Understand Amazon Deforestation using Neural Network

Chao Liang
Department of Geophysics
chao2@stanford.edu

Meng Tang
Energy Resources Engineering
mengtang@stanford.edu

Abstract

We investigate a detection algorithm to detect and track changes in forests in Amazon Basin. This problem is interesting because deforestation in the Amazon Basin is becoming more and more severe, contributing to reduced biodiversity, habitat loss, climate change, and other devastating effects. And better data about deforestation situations and causes can help governments and local stakeholders respond more quickly and effectively to protect the forest. In this project, we use a data set of satellite images to train neural network models to understand the deforestation in Amazon forest using high resolution satellite images. We mainly leverage pretrained VGG model to extract low level features from RGB channels and currently disregard the information from infrared channels. We use both data augmentation and more specialized neural nets to overcome the severely skewed class label distribution. Through this project, we explore the effect of data augmentation, training strategy and network architectures on prediction accuracy.

1. Introduction

The largest deforestation of the world is happening in Amazon Basin every day, which also speeds up reduced biodiversity, climate change, habitat loss, and other devastating effects. To help address this problem, thousands of satellite image chips with atmospheric conditions and different classes of labels have been collected. However, no efficient working methods have been developed to label those planet images to differentiate different causes of forest loss.

We investigated using pretrained convolutional neural network (CNN) to detect and track changes of forests in Amazon Basin based on those high resolution satellite images. To be more specific, we aimed at assigning multi-labels to each image including categories of atmospheric conditions, common land cover/land use phenomena,and rare land cover/land use phenomena. Each image has one atmospheric label and zero or more common and rare labels. And the different labels are not independent. For example, chips that are labeled as cloudy should have no other labels, because if it's cloudy, no view on the ground can be achieved. However, there may be labeling errors due to labeling process and ambiguity of features. Some class labels may be missing or incorrect. We are expecting that we are working with data with relatively high noise.

The input to our algorithm is a satellite image with four channels: red (R), green (G), blue (B) and near infrared (NIR). We then use a CNN to output predicted multiple labels. Each chip will have exactly one atmospheric label and zero or more common and rare labels. Chips that are labeled as cloudy for atmospheric label should have no other labels. Different methods have been tried for object recognition in satellite images.

Training data provided contains 40479 satellite images with 4 channels (RGB and near infrared) with 16 bits color [12]. Each image comes with a tag, which can include three types of labels: atmospheric condition, common land cover/land use phenomena, and rare land cover/land use phenomena. We split the provided training data into training set and validation set. The training set contains 36000 images randomly picked from the training data, the validation set contains the left 4479 images from the validation set.

Three categories of labels includes 17 specific labels in total. To be more specific, weather category contains: clear, partly cloudy, cloudy, and haze, which closely reflect what we observe in a local weather forecast. Haze is defined where atmospheric clouds are visible but they are not opaque enough to block the ground view. Cloudy is defined as 90% of the ground view on the image blocked by opaque cloud cover. Partly cloudy is defined as opaque cloud covers part of the image. Clear is defined as no cloud or haze exist and the ground is very clear. The common labels in this dataset includes primary rain forest, Water (rivers and lakes), Habitation, agriculture, road, cultivation and bare ground. Rare labels includes the slash and burn agriculture, selective logging, blooming, conventional mining, and ”artisinal” mining. And a random selection of 8 images and their corresponding labels are shown in Figure 1.

Our problem is a multi-class (non-exclusive) classifica-

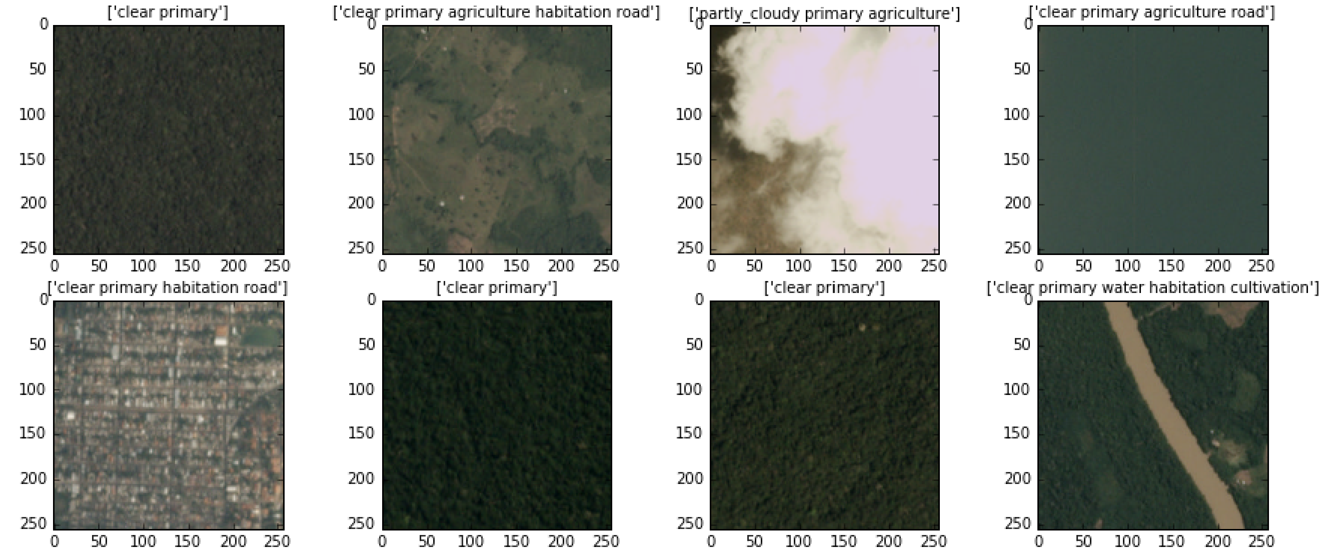
['clear primary']
['clear primary agriculture habitation road']
['partly_cloudy primary agriculture']
['clear primary agriculture road']
['clear primary habitation road']
['clear primary']
['clear primary']
['clear primary water habitation cultivation']
0
50
100
150
200
250

Figure 1. Eight randomly sampled images (showing the RGB channel) with different tags.

tion problem. Two steps are needed to solve it: one is feature extraction and another is classification. We extract features using mainly CNNs and feed them into fully connected layer to outputs a score for each class. The scores are then used to output the presence or absence of each class label. To quantify the loss of multi-class(non-exclusive) classification problem, we introduce the binary cross entropy loss. The binary cross entropy loss generates a probability vector corresponding to each label class instead of a scalar in the end. So we select a probability criteria to decide whether the label exists or not. To be more specific, if the element in generated probability vector is greater than the criteria, then we assume set the predicted label for the corresponding class is 1, otherwise 0. At the same time, consider that the classes in the weather category are exclusive, we later employ the softmax cross entropy loss for weather label prediction and combine two losses together. The CNN investigated in this report is one simple CNN and pretrained VGG-16 [12].

2. Background and Related Work

As a challenging topic, satellite image classification usually involves in tons of data and large variations. Traditional machine learning algorithms such as random forest cannot handle extractions of a huge amount of features. Recently, people begins to turn to focus on machine learning, CNN and Deep belief Network for classification of satellite images. Muhammad et al. [1] extract attributes including organization of color pixels and pixel intensity using decision tree for training, although the test is only conducted on a very limited data set. Goswami et al. [2] used satellite images to train a multi-layer perceptron (MLP) model which employs back propagation (EBP) learning algorithm for a two-class waterbody object detection problem. Although the problem itself is simplified because there's only two classes, but they can achieve comparatively high test accuracy and demonstrate the applicability of neural network on satellite image classification. Basu et al. [13] builds a classification framework which extracts features from an input images and feeds the normalized feature vectors to a Deep Belief Network for classification. The authors show that their framework outperforms CNN over 10% on two datasets they build up. Penatti et al. [14] use arial and remote sensing images to train CNNs and make comparisons between the performance of the ConvNets and other low-level color descriptors and show the advantages CNNs get. They also shows the possibility of combining different CNNs with other descriptors or fusing multiple CNNs. Nogueira et al [15] further explore the CNN for classification of remote sensing images. They extract features using CNNs and conduct experiments over 3 remote sensing datasets by using six popular CNN models and achieve state-of-art results.

Another important topic directly related to satellite image classification problem is object detection because there are usually multiple labels for a single satellite image and what are in the image are always required to be clarified. A naive approach for locating objects in images is sliding window boxes with different sizes at selected locations of the image and classify those window boxes [3, 4]. This can lead to satisfying accuracy but incur considerable computational cost [5]. By replacing fully connected layer with convolutional neural network, the computational cost is reduced significantly [5]. Regional convolutional neural network (R-CNN) rises as an efficient and accurate approach to addressing object detection problems [6]. It first employs a regional proposal method to locate regions of possible interest in the image, then applies neural network to classify the object proposed [7,8]. Segmentation works as another state-of-the-art approach for object detection [9, 10, 11]. The basic idea of segmentation approach is to generate a label map for each pixel in the image.

3. Approach

As mentioned in the introduction section, we mainly use provided training data set which contains 40479 images with 4 channels (RGB and near infrared) and 16 bits color. Each image in the training dataset has dimension of $256 \times 256 \times 4$. Each image can have multiple labels at the same time.

Consider the loading efficiency for a comparatively large data set, we first performed data preprocessing by loading all the training images and their labels into a numpy array. We compress the training images from 16 bits color into 8 bits color to reduce the data file size and improve the loading efficiency. By conducting this, we finally obtain a numpy array which contains 40479 images data and can be loaded within 2 minutes for future training. We further split the training data into training set (36000) and validation set (4479).

Some work has been done and posted on Kaggle discussion panel about the little impact of NIR data on the improvement of trained model and prediction accuracy. While at the same time, the fourth channel increase the data size enormously. We believe that the NIR (near infrared) can hardly help the prediction and consider that it can possibly overload GPU, so currently we disregard the NIR data in our project. So we truncate the dimension of our data set to be $40479 \times 256 \times 256 \times 3$ by ignoring the NIR channel.

Next, to address the multi-label (non-exclusive) classification problem, we intuitively adopt the binary cross entropy loss to detour around more complicated object detection technique, which proves to work well later. Let there be m possible class labels present in the tag of each image. Let p_j be the probability of label j appearing in the image. The correct label for the image is expressed by a vector y_j:

y_j=1, if the j_{th} label appears in the tag of image.

y_j=0, if the j_{th} label does not appear in the tag of image.

Now, the loss for this image can be defined as:

$$L = \sum_{j}^{m} -log(p_j)y_j - log(1-\sigma(s_j))(1-y_j)$$

$$L = \sum_{j}^{m} -log(\sigma(s_j))y_j - log(1-\sigma(s_j))(1-y_j)$$

where, $\sigma(x)$ is the sigmoid function. We minimize the mean loss defined by the equation above. Essentially, this loss function tries to minimize the score for each incorrect class label and maximize the score for each correct class label. After defined our method, we now proceed to outline our model strategy and different experiments that we plan to conduct.

One feature of the label we notice is that weather labels are exclusive, that is we can have one and one only weather label, we assign weather label a different loss function: softmax cross entropy loss. And it is defined by the equation below:

$$L = \frac{\exp(s_y)}{\sum_j \exp(s_j)}$$

.

To evaluate the performance of the trained model, F_2 score is adopted. The F score, commonly used in information retrieval, measures accuracy using the precision p and recall r. Precision is the ratio of true positives t_p to all predicted positives $(t_p + f_p)$, where f_p represent number of false positives. Recall is the ratio of true positives to all actual positives $(t_p + f_n)$, where f_n represent the false non positives. The F_2 score is given by

$$F_2 = (1+\beta^2)\frac{pr}{\beta^2 p + r},$$

$$where p = \frac{tp}{tp+fp}, \; r = \frac{tp}{tp+fn}, \beta = 2.$$

The mean F_2 score is computed by averaging the individual F_2 scores on each row in the validation set.

We first try building a simple CNN model which contains 4 convolutional layers and corresponding activation and regularization process and three fully connected layers specifically for weather label prediction. As we mentioned, the four weather labels (clear, haze, ”partly cloudy” and cloudy) are mutually exclusive. We develop a model to classify all the images into these 4 weather classes. If the cloudy is predicted, we also know that no other labels will be present. This will let other model nets be more focused on predicting other type of labels. Besides, we can simply adopt softmax cross entropy loss to train the model.

Next, we use the pretrained VGG-16 model to extract the useful low level features from the RGB channels of the image. We discard the NIR channel and disable data augmentation for now. In this model, we train the network to output prediction of the presence of all 17 class labels at the same time, but use softmax cross entropy loss for the weather and sigmoid cross entropy loss for other non-exclusive common and less common labels. The difference between the previous VGG-16 model and modified one is shown in figure 2.

Finally, we use the pretrained VGG-16 model but enable data augmentation and balancing this time. Exploring the impacts of data augmentation and data balancing on

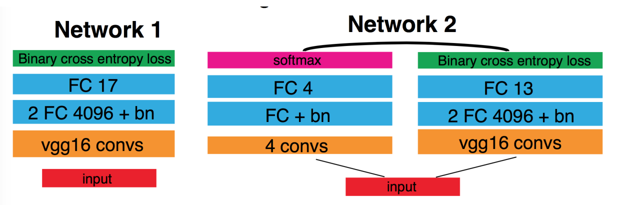

Figure 2. Network Architecture for two modified VGG-16 models.

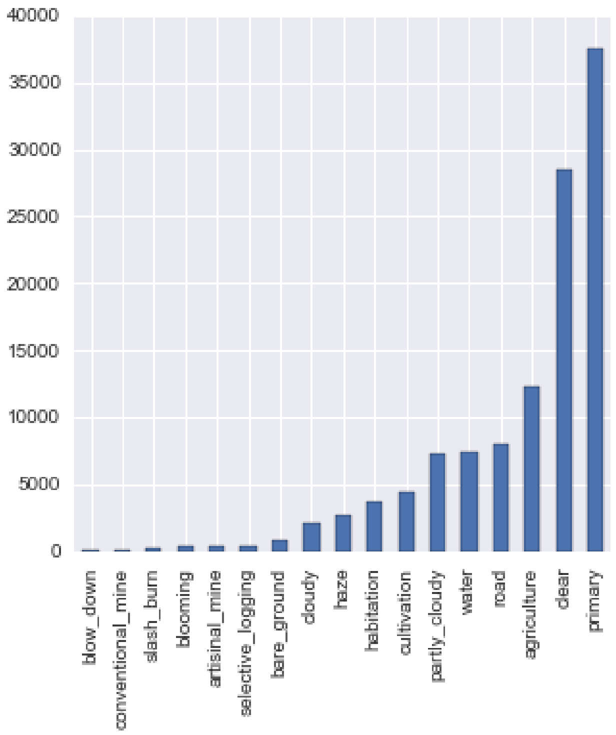

Figure 3. Distribution of class labels.

improving training result is one of the main topics in this project.

One obvious feature of this data is that our training data set is extremely unbalanced. For example, The overwhelming majority of the data set is labeled as ”primary”, which stands for primary rainforest. The distribution of the each individual label is shown in figure 3. We can easily observe that the dominant labels are weather label ’clear’ and common label ’primary’. The highly skewed distribution of label from random sampling indicates that data augmentation and balance may be necessary to train a reliable model. We also see high dependency between agriculture, habitation and roads.

Data augmentation and data balancing can improve our training result. In this first place, we used a relatively small training data set, which only contains 8000 images for our model training, and kept receiving undesirable training results including fast convergence to local minimum and low training accuracy. After we switch to a larger training dataset, everything starts to work out. And we know that data augmentation is extremely helpful to increase the size training set, which particularly helps those labels with limited number of image examples. We randomly flip, rotate and shift the training images while training to an infinite flow of transformed training images. A simple illustration of flip, rotate and shift of images is shown in Figure 4.

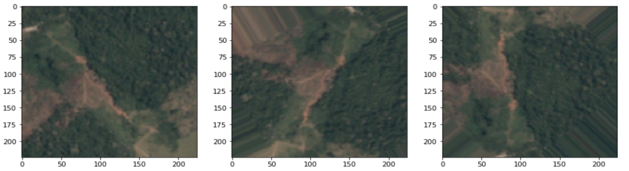

Figure 4. Examples of training images generated by data augmentation. The middle and right-most images are generated by randomly rotating and fliping left-most image.

To make up for the unevenly distributed labels and their corresponding images, we carried out data balancing by randomly downsample the top 5 major labels to the fregure Data balancing is done by randomly downsample the top 5 majoriy labels to the frequency of the first five majority labels. The heavy data augmentation and data balancing are conducted to prevent training and network that can only precisely identify clear primary and fail on other labels.

4. Experiment Result

4.1. Weather label forecasting

Since the four weather classes (clear,haze,partially cloudy, cloudy) are mutually exclusive, we designed the small CNN network to classify them separately from the network that focuses on classify the land forms. This rather simple net achieves 91% validation accuracy. And the history of training and validation accuracy over iterations is shown in figure 5.

We can observe that training accuracy increases steadily while validation accuracy converges to 91% after 2×10^4 iterations. Here we take a batch of 100 randomly selected samples from 36000 training examples every iteration due to the memory limit of GPU. For validation accuracy, we iterate to compute the validation accuracy over 4479 validation examples to detour around the GPU memory limit. Here, no data augmentation nor balancing is conducted to assist training. We can observe some sudden decrease in both training and validation accuracy, which is mainly due to sample batch with large noise or unbalanced data. This can be partly overcome by proper data augmentation and balancing.

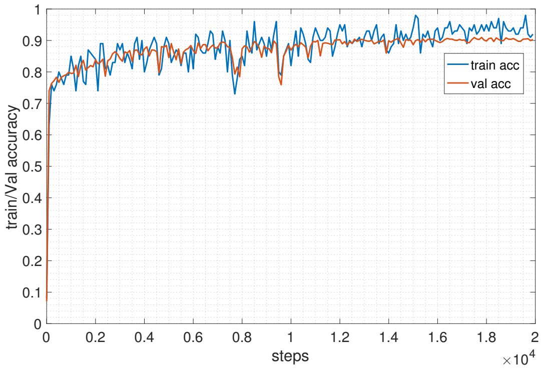

Figure 5. Training and validation accuracy for weather prediction.

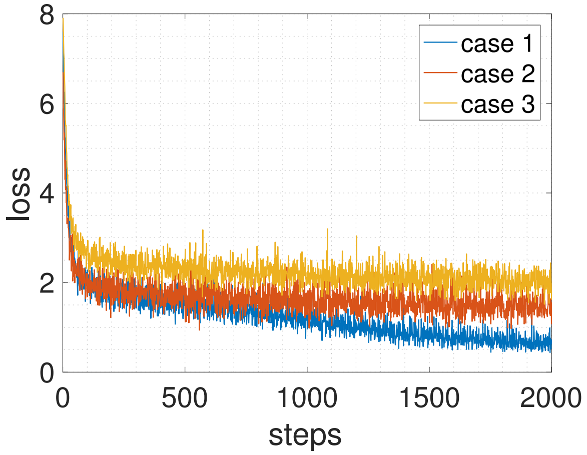

Figure 6. Training loss comparison for different data preprocessing (case 1: skewed and no augmentation; case 2: skewed and augmentation; case 3: balanced and augmentation).

4.2. Impact of data augmentation and balancing

Several control experiment has been conducted to study the impact of data augmentation and balancing in this project. And all the experiments are carried out based on using the pretrained VGG-16 network. Because the input size of images for pretrained VGG-16 is $224 \times 224 \times 3$, so we randomly truncated the $256 \times 256 \times 3$ images into $224 \times 224 \times 3$. Resizing the images in this way can hardly affect the training result because the features of the image can easily be maintained by a large portion of the image.

We compared the training results over a training data set with size 5000 among using skewed and augmented data, skewed and non-augmented data, and balanced and augmented data. The comparison of training loss for different data preprocessing is illustrated in Figure 6. And the comparison of F_2 scores for training and validation using different data preprocessing is shown in Figure 7.

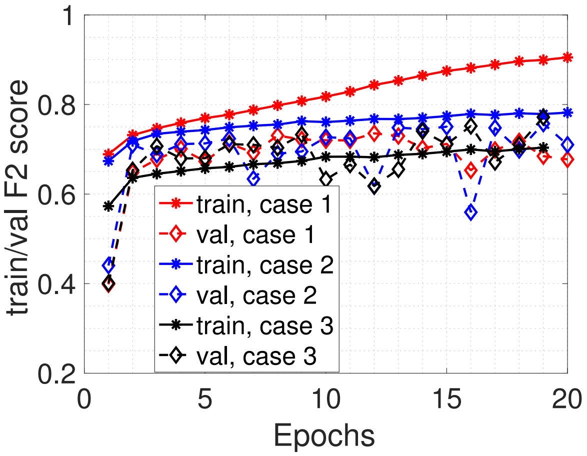

Figure 7. Training loss comparison for different data preprocessing (case 1: skewed and no augmentation; case 2: skewed and augmentation; case 3: balanced and augmentation).

We can see from Figure 7 that when using skewed data with no augmentation, we can achieve very high training accuracy while relatively low validation accuracy, which is a sign of overfitting. With augmentation added in case 2, the training and validation score track nicely with each other, effectively preventing overfitting. When data balancing is applied in case 3, we also observe that over-fitting is overcome but with higher oscillation in validation score, which we argue is from the different distribution of training and validation set. In our experiment, only training set is balanced and validation set is still skew. We also notice that both data augmentation and data balancing increases the training loss compared to the case without balancing or data augmentation, shown in Figure 6. We concluded that it's more difficult to train on a blanced dataset. This might be counter-intuitive but it actually makes sense. If the dataset is skew, it's relatively ”easy” for the network to learn just the feature of the majority class and converge. With a balanced dataset, the network needs to learn more features and thus takes more iterations and more careful tuning to reach high accuracy.

5. Conclusion and Future Work

By the control experiments conducted, we realize that direct training on dataset with highly skew class distribution can bias the model to predict the majority class. Data augmentation is helpful in improving learning and preventing overfitting while data balancing is helpful in preventing the model fitting to the majority class but also more difficult to learn well. The particularly exclusive signature of different weather labels can be classified separately by simpler net and loss function, which reduce the burden of the other net by letting it focused on classifying different landforms.

So far we only employed the RGB channels for training but ignored the NIR channel because as we mentioned some work indicates that NIR channel is not very helpful. But it can be a model by model case. So we plan to explore using NIR data as the fourth channel on our model. Besides, color histogram and histogram of gradient orientation can also be adopted to assist CNN to capture features more fast and accurately. For example, we expect that the histograms of images with different weather tags can be reflected by the area where white color is synthesized on the color histogram. More importantly, we believe that separating the prediction of labels into different classifiers (CNN models) is also an interesting and promising way worth trying. On one hand, separating the rare labels into a group and using an individual CNN classifier for training is equivalent to data balancing. Because images with those labels have the same magnitude of quantities. We believe with the help of data augmentation to increase the size of training data set, better training and validation results can be achieved by setting up individual VGG models for different categories of labels.

References

1] Shahbaz, Muhammad, et al. ”Classification by object recognition in satellite images by using data mining.” Proceedings of the World Congress on Engineering. Vol. 1. 2012. APA

[2] Goswami, Anil Kumar, Shalini Gakhar, and Harneet Kaur. ”Automatic object recognition from satellite images using Artificial Neural Network.” International Journal of Computer Applications 95.10 (2014).

[3] Dalal, Navneet, and Bill Triggs. ”Histograms of oriented gradients for human detection.” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[4] Viola, Paul, and Michael Jones. ”Rapid object detection using a boosted cascade of simple features.” Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[5] Sermanet, Pierre, et al. ”Overfeat: Integrated recognition, localization and detection using convolutional networks.” arXiv preprint arXiv:1312.6229 (2013).

[6] Girshick, Ross, et al. ”Rich feature hierarchies for accurate object detection and semantic segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[7] Ren, Shaoqing, et al. ”Faster r-cnn: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems. 2015.

[8] Farabet, Clement, et al. ”Learning hierarchical features for scene labeling.” IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1915-1929.

[9] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. ”Fully convolutional networks for semantic segmentation.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[10] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. ”Learning deconvolution network for semantic segmentation.” Proceedings of the IEEE International Conference on Computer Vision. 2015.

[11] Liu, Yu, et al. ”An improved cloud classification algorithm for Chinas FY-2C multi-channel images using artificial neural network.” Sensors 9.7 (2009): 5558-5579.

[12] https://www.kaggle.com/c/planet-understanding-the-amazon-from-space. Last accessed May 16, 2017.

[13] Basu, Saikat, et al. ”Deepsat: a learning framework for satellite imagery.” Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2015.

[14] Penatti, Otvio AB, Keiller Nogueira, and Jefersson A. dos Santos. ”Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015.

[15] Nogueira, Keiller, Otvio AB Penatti, and Jefersson A. dos Santos. ”Towards better exploiting convolutional neural networks for remote sensing scene classification.” Pattern Recognition 61 (2017): 539-556.