, "Atmospherically resistant vegetation index (ARVI) for EOS-MODIS," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 30, no. 2, pp. 261–270, Mar 1992.

[39] Y.-L. Boureau, J. Ponce, and Y. Lecun, "A theoretical analysis of feature pooling in visual recognition," in *27th International Conference on Machine Learning, Haifa, Isreal*, 2010.

[40] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[41] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1207.html#abs-1207-0580

[42] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[43] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813

[44] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.

[45] P. Kohli, L. Ladický, and P. H. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vision*, vol. 82, no. 3, pp. 302–324, May 2009. [Online]. Available: http://dx.doi.org/10.1007/s11263-008-0202-0

[46] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001. [Online]. Available: http://dx.doi.org/10.1109/34.969114

[47] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jul. 1990. [Online]. Available: http://dx.doi.org/10.1023/A:1022648800760

[48] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2006.233

[49] WWW4, "Pleiades," https://nex.nasa.gov/nex/static/media/other/Intro_to_Pleiades.pdf.

[50] J. Vogelmann, T. Sohl, P. Campbell, and D. Shaw, "Regional land cover characterization using Landsat Thematic Mapper Data and Ancillary Data Sources," *Environmental Monitoring and Assessment*, vol. 51, no. 1-2, pp. 415–428, 1998. [Online]. Available: http://dx.doi.org/10.1023/A%3A1005996900217
[51] A. Liaw and M. Weiner, "Classification and regression by random forests," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[52] WWW5, "R-project," http://www.r-project.org.

Saikat Basu (S'14) is a PhD candidate in the Computer Science Department at Louisiana State University, Baton Rouge, USA. He received his BS in Computer Science from National Institute of Technology, Durgapur, India in 2011. His primary research interests include applications of Computer Vision and Machine Learning to satellite imagery data and using High Performance Computing architectures to address bigdata analytics problems for very high-resolution satellite datasets. He is particularly interested in the applications of various Deep Learning frameworks to satellite image understanding and representations.

Sangram Ganguly is a senior research scientist at the Bay Area Environmental Research Institute and research staff at the Biosphere Science Branch at NASA Ames Research Center, Moffett Field, California. He received his PhD from Boston University, USA in 2008 and his BS and MS degrees in geosciences from the Indian Institute of Technology (IIT), Kharagpur, India in 2004. His research interests include cloud computing solutions for big data science and analytics, machine learning, advanced satellite remote sensing and image analytics and climate sciences.

Ramakrishna R. Nemani is a Senior Earth Scientist at the Ecological Forecasting Laboratory at NASA Ames Research Center in California, USA. He received his PhD from the University of Montana, USA in 1987, MS from the Punjab Agricultural University, Ludhiana, India in 1982 and BS from the Andhra Pradesh Agricultural University, Bapatla, India in 1979. His research interests include Ecological Forecasting and Collaborative computing in the Earth sciences.

Supratik Mukhopadhyay is a Professor in the Computer Science Department at Louisiana State University, USA. He received his PhD from the Max Planck Institut fuer Informatik and the University of Saarland in 2001, MS from the Indian Statistical Institute, India and his BS from Jadavpur University, India. His areas of specialization include formal verification of software, inference engines, bigdata analytics distributed systems, parallel computing framework and program synthesis.

Gong Zhang is a Research Scientist in the Earth Sciences division at NASA Ames Research Center, California, USA. He received his PhD in Ecology from Utah State University, USA in 2012. He received his MS in Geography from Peking University, China in 2007 and BS in Meteorology from Nanjing University, China in 2004. His research interests include Mathematical Programming, Stochastic optimization, High performance computing, Data visualization, LiDAR/RADAR data analysis, and Atmosphere/Biophysics/Hydrology Modeling.

Cristina Milesi is a Research Scientist in the Ecological Forecasting Laboratory at NASA Ames Research Center, California, USA. She received her PhD in Ecosystem Modeling and Remote Sensing from the University of Montana, USA in 2004 and BSc in Biology from the University of Trieste in 1995. Her research interests include Remote sensing and ecosystem process modeling of the interactions between water, carbon and nitrogen cycling, Urbanization and its impacts on Earth's biophysical and biogeochemical processes and Human population and the environment.

Andrew Michaelis is a software engineer with the NASA Ames Research Center & University Corporation Monterey. He received his BS in Computer Science from the University of Montana in 2002. His research interests include Modeling and Analysis Compute Environments, utilizing Virtualization Technology in the Climate and Earth Systems Science domain and development and deployment of Collaborative computing platforms in the Earth sciences.

Petr Votava is a Sr. Software Engineer at NASA Ames Research Center & University Corporation at Monterey Bay. He received his M.S. and B.S. in Computer Science from the University of Montana, USA. His research interests include Satellite data processing, software engineering, operating systems and architecture, parallel processing, sensor web, Data mining/anomaly detection, semantic web, information extraction and natural language processing.

Ralph Dubayah is a Professor of Geographical Sciences at the University of Maryland, College Park. He received his B.A. in 1982 from the University of California, Berkeley, and his M.A. and Ph.D. degrees from the University of California, Santa Barbara in 1985 and 1990. His main areas of interest are ecosystem characterization for carbon modeling, habitat and biodiversity studies, land surface energy and water balance modeling, spatial analysis and remote sensing science.

Laura Duncanson is a PhD candidate in Geographical Sciences at the University of Maryland, College Park, USA. She received her B.Sc. Honors in Earth Systems Sciences in 2007 from Queens University at Kingston, Ontario, MSc in Geography from the University of Victoria in 2009. Her research is focused on increasing the understanding of forest structure and dynamics and high relief forests, in particular.

Bruce Cook is a Physical Scientist at NASAs Goddard Space Flight Center, Code 614.4, Greenbelt, MD, USA. He received his PhD in Natural Resources - Assessment, Monitoring, and Geospatial Analysis from the University of Minnesota, St. Paul, MN in 2008, MS in Soil, Water, and Climate from the University of Minnesota, St. Paul, MN in 1990 and BS in Biology from Huntington College, IN in 1985. His research is focused on LiDAR remote sensing and terrestrial carbon cycling in wetland and upland ecosystems, with expertise in field/laboratory and flux tower measurements, satellite remote sensing of plant communities, and terrestrial ecology.

Yifan Yu is a PhD candidate in the Atmospheric and Oceanic Sciences department at the University of California Los Angeles. Yifan's background includes remote sensing of terrestrial biomass and regional climate modeling.

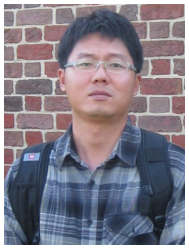
Sassan Saatchi is a senior researcher with the Radar Science & Engineering Section, with the UCLA Institute of the Environment, and with the UCLA Center for Tropical Research. He received his PhD in Electrophysics and Applied Mathematics from George Washington University in 1988 and BS and MS in Electrical Engineering from the University of Illinois at Chicago 1981 and 1983 resp. He has been a principal investigator on several interdisciplinary international projects including FIFE, EFEDA, Magellan, Mac-Hydro, Hapex-Sahel, BOREAS, LCLUC, and LBA, where he contributed by developing the remote sensing techniques for measuring regional scale ecological and hydrological variables. He has also participated and contributed in developing the concepts and design of several satellite sensors launched by NASA.

Robert DiBiano is a PhD candidate in the Computer and Electrical Engineering department at Louisiana State University, USA. His research interests include image analytics and machine learning for satellite imagery data.

Manohar Karki is a PhD candidate in the Computer Science Department at Louisiana State University, USA. He completed his BS in Computer Science from the University of Arlington, USA in 2010. His research interests include applications of Machine Learning and Computer Vision to image and video analytics.

Edward Boyda graduated with an A.B. in physics from Harvard University, M.A. and Ph.D. from the University of California Berkeley. He teaches on the physics faculty at Saint Mary's College of California. Trained as a particle physicist, he is currently interested in quantum computing and applications in computer vision.

Uttam Kumar is a Research Scientist with the Oak Ridge Associated Universities/NASA Ames Research Center, Moffett Field, CA, USA. His research interests include Spatial data mining, Environment Management, Remote sensing, Land use / Land cover change analysis, Statistical Methods, Spatio-temporal data analysis, Urbanisation and public policy, Geographical Information System and Management Information System.



Shuang Li received the M.S. degree from Henan University, Henan, China, in 2003, and the Ph.D. degree from The Graduate School of Chinese Academy of Sciences, Beijing, China, in 2006, respectively, both in GIS and environmental remote sensing. He has more than10 years of experience in geospatial analysis and remote sensing in the field of earth science. He is currently a research scientist in the EcoCast Lab at NASA Ames Research Center. He is also Adjunct Research Professor of California State University, Monterey Bay, USA. He works on Remote Sensing of terrestrial ecosystem, conducting Long Term Climate Change Trend Detection, Regional Drought detection, Soil properties mapping on arid and semi-arid area, and the vegetation recovery retrieval of post-fire ecosystem.