A Deep Learning Approach to Assessing Urban Tree Canopy Using Satellite Imagery in the City of Atlanta

Nilaksh Das
School of Computational Science and Engineering
Georgia Institute of Technology
nilakshdas@gatech.edu

Bistra Dilkina
School of Computational Science and Engineering
Georgia Institute of Technology
bdilkina@cc.gatech.edu

Abstract

Classification of satellite imagery to determine land cover type is a challenging task, primarily owing to the high intra-class variability in the classes of land cover under consideration. In this paper, we explore different pre-trained deep neural networks for this purpose. By fine-tuning and adapting the models to train on a labeled satellite imagery dataset, we find that these redesigned deep architectures outperform other proposed methods which involve complex representations of the imagery, simply by training on raw images without any kind of elaborate transformations. We finally determine the best model thus obtained, and use it to then quantify the urban tree canopy coverage in the city of Atlanta.

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 [1]. This ability gives way to leveraging the "knowledge" gained by a deep architecture in one domain in order to establish an intrinsic representation that is universal across other similar domains.

In this paper, we try to substantiate this assertion by adopting deep convolutional neural networks that have been trained to identify different taxonomies of objects and scenes from everyday life, and repurpose them to predict urban tree canopy coverage from satellite imagery.

Under the hood, deep architectures are based on convolutional neural networks, which are very similar to ordinary neural networks: they are made up of neurons that have learnable weights and biases. However, convolutional neural networks make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture [4].

Traditional neural networks receive an input (a single vector), and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the output

layer and in classification settings it represents the class scores. However, regular neural networks dont scale well to full images. The full connectivity in neural networks is wasteful and the huge number of parameters quickly lead to overfitting. On the other hand, convolutional neural networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way, which we will see in the next section.

Our aim in this paper is to leverage this sensibility, coupled with prior knowledge of high level representations, to accomplish the task of classifying satellite imagery.

2 Related Work

Deep learning has recently proven to break many barriers in the field of machine learning, especially in the domain of image recognition. In this regard, it is also an upcoming tool in the domain of satellite imagery classification.

Castelluccio et al. in [3] show promising results for using convolutional neural networks to determine land use type with remote sensing images. They demonstrate how fine-tuning a pre-trained deep convolutional neural network may lead to better classification accuracies on the UC Merced Land Use dataset and the Brazilian Coffee Scenes dataset.

Basu et al. in [1] further show how deep architectures could be coupled with newly derived features such as Enhanced Vegetation Index (EVI), Normalized Difference Vegetation Index (NDVI) and Atmospherically Resistant Vegetation Index (ARVI) to segregate even highly correlated classes such as trees and grasslands. These indices are determined from satellite imagery RGB and near-IR (NIR) spectra as follows:

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

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.

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

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

They propose a novel framework based on deep learning called *DeepSat* for classifying land cover types from satellite imagery. They also make available labelled satellite imagery datasets, SAT-4 and SAT-6, for researchers working in the domain of classification of satellite imagery.

Works such as these lend greatly to the utility of deep learning in the domain of classification of remote sensing data. Deep learning architectures such as those used in the above cited works are greatly vested in the efficient functionality of convolutional neural networks. Although this concept has been around for a long time [5], only recent technological progress, especially in the field of Graphical Processing Units (GPUs), has made these highly viable for learning. Typically, convolutional neural network architectures comprise of the following types of components [3]:

1. *Convolutional layers*: they compute the convolution of the input image with the weights of the network. Neurons in the first hidden layer view only a small image window, and learn low-level features. Those in deeper layers view (indirectly) larger portions of the image, and are able to learn more expressive features by combining low-level ones. Each layer is characterized by a few hyper-parameters: the number of filters to learn, their spatial support, the stride between different windows and an optional zero-padding which controls the size of the layer output.
2. *Pooling layers*: reduce the size of the input layer through some local non-linear operations, for example max(), so as to reduce the number of parameters to learn and provide some translation invariance. The most relevant hyper-parameters are the support of the pooling window and the stride between different windows.

3. *Normalization layers*: inspired by inhibition schemes present in the real neurons of the brain, aim at improving generalization.

4. *Fully-connected layers*: are typically used as the last few layers of the network. By removing constraints, they can better summarize the information conveyed by lower-level layers in view of the final decision. Despite full connectivity, their complexity is still affordable thanks to the previous size-reducing layers.

3 Data

3.1 SAT-6

We use the SAT-6 dataset [1] to train and evaluate the pre-trained deep convolutional neural networks. It consists of a total of 405,000 image patches each of size 28×28 and covering 6 land cover classes as shown in Table 1. Images were extracted from the National Agriculture Imagery Program (NAIP [8]) dataset. The NAIP dataset contains a total of 330,000 scenes spanning the whole of the Continental United States (CONUS). SAT-6 uses the uncompressed digital Ortho quarter quad tiles (DOQQs) which are GeoTIFF images the area of which 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 ground sample distance (GSD) of 1 meter. The horizontal accuracy lies within 6 meters of ground control points identifiable from the acquired imagery. 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, the image patches in SAT-6 are sampled from a multitude of scenes (a total of 1500 image tiles) covering different landscapes like rural areas, urban areas, densely forested regions, mountainous terrain, small to large water bodies, agricultural areas, etc. covering the whole state of California. An image labeling tool was used to manually label uniform image patches belonging to a particular landcover class to create the labelled SAT-6 dataset.

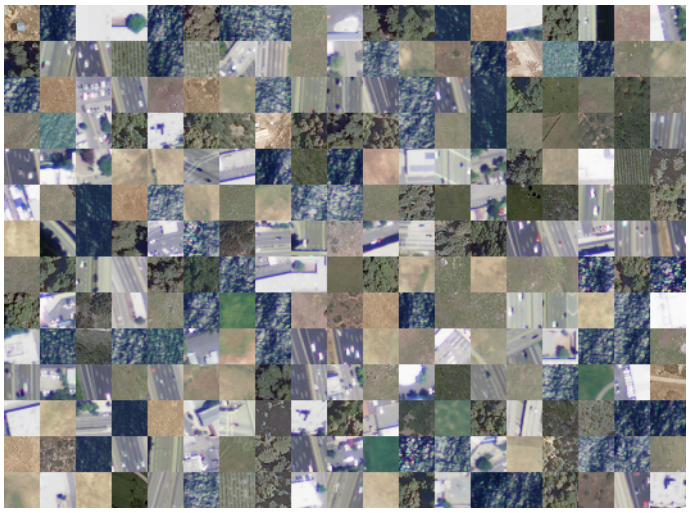

Figure 1: Sample images from the SAT-6 dataset

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. 324,000 images (comprising of four-fifth of the total dataset) were chosen as the training dataset and 81,000 (one-fifth) were chosen as the testing dataset. the training and test sets were selected from disjoint NAIP tiles. Once generated, the images in the dataset were randomized. The specifications for the various landcover classes of SAT-6 were adopted from those used in National Land Cover Data (NLCD) [9].

Table 1: Land cover classes covered by SAT-6 dataset with their corresponding labels

Label	Class	no. of samples (train)	no. of samples (test)
0	barren land	14,923	3,714
1	grassland	73,397	18,367
2	trees	56,809	14,185
3	roads	50,347	12,596
4	buildings	8,192	2,070
5	water bodies	120,332	30,068

3.2 WorldView-2 Satellite Imagery

The satellite imagery for the city of Atlanta was obtained from satellite images taken by DigitalGlobe's WorldView-2 Satellite Sensor. This satellite operates at an altitude of 770 kilometers and is able to sweep nearly 1 million km^2 every day. Its advanced on-board imaging system can capture pan-sharpened (0.46 meters GSD at Nadir, 0.52 meters GSD at 20° Off-Nadir) as well as multispectral (1.84 meters GSD at Nadir, 2.4 meters GSD at 20° Off-Nadir) images.

The imagery for the city of Atlanta is provided in the form 2 overlapping products as shown in Figure 2. Each product further consists of 2 formats - pan-sharpened and multispectral.

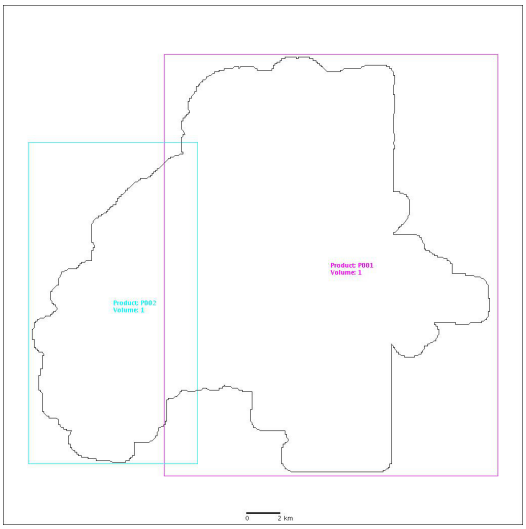

Figure 2: WorldView-2 satellite imagery layout

The multispectral imagery which is obtained at a resolution of 1.84 meters GSD consists of 8-band spectra: red, green, blue, near-IR, red edge, coastal, yellow, near-IR2. The first 3 bands were then combined to create a composite that represented the true color of the satellite imagery as shown in Figure 4.

Having obtained the true color of the satellite imagery, the products were clipped into 28×28 non-overlapping sliding window blocks following the format of the SAT-6 dataset. We determined that the resolution of the multispectral imagery was sufficient to provide the same context as the SAT-6 dataset when clipped, and hence decided to only use the RGB bands for this paper as opposed to the pan-sharpened imagery which was available at a higher resolution.

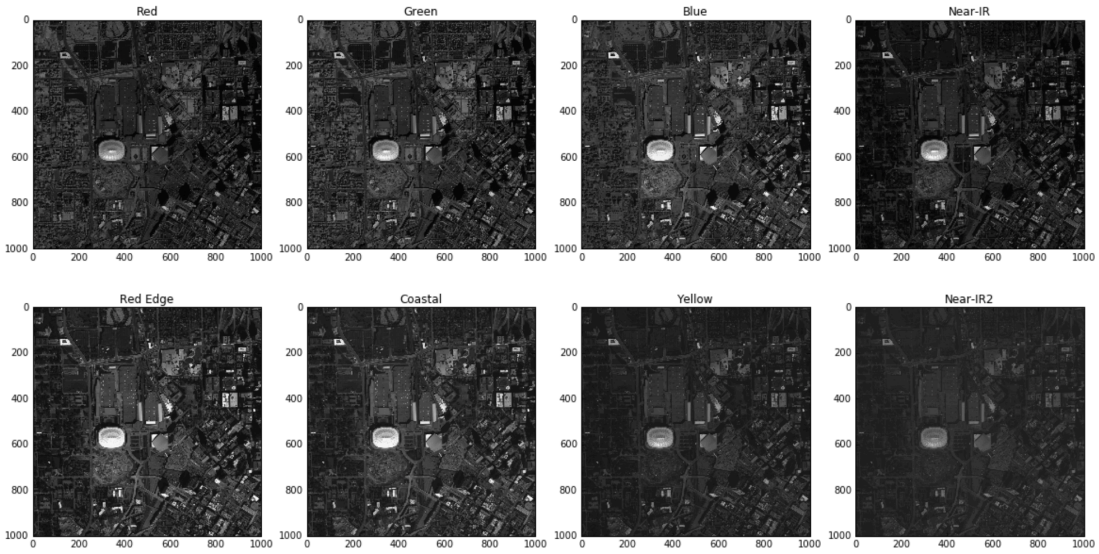

Figure 3: WorldView-2 multispectral bands

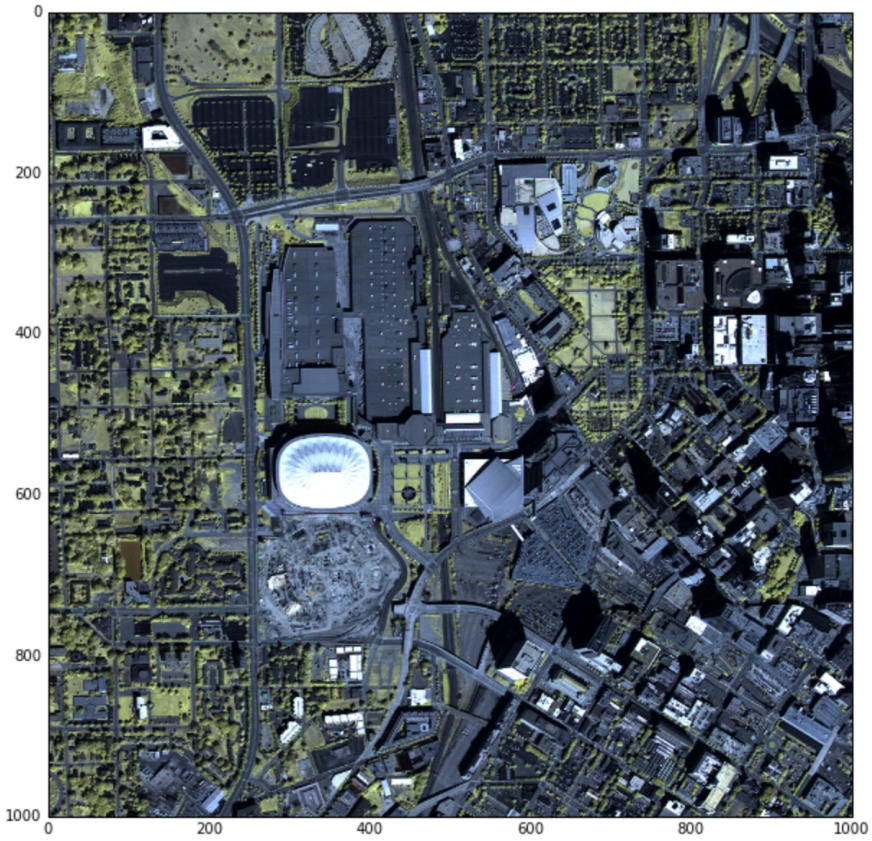

Figure 4: True color imagery obtained from RGB multispectral bands

4 Methodology and Investigation

The approach we follow in this paper to assess the urban tree canopy in Atlanta can be broadly broken down into the following steps:

1. fine-tune existing state-of-the-art, pre-trained deep neural networks so that they can be trained on the SAT-6 dataset
2. train the fine-tuned models on the SAT-6 dataset and determine the best model that can classify land cover types based on testing accuracy
3. use the best trained model to determine the land cover type across Atlanta using the city's satellite imagery dataset
4. evaluate the urban tree cover of Atlanta based on the land cover classification obtained.

4.1 Fine-Tuning Pre-Trained Deep Convolutional Neural Networks

For this paper, we consider 3 pre-trained networks, namely:

1. AlexNet [7], trained on ILSVRC 2012 dataset for 360,000 iterations
2. GoogLeNet [12], trained on ILSVRC 2012 dataset for 2,400,000 iterations
3. Places205-VGG [10], which itself is a fine-tuned variant of the popular VGG-16 CNN [11], trained on 205 scene categories of Places Database with 2.5 million images

We used the Caffe [6] deep learning framework to fine-tune, train and deploy the models. These models were obtained from the Caffe Model Zoo [2], which is a repository for popular CNN architectures, and also provides access to pre-trained model weights in the form of *.caffemodel* binaries. The models are stored in the form of *.prototxt* files and simply contain a sequential layer-by-layer description of the model.

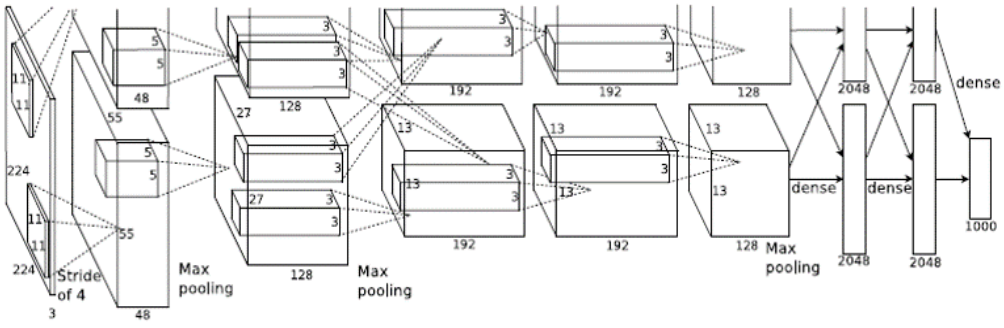

Figure 5: Original architecture of AlexNet

To fine-tune the models, we began with replacing the last layer of each model (usually a fully-connected neural network with 1,000 output nodes) with a fully-connected neural network with 6 output nodes, corresponding to the 6 classes of land cover types covered by the SAT-6 dataset. We then updated intermediate pooling layers so that signals from our images of size 28×28 could propagate through the network.

Table 2: Best performing model parameters for each architecture

Model	Learning Rate	Gamma	Weight Decay	Snapshot Iteration
AlexNet	0.001	0.3	0.0005	8,400
GoogLeNet	0.0001	0.96	0.0002	8,800
Places205-VGG	0.00001	0.1	0.0004	9,700

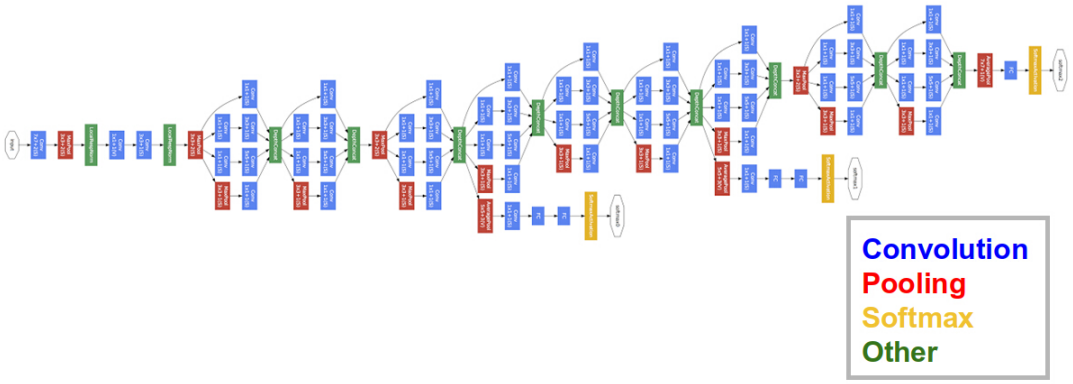

Figure 6: Original architecture of GoogLeNet

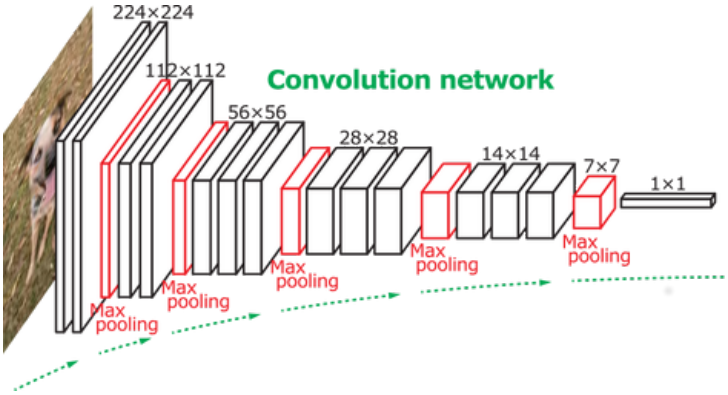

Figure 7: Original architecture of VGG-16

4.2 Training Deep CNN Architectures

We experimented with different learning rates, discount factors (gamma) and weight decay values for each model, and trained each model for 10,000 iterations. Snapshots of the weights learned by the model were taken at every 100 iterations, from which we picked the model state that yielded the highest testing accuracy on a small sample of the test set (800 images). This snapshot was then used to evaluate the final overall testing accuracy for each model. Table 2 shows the parameters for the models that were finally chosen for each architecture. Raw composite images were recreated from the RGB values in the SAT-6 dataset for training.

Figure 8 shows the training loss for each model evaluated at intermediate points during the training phase. We see here that AlexNet and GoogLeNet, which were pre-trained on ILSVRC 2012 dataset converge faster than the Places205-VGG model which has been pre-trained on scenes from the Places Database. It is also interesting to note here that the loss does not converge satisfactorily for the AlexNet and Places205-VGG models, which may intuitively lead to the argument that lowering the value of the learning rate may yield a better convergence. However, trying lower values of learning rates to train the models actually led to numeric underflows which completely halted the optimization.

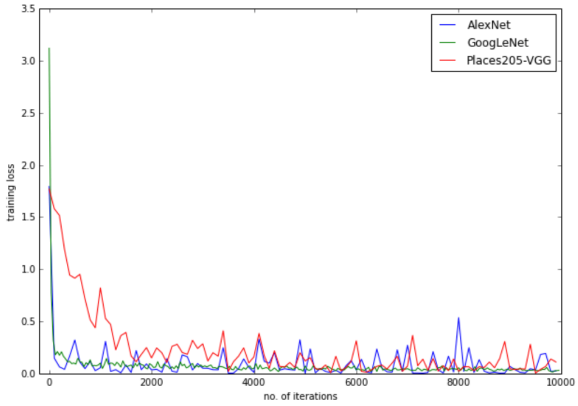

Figure 8: Training loss for each model evaluated at intermediate iterations

4.3 Determining Land Cover Type from Satellite Imagery of Atlanta

To determine the land cover type, the multispectral satellite imagery products of Atlanta were first divided into patches of size 28×28. As can be seen in Figure 2, the products are rectangular in shape and not all pixels contain satellite imagery, the patches with less than 60% imagery data were discarded. This yielded 106,914 patches from product 1 and 40,538 patches from product 2. These patches were then passed to the best performing model to predict it's land cover type. Since the products were overlapping, we decided not to merge them together and pursue our experiments on both of them independently.

5 Experimental Results

5.1 Training and Testing Fine-Tuned CNNs

We trained a number of deep models by varying the model parameters for each architecture under consideration, which amounted to ∼120 hours of training time across 3 computers. In Table 3 we present a comparison of the best of our fine-tuned models thus obtained with other proposed methods in [1] that were trained and tested on the SAT-6 dataset.

Table 3: Classification accuracy of various models

Model	**Testing Accuracy on SAT-6 (%)**
Random Forest [100 trees]	54
Deep-Belief Network [100 n/L, 3 L]	76.47
CNN [6c-3s(a)-12c-3s(m), 5×5 kernel]	79.063
Stacked Denoising Autoencoder [100 n/L, 5 L]	78.43
DeepSat [50 n/L, 2 L]	93.916
fine-tuned AlexNet	98.227
fine-tuned GoogLeNet	**99.058**
fine-tuned Places205-VGG	97.896

In Table 3, 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. Furthermore, the *x n/L y L* notation denotes that the network has x neurons per layer and y such layers. It is very interesting to note here that the fine-tuned deep neural neural networks which are only trained on raw images outperform even the best proposed model in [1], *DeepSat*, which utilizes highly complicated reconstructed features.

Figure 9 shows the number of samples misclassifed by the models for each class. One compelling insight that can be gleaned from this figure is that all 3 models almost perfectly learn how to classify

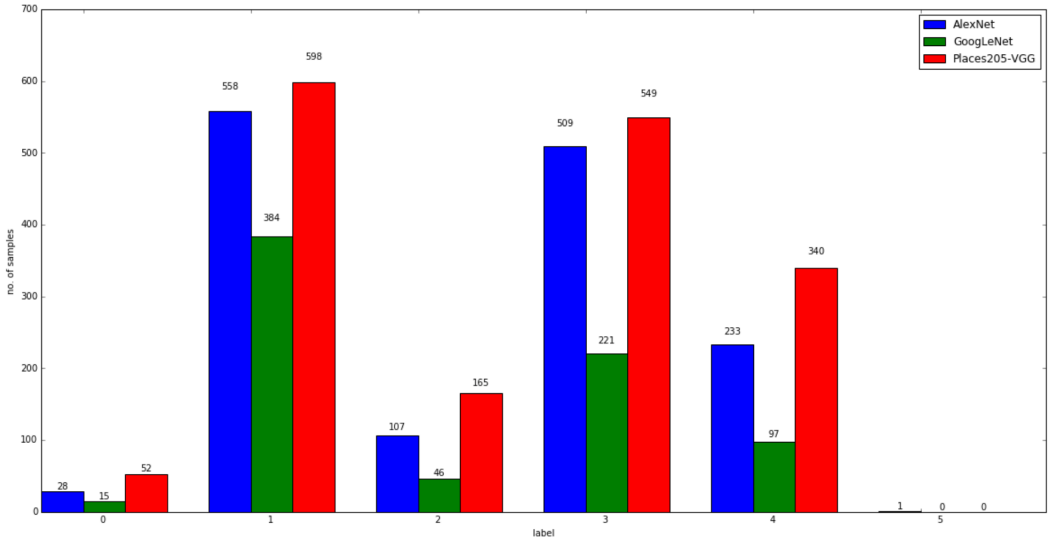

Figure 9: No. of samples misclassified per class, for each model

water bodies, which can be attributed to the fact that water bodies have very low inter-class overlap as compared to the other land-based classes.

5.2 Urban Tree Canopy of Atlanta

The fine-tuned GoogLeNet was used to predict the tree canopy using the 28×28 patches clipped from the satellite imagery of Atlanta. The patches classified as "trees" were then colored green to obtain tree canopy maps of the city as shown in Figures 10 and 11.

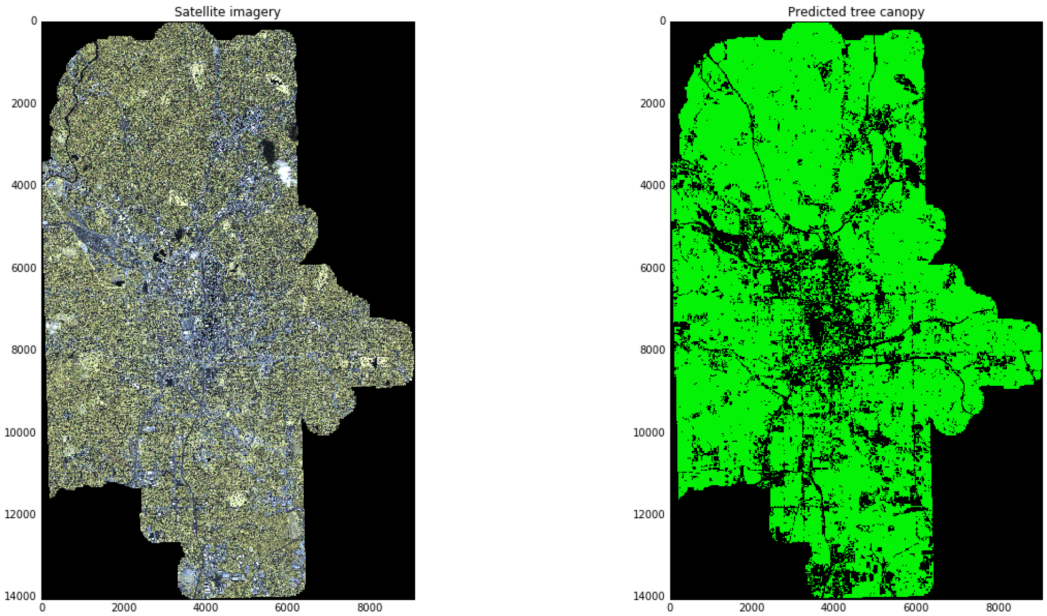

Figure 10: Tree canopy map for Product 1 of satellite imagery

Figures 10 and 11 show the satellite imagery on the left and the areas with predicted tree canopy on the right. To get a better perspective of the predicted tree canopy, we overlayed the tree canopy

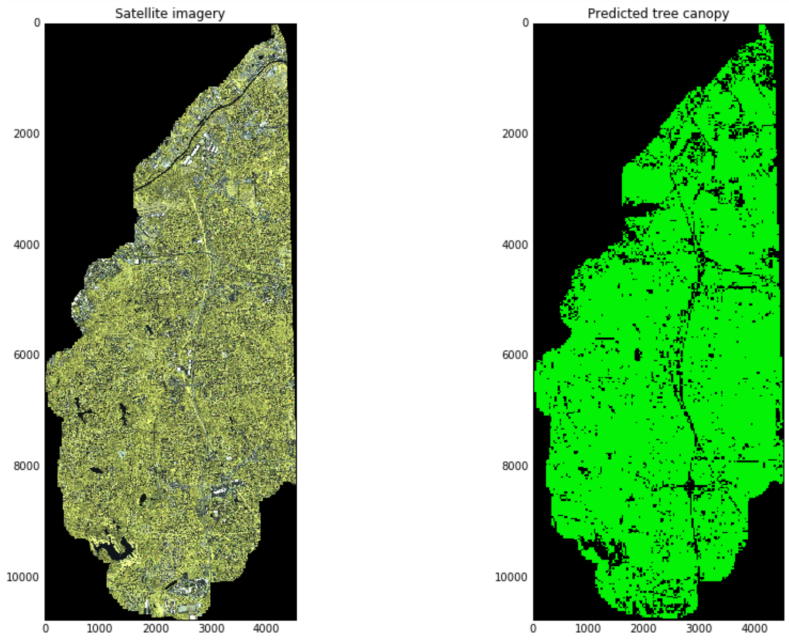

Figure 11: Tree canopy map for Product 2 of satellite imagery

map on top of the imagery and reduced the opacity of the canopy layer to evaluate the quality of the predictions. Figures 12 and 13 show the results thus obtained.

Figure 12: Tree canopy overlayed on satellite imagery

At first glance, our proposed method of determining urban tree canopy seems to give satisfactory results (Figure 12). However, taking a closer look at the predictions (Figure 13) reaveal that the classifier is still not as perfect as one would desire. It seems to classify any patch that contains even

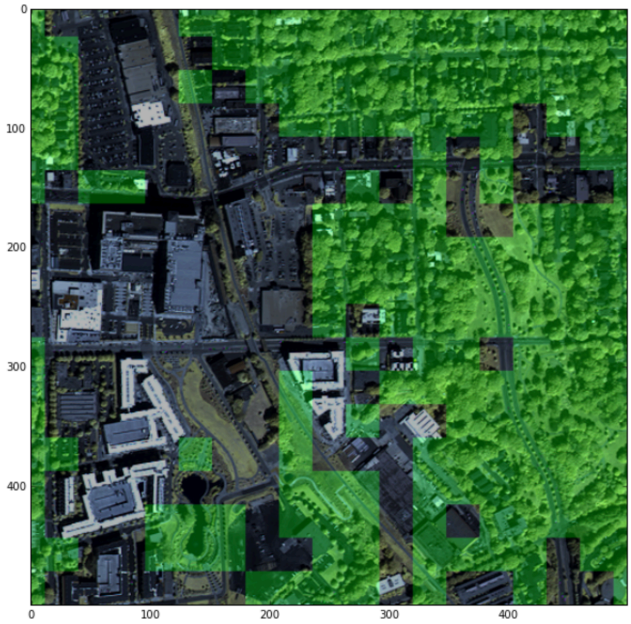

Figure 13: Tree canopy overlayed on satellite imagery (further zoomed in)

a portion of a tree in minor composition as containing trees as a whole. This may be be an artifact of the manner in which the training data was annotated or stem from the fact that the SAT-6 dataset was at a different resolution of GSD than Atlanta's satellite imagery. It may be possible to make the predictions at a finer scale by sampling overlapping patches.

6 Evaluation

Since there is no readily available annotated data for the satellite imagery of Atlanta, directly evaluating our proposed method to determine urban tree canopy is not trivial. We visually inspected the results to ensure the predictions are satisfactory. We randomly sampled the satellite imagery in a stratified manner and annotated 100 patches. We found that 86 of those patches were correctly classified. Out of the incorrectly classified, 11 patches were such that were composed of trees in a minor composition, and were classified as trees, whereas the annotator classified them differently.

7 Discussion and Future Directions

In this paper we show that raw satellite imagery is sufficient to classify land cover types using deep convolutional neural networks, which are pre-trained to identify higher level hierarchical representations of a completely different problem domain. We saw that the "knowledge" gained by these models on datasets such as ILSVRC 2012 and the Places Database can be reused to quickly learn new representations of satellite imagery.We also show that these models outperform some proposed methods which employ far more complicated representations of the SAT-6 dataset, for which we were able to obtain an accuracy of 99.06% - a significant jump up from the state-of-the-art. We then leveraged this newly trained model to obtain a tree canopy coverage for the city of Atlanta.

Even though we were able to achieve a model that surpassed the published best for the SAT-6 dataset, closer inspection of our performance in predicting urban tree canopy for Atlanta only revealed satisfactory results. We plan to make predictions at a much finer scale by sampling the patches in overlapping strides, and determining the class labels based on majority of the prediction for the overlapping patches. We also plan to investigate the performance of these models on multispectral data trained from scratch. In order to realize this, we plan to generate a SAT-6 style annotated dataset, which may not be as huge, but sufficient for our purpose.

We believe that our results would open debate about simpler representation and serve as a baseline for creating better architectures specially conforming to our problem of satellite imagery classification.

References

[1] Saikat Basu et al. "DeepSat-A learning framework for satellite imagery". In: *arXiv preprint arXiv:1509.03602* (2015).

[2] *Caffe Model Zoo*. URL: https://github.com/BVLC/caffe/wiki/Model-Zoo.

[3] Marco Castelluccio et al. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks". In: *arXiv preprint arXiv:1508.00092* (2015).

[4] *CS231n Convolutional Neural Networks for Visual Recognition*. URL: http://cs231n.github.io/convolutional-networks/.

[5] Kunihiko Fukushima and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[6] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[8] *NAIP*. URL: http://www.fsa.usda.gov/Internet/FSA_File/naip_2009_info_final.pdf.

[9] *NLCD*. URL: http://www.mrlc.gov/nlcd11_data.php.

[10] *Places205-VGG*. URL: http://places.csail.mit.edu/downloadCNN.html.

[11] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[12] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.