DeepSat – A Learning framework for Satellite Imagery

Saikat Basu*
Department of Computer Science, Louisiana State University
Baton Rouge
Louisiana 70803, USA
sbasu8@lsu.edu

Sangram Ganguly
Bay Area Environmental Research Institute/NASA Ames Research Center
Moffett Field
California, USA
sangram.ganguly@nasa.gov

Supratik Mukhopadhyay
Department of Computer Science, Louisiana State University
Baton Rouge
Louisiana 70803, USA
supratik@csc.lsu.edu

Robert DiBiano
Department of Computer Science, Louisiana State University
Baton Rouge
Louisiana 70803, USA
robertdibiano@gmail.com

Manohar Karki
Department of Computer Science, Louisiana State University
Baton Rouge
Louisiana 70803, USA
mkarki2@gmail.com

Ramakrishna Nemani
NASA Advanced Supercomputing Division/NASA Ames Research Center
Moffett Field, California, USA
rama.nemani@nasa.gov

ABSTRACT

Satellite image classification is a challenging problem that lies at the crossroads of remote sensing, computer vision, and machine learning. Due to the high variability inherent in satellite data, most of the current object classification approaches are not suitable for handling satellite datasets. The progress of satellite image analytics has also been inhibited by the lack of a single labeled high-resolution dataset with multiple class labels. The contributions of this paper are twofold – (1) first, we present two new satellite datasets called SAT-4 and SAT-6, and (2) then, we propose a classification framework that extracts features from an input image, normalizes them and feeds the normalized feature vectors to a Deep Belief Network for classification. On the SAT-4 dataset, our best network produces a classification accuracy of 97.95% and outperforms three state-of-the-art object recognition algorithms, namely - Deep Belief Networks, Convolutional Neural Networks and Stacked Denoising Autoencoders by $\sim$11%. On SAT-6, it produces a classification accuracy of 93.9% and outperforms the other algorithms by $\sim$15%. Comparative studies with a Random Forest classifier show the advantage of an unsupervised learning approach over traditional supervised learning techniques. A statistical analysis based on Distribution Separability Criterion and Intrinsic Dimensionality Estimation substantiates the effectiveness of our approach in learning better representations for satellite imagery.

*Corresponding Author

SIGSPATIAL '15 November 03-06, 2015, Bellevue, WA, USA

http://dx.doi.org/10.1145/2820783.2820816.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous; I.2.10 [**Vision and Scene Understanding**]: Texture; I.5.1 [**Pattern Recognition**]: Models—*Neural Nets*

Keywords

Satellite Imagery, Deep Learning, High Resolution

1. INTRODUCTION

Deep Learning has gained popularity over the last decade due to its ability to learn data representations in an unsupervised manner and generalize to unseen data samples using hierarchical representations. The most recent and best-known *Deep learning model* is the *Deep Belief Network* [15]. Over the last decade, numerous breakthroughs have been made in the field of Deep Learning; a notable one being [22], where a locally connected sparse autoencoder was used to detect objects in the ImageNet dataset [11] producing state-of-the-art results. In [27], Deep Belief Networks have been used for modeling acoustic signals and have been shown to outperform traditional approaches using Gaussian Mixture Models for Automatic Speech Recognition (ASR). They have also been found useful in hybrid learning models for noisy handwritten digit classification [2]. Another closely related approach, which has gained much traction over the last decade, is the Convolutional Neural Network [23]. This has been shown to outperform Deep Belief Network in classical object recognition tasks like MNIST [39], and CIFAR [20].

A related and equally hard problem is Satellite[1] image classification. It involves terabytes of data and significant variations due to conditions in data acquisition, pre-processing and filtering. Traditional supervised learning methods like Random Forests [6] do not generalize well for such a large-scale learning problem. A novel classification algorithm for detecting roads in Aerial imagery using Deep Neural Networks was proposed in [26]. The problem of detecting various land cover classes in general is a difficult problem

[1]Note that we use the terms satellite and airborne interchangeably in this paper because the extracted features and learning algorithms are generic enough to handle both satellite and airborne datasets.

considering the significantly higher intra-class variability in land cover types such as trees, grasslands, barren lands, water bodies, etc. as compared to that of roads. Also, in [26], the authors used a window of size 64×64 to derive contextual information. For our general classification problem, a 64×64 window is too big a context covering a total area of 64m×64m. A tree canopy, or a grassy patch can typically be much smaller than this area and hence we are constrained to use a contextual window having a maximum dimension of 28m×28m.

Traditional supervised learning approaches require carefully selected handcrafted features and substantial amounts of labeled data. On the other hand, purely unsupervised approaches are not able to learn the higher order dependencies inherent in the land cover classification problem. So, we propose a combination of handcrafted features that were first used in [14] and an unsupervised learning framework using Deep Belief Network [15] that can learn data representations from large amounts of unlabeled data.

There has been limited research in the field of satellite image classification due to a dearth of labeled satellite image datasets. The most well known labeled satellite dataset is the NLCD 2006 [38], which covers the entire globe and provide a spatial resolution of 30m. However, at this resolution, it becomes extremely difficult to distinguish between various landcover types. A high-resolution dataset acquired at a spatial resolution of 1.2m was used in [26]. However, the total area covered by the datasets namely URBAN1 and URBAN2 was ∼600 square kilometers, which included both training and testing datasets. The labeling was also available only for roads. Satellite/airborne image classification at a spatial resolution of 1-m was addressed in [1]. However, they performed tree-cover delineation by training a binary classifier based on Feedforward Backpropagation Neural Networks.

The main contributions of our work are twofold – (1) We first present two labeled datasets of airborne images – SAT-4 and SAT-6 covering a total area of ∼800 square kilometers, which can be used to further the research and investigate the use of various learning models for airborne image classification. Both SAT-4 and SAT-6 are sampled from a much larger dataset [40], which covers the whole of continental United States and can be used to create labeled landcover maps, which can then be used for various applications such as measuring ground carbon content or estimating total area of rooftops for solar power generation.

(2) Next, we present a framework for the classification of satellite/airborne imagery that a) extracts features from the image, b) normalizes the features, and c) feeds the normalized feature vectors to a Deep Belief Network for classification. On the SAT-4 dataset, our framework outperforms three state-of-the-art object recognition algorithms - Deep Belief Networks, Convolutional Neural Networks and Stacked Denoising Autoencoders by ∼11% and produces an accuracy of 97.95%. On SAT-6, it produces an accuracy of 93.9% and outperforms the other algorithms by ∼15%. We also present a statistical analysis based on Distribution Separability Criterion and Intrinsic Dimensionality Estimation to justify the effectiveness of our feature extraction approach to obtain better representations for satellite data.

2. DATASET[2]

Images were extracted from the National Agriculture Imagery Program (NAIP [40]) 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 and the area corresponds to the United States Geological Survey (USGS) topographic quadrangles. The average image tiles are ∼6000 pixels in width and ∼7000 pixels in height, measuring around 200 megabytes each. The entire NAIP dataset for CONUS is ∼65 terabytes. The imagery is acquired at a 1-m ground sample distance (GSD) with a horizontal accuracy that lies within six meters of photo-identifiable ground control points [41]. The images consist of 4 bands – red, green, blue and Near Infrared (NIR). In order to maintain the high variance inherent in the entire NAIP dataset, we sample image patches from a multitude of scenes (a total of 1500 image tiles) covering different landscapes like rural areas, urban areas, densely forested, mountainous terrain, small to large water bodies, agricultural areas, etc. covering the whole state of California. An image labeling tool developed as part of this study was used to manually label uniform image patches belonging to a particular landcover class. Once labeled, 28×28 non-overlapping sliding window blocks were extracted from the uniform image patch and saved to the dataset with the corresponding label. We chose 28×28 as the window size to maintain a significantly bigger context as pointed by [26], and at the same time not to make it as big as to drop the relative statistical properties of the target class conditional distributions within the contextual window. Care was taken to avoid interclass overlaps within a selected and labeled image patch. Sample images from the dataset are shown in Figure 1.

2.1 SAT-4

SAT-4 consists of a total of 500,000 image patches covering four broad land cover classes. These include – barren land, trees, grassland and a class that consists of all land cover classes other than the above three. 400,000 patches (comprising of four-fifths of the total dataset) were chosen for training and the remaining 100,000 (one-fifths) were chosen as the testing dataset. We ensured that the training and test datasets belong to disjoint set of image tiles. Each image patch is size normalized to 28×28 pixels. Once generated, both the training and testing datasets were randomized using a pseudo-random number generator.

2.2 SAT-6

SAT-6 consists of a total of 405,000 image patches each of size 28×28 and covering 6 landcover classes - barren land, trees, grassland, roads, buildings and water bodies. 324,000 images (comprising of four-fifths of the total dataset) were chosen as the training dataset and 81,000 (one fifths) were chosen as the testing dataset. Similar to SAT-4, the training and test sets were selected from disjoint NAIP tiles. Once generated, the images in the dataset were randomized in the same way as that for SAT-4. The specifications for the various landcover classes of SAT-4 and SAT-6 were adopted from those used in the National Land Cover Data (NLCD) algorithm [43].

3. INVESTIGATION OF VARIOUS DEEP LEARNING MODELS

3.1 Deep Belief Network

Deep Belief Network (DBN) consists of multiple layers of stochastic, latent variables trained using an unsupervised learning algorithm followed by a supervised learning phase using feedforward backpropagation Neural Networks. In the unsupervised pre-training stage, each layer is trained using a Restricted Boltzmann Machine (RBM). Unsupervised pre-training is an important step in solving a classification problem with terabytes of data and high variability. A

[2]THE SAT-4 AND SAT-6 DATASETS ARE AVAILABLE AT THE WEB LINK [42]

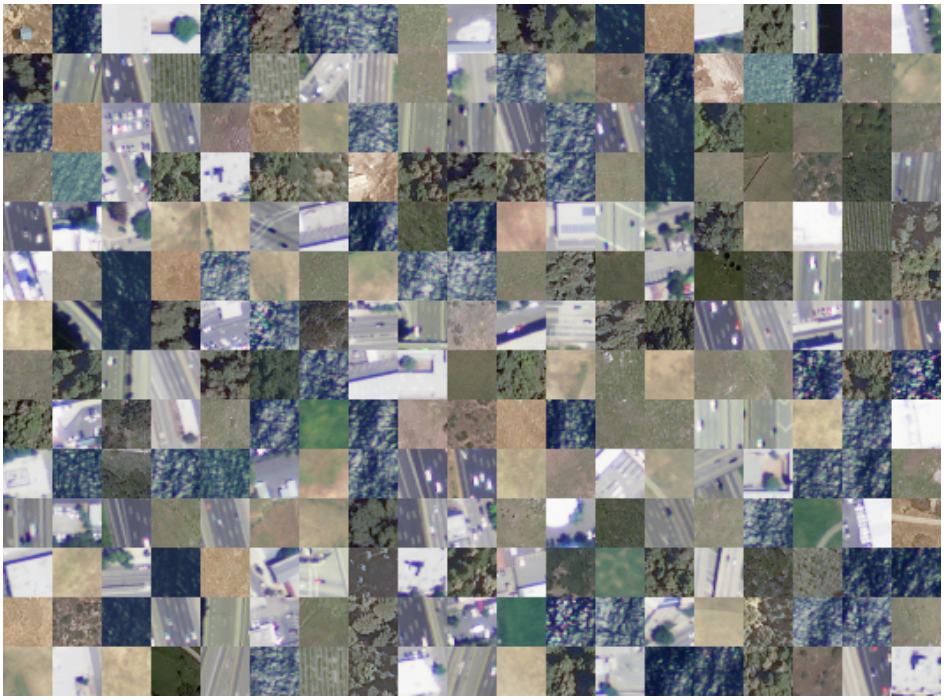


Figure 1: Sample images from the SAT-6 dataset

DBN is a graphical model [19] where neurons of the hidden layer are conditionally independent of each other given a particular configuration of the visible layer and vice versa. A DBN can be trained layer-wise by iteratively maximizing the conditional probability of the input vectors or visible vectors given the hidden vectors and a particular set of layer weights. As shown in [15], this layer-wise training can help in improving the variational lower bound on the probability of the input training data, which in turn leads to an improvement of the overall generative model.

We first provide a formal introduction to the Restricted Boltzmann Machine. The RBM can be denoted by the energy function:

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} v_i \quad (1)$$

where, the RBM consists of a matrix of layer weights $W = (w_{i,j})$ between the hidden units h_j and the visible units v_i. The a_i and b_j are the bias weights for the visible units and the hidden units respectively. The RBM takes the structure of a bipartite graph and hence it only has inter-layer connections between the hidden or visible layer neurons but no intra-layer connections within the hidden or visible layers. So, the visible unit activations are mutually independent given a particular set of hidden unit activations and vice versa [7]. Hence, by setting either h or v constant, we can compute the conditional distribution of the other as follows:

$$P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{m} w_{i,j} v_i) \quad (2)$$

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} w_{i,j} h_j) \quad (3)$$

where, σ denotes the log sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \quad (4)$$

The training algorithm maximizes the expected log probability assigned to the training dataset V. So if the training dataset V consists of the visible vectors v, then the objective function is as follows:

$$\underset{W}{\operatorname{argmax}}\, E\Big[\sum_{v \in V} \log P(v)\Big] \quad (5)$$

A RBM is trained using a *Contrastive Divergence* algorithm [7]. Once trained, the DBN can be used to initialize the weights of the Neural Network for the supervised learning phase [3].

Next, we investigate the classification accuracy of various architectures of DBN on both SAT-4 and SAT-6 datasets.

3.1.1 DBN Results on SAT-4 & SAT-6

To investigate the performance of the DBN, we experiment with both *big* and *deep* neural architectures. This is done by varying the number of neurons per layer as well as the total number of layers in the network. Our objective is to investigate whether the more complex features learned in the deeper layers of the DBN are able to provide the network with the discriminative power required to handle higher-order texture features typical of satellite imagery data. The results from the DBN for various network architectures for SAT-4 and SAT-6 are enumerated in Table 1. Each network was trained for a maximum of 500 epochs and the network state with the lowest validation error was used for testing. Regularization is done using L_2 norm-regularization. It can be seen from the table that for both SAT-4 and SAT-6, the classifier accuracy initially improves and then falls as more neurons or layers are added to the network.

3.2 Convolutional Neural Network

Network Arch. Neurons/layer [Layers]	Classifier Accuracy SAT-4 (%)	Classifier Accuracy SAT-6 (%)
100 [2]	79.74	68.51
100 [3]	**81.78**	**76.47**
100 [4]	79.802	74.44
100 [5]	62.776	63.14
500 [2]	68.916	60.35
500 [3]	71.674	61.12
500 [4]	65.002	57.31
500 [5]	64.174	55.78

Table 1: Classification Accuracy of DBN with various architectures on SAT-4 and SAT-6

Convolutional Neural Network (CNN) first introduced in [13] is a hierarchical model inspired by the human visual cortical system [16]. It was significantly improved and applied to document recognition in [23]. A committee of 35 convolutional neural nets with elastic distortions and width normalization [9] has produced state-of-the-art results on the MNIST handwritten digits dataset. CNN consists of a hierarchical representation using convolutional layers and fully connected layers, with non-linear transformations and feature pooling.

They also include local or global pooling layers. Pooling can be implemented in the form of subsampling, averaging, max-pooling or stochastic pooling. Each of these pooling architectures has its own advantages and limitations and numerous studies are in place that investigate the effect of different pooling functions on representation power of the model ([31],[30]). A very important feature of Convolutional Neural Network is weight sharing in the convolutional layers, which means that the same filter bank can be used for all pixels in a particular layer; thereby generating sparse networks that can generalize well to unseen data samples while maintaining the representational power inherent in deep hierarchical architectures.

We investigate the use of different CNN architectures for SAT-4 and SAT-6 as detailed below.

3.2.1 CNN Results on SAT-4 & SAT-6

For CNN, we vary the number of feature maps in each layer as well as the total number of convolutional and subsampling layers. The results from various network configurations with increasing number of maps and layers is enumerated in Table 2. For the experiments, we used both 3×3 and 5×5 kernels for the convolutional layers and 3×3 averaging and max-pooling kernels for the sub-sampling layers. We also use overlapping pooling windows with a stride size of 2 pixels. The last sub-sampling layer is connected to a fully-connected layer with 64 neurons. The output of the fully-connected layer is fed into a 4-way softmax function that generates a probability distribution over the 4 class labels of SAT-4 and a 6-way softmax for the 6 class labels of SAT-6. In Table 2, the "Ac-Bs(n)" notation denotes that the network has a convolutional layer with A feature maps followed by a sub-sampling layer with a kernel of size $B\times B$. 'n' denotes the type of pooling function in the sub-sampling layer, 'a' denotes average pooling while 'm' denotes max-pooling. From the table, it can be seen that the smallest networks consistently produce the best results. Also, both for SAT-4 and SAT-6, using networks with convolution kernels of size 3×3 leads to a significant drop in classifier accuracy. The biggest networks with 50 maps per layer also exhibit significant drop in classifier accuracy.

Network Architecture (Convolution kernel size)	Accuracy SAT-4 (%)	Accuracy SAT-6 (%)
6c-3s(a)-12c-3s(m) (5×5)	**86.827**	**79.063**
18c-3s(a)-36c-3s(m) (5×5)	82.325	78.704
6c-3s(a)-12c-3s(m)-12c -3s(m)(5×5)	81.907	76.963
50c-3s(a)-50c-3s(m)-50c -3s(m)(5×5)	73.85	75.689
6c-3s(a)-12c-3s(m) (3×3)	73.811	54.385
6c-3s(m)-12c-3s(m) (5×5)	85.612	77.636

Table 2: Classification Accuracy of CNN with various architectures on SAT-4

3.3 Stacked Denoising Autoencoder

A Stacked Denoising Autoencoder (SDAE) [37] consists of a combination of multiple sparse autoencoders, which can be trained in a greedy-layerwise fashion similar to that of Restricted Boltzmann Machines in a DBN. Each autoencoder is associated with a set of weights and biases. In the SDAE, each layer can be trained independent of the other layers. Once trained, the parameters of an autoencoder are frozen in place. The training algorithm consists of two passes – a forward pass and a backward pass. The forward pass, also called as the encoding phase encodes raw image pixels into an increasingly higher-order representation. The backward pass simply performs the reverse operation by decoding these higher-order features into simpler representations. The encoding step is given as:

$$a^{(l)} = f(z^{(l)}) \tag{6}$$

$$z^{(l+1)} = W^{(l,1)}a^{(l)} + b^{(l,1)} \tag{7}$$

And the decoding step is as follows:

$$a^{(n+l)} = f(z^{(n+l)}) \tag{8}$$

$$z^{(n+l+1)} = W^{(n-l,2)}a^{(n+l)} + b^{(n-l,2)} \tag{9}$$

The hidden unit activations of the neurons in the deepest layer are used for classification after a supervised fine-tuning using back-propagation.

3.3.1 SDAE Results on SAT-4 & SAT-6

Different network configurations were chosen for the SDAE in a manner similar to that described above for DBN and CNN. The results are enumerated in Table 3. Similar to DBN, each network is trained for a maximum of 500 epochs and the lowest test error is considered for evaluation. As highlighted in the Table, networks with 5 layers and 100 neurons in each layer produce the best results on both SAT-4 and SAT-6. It can be seen from the table that on both datasets, the classifier accuracy initially improves and then drops with increasing number of neurons and layers, similar to that of DBN. Also, the biggest networks with 500 and 2352 neurons in each layer exhibit a significant drop in classifier accuracy.

4. DEEPSAT - A DETAILED ARCHITECTURAL OVERVIEW

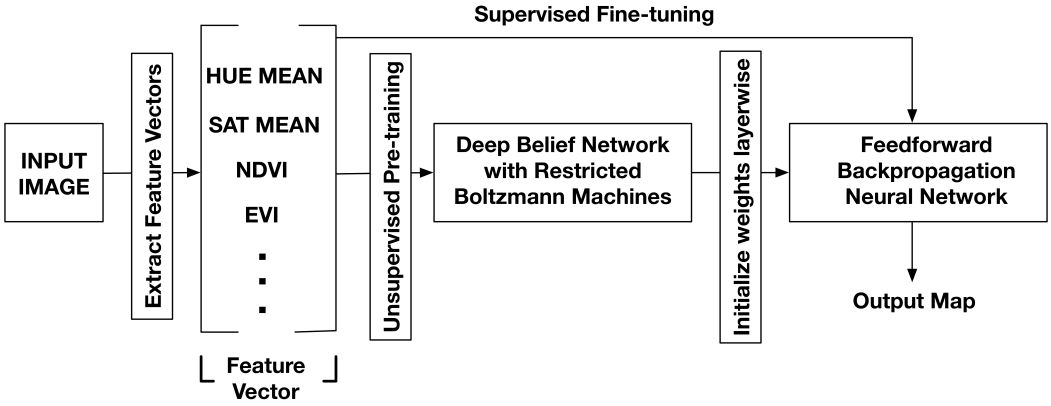


Figure 2: Schematic of the DeepSat classification framework

Network Arch. Neurons/layer [Layers]	Classifier Accuracy SAT-4 (%)	Classifier Accuracy SAT-6 (%)
100 [1]	75.88	74.89
100 [2]	76.854	76.12
100 [3]	77.804	76.45
100 [4]	78.674	76.52
100 [5]	**79.978**	**78.43**
100 [6]	75.766	76.72
500 [3]	63.832	54.37
2352 [2]	51.766	37.121

Table 3: Classification Accuracy of SDAE with various architectures on SAT-4 and SAT-6

Figure 2 schematically describes our proposed classification framework. Instead of the traditional DBN model described in Section 3.1, which takes as input the multi-channel image pixels reshaped as a linear vector, our classification framework first extracts features from the image which in turn are fed as input to the DBN after normalizing the feature vectors.

4.1 Feature Extraction

The feature extraction phase computes 150 features from the input imagery. The key features that we use for classification 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 ([14], [32], [10]). Since two of the classes in SAT-4 and SAT-6 are trees and grasslands, we incorporate features that are useful determinants for segregation of vegetated areas from non-vegetated ones. The red band already provides a useful feature for discrimination of vegetated and non-vegetated areas based on chlorophyll reflectance, however, we also use derived features (vegetation indices derived from spectral band combinations) that are more representative of vegetation greenness – this includes the Enhanced Vegetation Index (EVI [17]), Normalized Difference Vegetation Index (NDVI [29], [35]) and Atmospherically Resistant Vegetation Index (ARVI [18]).

These indices are expressed as follows:

$$EVI = G \times \frac{NIR - Red}{NIR + c_{red} \times Red - c_{blue} \times Blue + L} \quad (10)$$

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 [41].

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (11)$$

$$ARVI = \frac{NIR - (2 \times Red - Blue)}{NIR + (2 \times Red + Blue)} \quad (12)$$

The performance of our learner 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 [5]. Details of the feature ranking method along with the ranking for all the 22 features used in our framework is listed in Section 6.1.1.

4.2 Data Normalization

The feature vectors extracted from the training and test datasets are separately normalized to lie in the range $[0, 1]$. This is done using the following equation:

$$F_{normalized} = \frac{F - F_{min}}{F_{max} - F_{min}} \quad (13)$$

where, F_{min} and F_{max} are computed for a particular feature type over all images in the dataset.

4.3 Classification

The set of normalized feature descriptors extracted from the input image is fed into the DBN, which is then trained using *Contrastive divergence* in the same way as explained in Section 3.1. Once trained the DBN is used to initialize the weights of a feedforward backpropagation neural network.

The neural network gives an estimate of the posterior probabilities of the class labels, given the input vectors, which is the feature vector in our case. As illustrated in [4], the outputs of a neural network trained by minimizing the sum of squares error function approximates the conditional averages of the target data

$$y_k(x) = \langle t_k|x\rangle = \int t_k p(t_k|x)dt_k \tag{14}$$

Here, t_k are the set of target values that represent the class membership of the input vector x_k. For a binary classification problem, in order 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 would then be given as

$$p(t_k|x) = \delta(t-1)P(C_1|x) + \delta(t)P(C_2|x) \tag{15}$$

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

$$\int_{-\infty}^{\infty} \delta(x)\,\mathrm{d}x = 1 \tag{16}$$

From 14 and 15, we get

$$y(x) = P(C_1|x) \tag{17}$$

So, 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 $P(C_2|x) = 1 - y(x)$. This argument can easily be extended to multiple class labels for a generalized multi-class classification problem.

The feature extraction phase proves to be a useful dimensionality reduction technique that helps improve the discriminative power of the DBN based classifier significantly.

5. RESULTS AND COMPARATIVE STUDIES

The feature vectors extracted from the dataset are fed into DBNs with different configurations. Since, the feature vectors create a low dimensional representation of the data, so, DeepSat converges to high accuracy even with a much smaller network with fewer layers and very few neurons per layer. This speeds up network training by several orders of magnitude. Various network architectures along with the classification accuracy for DeepSat on the SAT-4 and SAT-6 datasets are listed in Table 4. For regularization, we again use L_2 norm-regularization. From the Table, it is evident that the best performing DeepSat network outperforms the best traditional Deep Learning approach (CNN) by $\sim$11% on the SAT-4 dataset and by $\sim$15% on the SAT-6 dataset.

We also compare DeepSat with a Random Forest classifier to investigate the advantages gained by unsupervised pre-training in DBN as opposed to the traditional supervised learning in Random Forests. On SAT-4, the Random forest classifier produces an accuracy of 69% while on SAT-6, it produces an accuracy of 54%. The highest accuracy was obtained for a forest with 100 trees. Further increase in the number of trees did not yield any significant improvement in classifier accuracy. It can be easily seen that the various Deep architectures produce better classification accuracy than the Random Forest classifier which relies solely on supervised learning.

6. WHY TRADITIONAL DEEP ARCHITECTURES ARE NOT ENOUGH FOR SAT-4 & SAT-6?

Network Arch. Neurons/layer [Layers]	Classifier Accuracy SAT-4 (%)	Classifier Accuracy SAT-6 (%)
10 [2]	96.585	91.91
10 [3]	96.8	87.716
20 [2]	97.115	86.21
20 [3]	95.473	93.42
50 [2]	**97.946**	**93.916**
50 [3]	97.654	92.65
100 [2]	97.292	89.08
100 [3]	95.609	91.057

Table 4: Classification Accuracy of DeepSat with various network architectures on SAT-4 and SAT-6

While traditional Deep Learning approaches have produced state-of-the-art results for various pattern recognition problems like handwritten digit recognition [39], object recognition [20], face recognition [33], etc., but satellite datasets have high intra and inter-class variability and the amount of labeled data is much smaller as compared to the total size of the dataset. Also, higher-order texture features are a very important discriminative parameter for various landcover classes. On the contrary, shape/edge based features which are predominantly learned by various Deep architectures are not very useful in learning data representations for satellite imagery. This explains the fact why traditional Deep architectures are not able to converge to the global optima even for reasonably large as well as Deep architectures.

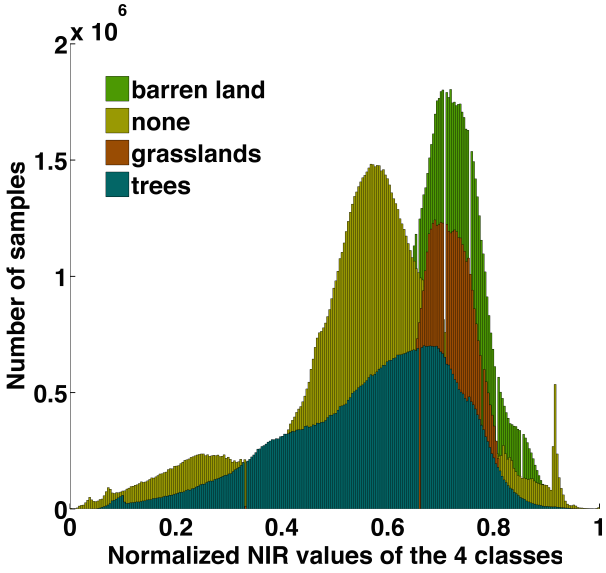
Also, spatially contextual information is another important parameter for modeling satellite imagery. In traditional Deep Learning approaches like DBN and SDAE, the relative spatial information of the pixels is lost. As a result the orderless pool of pixel values which acts as input to the Deep Networks lack sufficient discriminative power to be well-represented even by very big and/or deep networks. CNN however, involves feature-pooling from a local spatial neighborhood, which justifies its improved performance over the other two algorithms on both SAT-4 and SAT-6. Even though our approach extracts an orderless pool of feature vectors, the spatial context is already well-represented in the individual feature values themselves. We substantiate our arguments about the effectiveness of our feature extraction approach from a statistical point of view as detailed in the analysis below.

		Dist. b/w Means	Standard Deviations
SAT-4	Raw Images	0.1994	0.1166
	DeepSat Features	**0.8454**	**0.0435**
SAT-6	Raw Images	0.3247	0.1273
	DeepSat Features	**0.9726**	**0.0491**

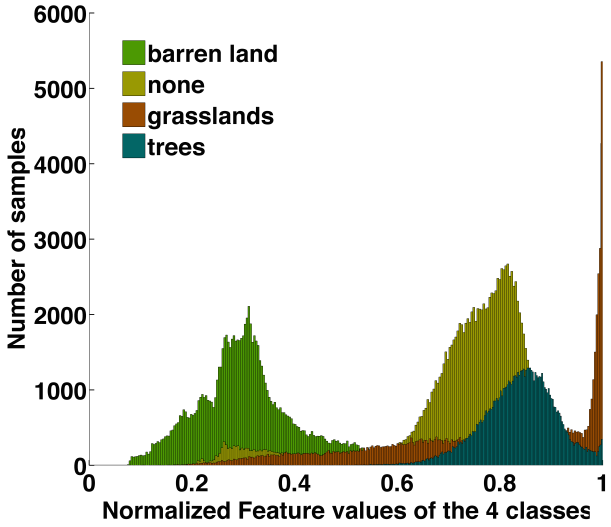
Table 5: Distance between Means and Standard Deviations for raw image values and DeepSat feature vectors for SAT-4 and SAT-6

6.1 A Statistical Perspective based on Distribution Separability Criterion

Improving classification accuracy can be viewed as maximizing the separability between the class-conditional distributions. Following the analysis presented in [5], we can view the problem of maximizing distribution separability as maximizing the distance between distribution means and minimizing their standard deviations. Figure 3 shows the histograms that represent the class-



(a) Distribution of NIR on the SAT-4 classes



(b) Distribution of a sample DeepSat feature (Autocorrelation of Hue Color co-occurance matrix) on the SAT-4 classes

Figure 3: Distributions of the raw NIR values for traditional Deep Learning Algorithms and a sample DeepSat feature for various classes on SAT-4 (*Best viewed in color*)

conditional distributions of the NIR channel and a sample feature extracted in the DeepSat framework. As illustrated in Table 5, the features extracted in DeepSat have a higher distance between means and a lower standard deviation as compared to the original image distributions, thereby ensuring better class separability.

6.1.1 *Feature Ranking*

Following the analysis proposed in Section 6.1 above, we can derive a metric for the Distribution Separability Criterion as follows:

$$D_s = \frac{\overline{\|\delta_{mean}\|}}{\overline{\delta_\sigma}} \tag{18}$$

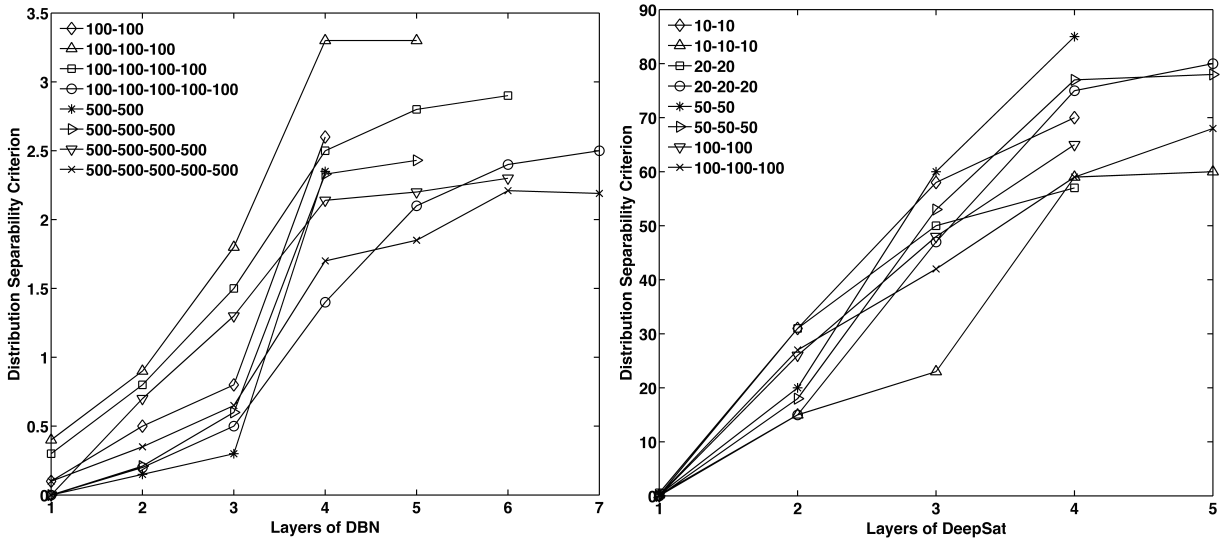
where $\overline{\|\delta_{mean}\|}$ indicates the mean of distance between means and $\overline{\delta_\sigma}$ indicates the mean of standard deviations of the class conditional distributions. Maximizing D_s over the feature space, a feature ranking can be obtained. Table 6 shows the ranking of the 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.

6.1.2 *Distribution Separability and Classifier Accuracy*

In order to analyze the improvements achieved in the learning framework due to the feature extraction step, we measured the Distribution Separability of the mean activation of the neurons in each layer of the DBN and that of DeepSat. The results are noted in Figure 4. It can be seen that the mean activation learned by each layer of DeepSat exhibit a significantly higher distribution separability (by several orders of magnitude) than the neurons of a DBN. This justifies the significant improvement in performance of DeepSat (using the features) as compared to the DBN based framework (using the raw pixel values as input). Also, a comparison of Figure 4 with Table 1 and Table 4 shows that the distribution separabilities using the various architectures of the DBN and DeepSat are positively correlated to the final classifier accuracy. This justifies the effectiveness of our distribution separability metric D_s as a measure of the final classifier accuracy.

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 6: Ranking of features based on Distribution Separability Criterion for SAT-6



(a) Distribution Separability Criterion of DBN

(b) Distribution Separability Criterion of DeepSat

Figure 4: Distribution Separability Criterion of the neurons in the layers of a DBN and DeepSat with various architectures on SAT-6

7. WHAT IS THE DIFFERENCE BETWEEN MNIST, CIFAR-10 AND SAT-6 IN TERMS OF DIMENSIONALITY?

We argue that handwritten digit datasets like MNIST and object recognition datasets like CIFAR-10 lie on a much lower dimensional manifold than the airborne SAT-6 dataset. Hence, even if Deep Neural Networks can effectively classify the raw feature space of object recognition datasets but the dimensionality of the airborne image datasets is such that Deep Neural Networks cannot classify them. In order to estimate the dimensionality of the datasets, we use the concept of *intrinsic dimension*[8].

7.1 Intrinsic Dimension Estimation using the DanCo algorithm

To estimate the intrinsic dimension of a dataset, we use the DANCo algorithm [8]. It exploits the twofold complementary information conveyed both by the normalized nearest neighbor distances and by the angles computed on couples of neighboring points.

Taking 10 rounds of 1000 random samples and averaging, we obtain the intrinsic dimension for the MNIST, CIFAR-10 and SAT-6 datasets and the Haralick features extracted from the SAT-6 dataset. The results are listed in Table 7.

Dataset	Intrinsic Dimension
MNIST	16
CIFAR-10	17
SAT-6	115
Haralick Features extracted from SAT-6	**4.2**

Table 7: Intrinsic Dimension estimation using DANCo on the MNIST, CIFAR-10, and SAT-6 datasets and the Haralick features extracted from the SAT-6 dataset.

So, it can be seen that the intrinsic dimensionality of the SAT-6 dataset is orders of magnitude higher than that of MNIST. So, a deep neural network finds it difficult to classify the SAT-6 dataset because of its intrinsically high dimensionality. However, as seen in the equation above, the features extracted from SAT-6 have a much lower intrinsic dimensionality and lie on a much lower dimensional manifold than the raw vectors and hence can be classified even by networks with relatively smaller architectures.

7.2 Visualizing Data in an n-dimensional space

We can visualize the data as distributed in an n-dimensional unit hypersphere
Volume of the sphere,

$$V_{sphere} = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} R^n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} \tag{19}$$

for n-dimensional Euclidean space and Γ is Euler's gamma function. Now, the total volume of the n-dimensional space can be accounted by the volume of an n-dimensional hypercube of length 2 embedding the hypersphere, i.e, Volume of the n-cube,

$$V_{cube} = R^n = 2^n \tag{20}$$

So, the relative fraction of the data points which lie on the sphere as compared to the data points on the n-dimensional embedding space is given as

$$V_{relative} = \frac{V_{sphere}}{V_{cube}} = \frac{\pi^{\frac{n}{2}}}{2^n \Gamma(\frac{n}{2}+1)} \tag{21}$$

$$V_{relative} \to 0 \text{ as } n \to \infty \tag{22}$$

This means that as the dimensionality of sample data approaches ∞, the spread or scatter of the data points approaches 0 with respect to the total search space. As a result, various classification and clustering algorithms lose their discriminative power in higher dimensional feature spaces.

8. RELATED WORK

Present classification algorithms used for Moderate-resolution Imaging Spectroradiometer (MODIS)(500-m) [12] or Landsat(30-m) based land cover maps like NLCD [38] produce accuracies of 75% and 78% resp. The relatively lower resolution of the datasets makes it difficult to analyze the performance of these algorithms

for 1-m imagery. A method based on object detection using Bayes framework and subsequent clustering of the objects using Latent Dirichlet Allocation was proposed in [36]. However, their approach detects object groups at a higher level of abstraction like parking lots. Detecting the objects like cars or trees in itself is not addressed in their work. A deep convolutional hierarchical framework was proposed recently by [28]. However, they report results on the AVIRIS Indiana's Indian Pines test site. The spatial resolution of the dataset is limited to 20m and it is difficult to evaluate the performance of their algorithm for object recognition tasks at a higher resolution. An evaluation of various feature learning strategies was done in [34]. They evaluated both feature extraction techniques as well as classifiers like DBN and Random Forest for various aerial datasets. However, since the training data was significantly limited, the DBN was not able to produce any improvements over Random Forest even when raw pixel values were fed into the classifier. In contrast, our study shows that DBNs can be better classifiers when there is significant amount of training data to initialize the neural network at a global error basin.

9. CONCLUSIONS AND FUTURE DIRECTIONS

Our semi-supervised learning framework produces an accuracy of 97.95% and 93.9% on the SAT-4 and SAT-6 datasets and significantly outperforms the state-of-the-art by $\sim$11% and $\sim$15% respectively. The Feature extraction phase is inspired by the remote sensing literature and significantly improves the discriminative power of the framework. For satellite datasets, with inherently high variability, traditional deep learning approaches are unable to converge to a global optima even with significantly big and deep architectures. A statistical analysis based on Distribution Separability Criterion justifies the effectiveness of our feature extraction approach.

We plan to investigate the use of various pooling techniques like SPM [21] as well as certain sparse representations like sparse coding [24] and Hierarchical representations like Convolutional DBN [25] to handle satellite datasets. We believe that SAT-4 and SAT-6 will enable researchers to learn better representations for satellite datasets and create benchmarks for the classification of satellite imagery.

10. ACKNOWLEDGMENTS

The project is supported by NASA Carbon Monitoring System through Grant #NNH14ZDA001-N-CMS and Army Research Office (ARO) under Grant #W911NF1010495. 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-NNX12AD05A, 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 that of NASA, ARO or the United States Government.

11. REFERENCES

[1] S. Basu, S. Ganguly, R. Nemani, S. Mukhopadhyay, G. Zhang, C. Milesi, A. Michaelis, P. Votava, R. Dubayah, L. Duncanson, B. Cook, Y. Yu, S. Saatchi, R. DiBiano, M. Karki, E. Boyda, U. Kumar, and S. Li. A semiautomated probabilistic framework for tree-cover delineation from 1-m naip imagery using a high-performance computing architecture. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(10):5690–5708, Oct 2015.

[2] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, and R. Nemani. Learning sparse feature representations using probabilistic quadtrees and deep belief nets. In *Proceedings of the European Symposium on Artificial Neural Networks, ESANN*, 2015.

[3] Y. Bengio. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009.

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

[5] 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.

[6] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[7] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. 2005.

[8] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli. Danco: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern Recognition*, 47(8):2569 – 2581, 2014.

[9] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3642–3649, Washington, DC, USA, 2012. IEEE Computer Society.

[10] D. A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Can. J. Remote Sensing*, 28(1):45–62, 2002.

[11] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[12] 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*, 114:168–182, 2009.

[13] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621, Nov. 1973.

[15] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[16] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology (London)*, 160:106–154, 1962.

[17] A. Huete, K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote Sensing of Environment*, 83(1-2):195–213, Nov. 2002.

[18] Y. Kaufman and D. Tanre. Atmospherically resistant vegetation index (arvi) for eos-modis. *Geoscience and Remote Sensing, IEEE Transactions on*, 30(2):261–270, Mar 1992.

[19] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[20] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[22] Q. V. Le, M. Ranzato, R. Monga, M. Devin, G. Corrado, K. Chen, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[24] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

[25] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[26] 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.

[27] A.-r. Mohamed, G. E. Dahl, and G. E. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech & Language Processing*, 20(1):14–22, 2012.

[28] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction of hyperspectral images. 2014.

[29] 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*, pages 309–317, 1974.

[30] A. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Ng. On random weights and unsupervised feature learning. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1089–1096, New York, NY, USA, June 2011. ACM.

[31] D. Scherer, A. MÃijller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In K. Diamantaras, W. Duch, and L. Iliadis, editors, *Artificial Neural Networks - ICANN 2010*, volume 6354 of *Lecture Notes in Computer Science*, pages 92–101. Springer Berlin Heidelberg, 2010.

[32] L. K. Soh and C. Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *Geoscience and Remote Sensing, IEEE Transactions on*, pages 780–795, 1999.

[33] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[34] P. Tokarczyk, J. Montoya, and K. Schindler. An evaluation of feature learning methods for high resolution image classification. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:389–394, 2012.

[35] C. J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127 – 150, 1979.

[36] 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*, pages 2506–2510, Aug 2012.

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec. 2010.

[38] 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*, 130:294–304, 2013.

[39] WWW1. MNIST. `http://yann.lecun.com/exdb/mnist/`.

[40] WWW2. NAIP. `http://www.fsa.usda.gov/Internet/FSA_File/naip_2009_info_final.pdf`.

[41] WWW3. MODIS. `http://vip.arizona.edu/documents/MODIS/MODIS_VI_UsersGuide_01_2012.pdf`.

[42] WWW4. DATASETS. `http://csc.lsu.edu/~saikat/deepsat/`.

[43] WWW5. NLCD. `http://www.gsd.harvard.edu/gis/manual/earthshelter/National%20Land-Cover%20Dataset%20%28NLCD%29%20Metadata%20%20US%20EPA.htm`.