Amazing Amazon: Detecting Deforestation in our Largest Rainforest

Aaron Loh
MS CS '17
Stanford University
aaronlwy@stanford.edu

Kenneth Soo
MS Stats '17
Stanford University
kensoo@stanford.edu

Abstract

In this paper, we tackle a multi-label classification problem, aiming to label satellite chips with tags corresponding to land cover, land use, as well as atmospheric conditions. Using a dataset from Planet, consisting of 256 x 256 x 4 satellite chips, we use CNNs with different architectures to generate classifications for each chip. We attempt transfer learning using VGGNet, and also train a model with the ResNet50 and InceptionNet architectures as a base. We further experiment with a simple 4 layer CNN. Our best result is able to get a F2 Score of 0.89.

1. Introduction

Deforestation is an urgent problem in our world today, as it contributes to reduced biodiversity, habitat loss, climate change, and other devastating effects. In the Amazon river basin, a rainforest that covers 40% of continental South America and spans across nine countries, 0.2% of the forest is lost to deforestation each year. We hope to tackle this problem by using satellite data to track deforestation and help researchers better understand where, how and why deforestation happens, and how to respond to it.

With advances in satellite imagery, detection of deforestation has become faster, more convenient, and more accurate than before. An example of an ongoing effort is the Real Time System for Detection of Deforestation (DETER) which has been credited for reducing the deforestation rate in Brazil by almost 80% since 2004, by alerting environmental police to large-scale forest clearing. [20] Current tracking efforts within rainforests largely depends on coarse-resolution imagery from Landsat (30 meter pixels) or MODIS (250 meter pixels). The challenges faced by these methods are the limited effectiveness in detecting small-scale deforestation or differentiating between human causes of forest loss and natural causes.

Planet, designer and builder of Earth-imaging satellites has a labelled dataset of land surfaces at the 3-5 meter resolution, and we propose leveraging modern deep learning techniques to identify activities happening within the images. We treat this as a multi-label classification problem, and we aim to label satellite image chips with one or more of 17 labels that indicates atmospheric conditions, land cover, and land use.

2. Related Work

As early as 1998, Cohen et al. [4] demonstrated that Landsat [16] imagery can be used to map forest clear cuts in the Pacific Northwest. Popatov et al. [21] and Hansen et al. [14] combined MODIS [17] and Landsat data to estimate forest cover change in boreal forests and the Congo Basin respectively. To improve the fusion of MODIS and Landsat data for analysis, Hilker et al. [8] developed a new data fusion model specifically to analyze forest disturbance. In 2012, Zhu et al. [26] developed a novel year-long, continuous, time-series model by monitoring multiple images taken during the growing season at the Savannah River site. Their work was further developed by Diaz. [5] The urgency of the problem of deforestation has also led to the Brazilian government to establish their own real-time system that monitors forest clearing, called DETER and PRODES. [20, 3]

Machine learning methods to analyze satellite data only came to prominence in the recent years, and there have been several attempts to do so. Kehl et al. [11] trained an Artificial Neural Network model on satellite images from the MODIS/TERRA sensor, and conducted a spectrum-temporal analysis of the study area. Mnih and Hinton [15] used large-scale neural networks, with additional help from local spatial coherence of the output labels, to detect roads in high-resolution aerial images. In addition, Jean et al. [9, 10] used convolutional neural networks to identify image features in satellite data that could explain up to 75% of the variation in local-level economic outcomes in five African countries. A different approach was used by Kluckner et al. [12], and they applied covariance descriptors to a multi-class randomized forest framework for semantic classification of aerial images.

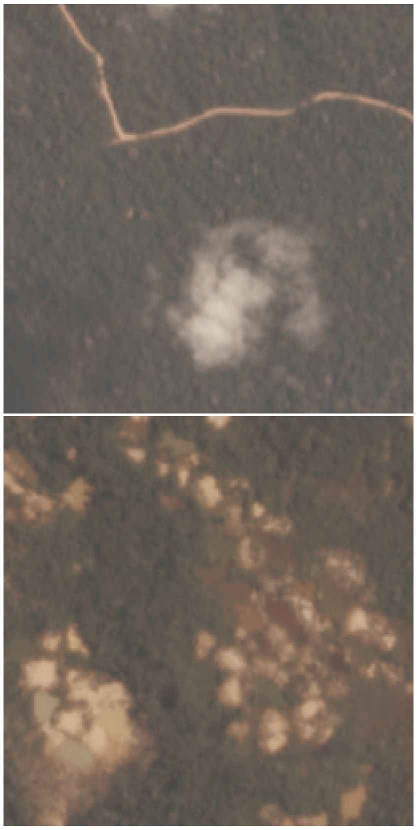


Figure 1. Sample images from the dataset. The top image contains the labels: Partially Cloudy, Primary, Water, Road. The bottom image contains the labels: Clear, Primary, Water, Bare Ground, Artisinal Mining. Illegal logging is present in the bottom image.

3. Dataset and Features

We split the data into roughly 35,000 images for training, 3000 images for validation, and 2000 images for testing. Each image contains 4 channels: red, green, blue, and near infrared. Sample images are presented in Figure 1.

3.1. Distribution of labels

Each image can contain one or more of 17 labels, which can be broadly classified as follows:

Cloud Cover Labels. Each image will have exactly one of the following four labels, where the parenthesis indicates the number of training images that contain that label: Clear (28431), Partially Cloudy (7261), Cloudy (2089), Haze (2697). Images that contain the Cloudy label will contain no other labels.

Common Labels. Each image can have zero or more of the following six labels, where the parenthesis indicates the number of training images that contain that label: Primary (37513), Agriculture (12315), Road (8071), Water (7411), Cultivation (4477), Habitation (3660).

Rare Labels. Each image can have zero or more of the following seven labels, where the parenthesis indicates the number of training images that contain that label: Bare Ground (862), Selective Logging (340), Artisinal Mining (339), Blooming (332), Slash and Burn (209), Conventional Mining (100), Blow Down (98).

3.2. Co-appearance of labels

As a preliminary analysis of the distribution of labels in the dataset, we investigated whether certain labels tend to co-appear in images.

To do so, we created a co-appearance metric that measures the change in appearance of a label when another label is present. Suppose we have n images and two sets of labels, x and y. We have $x_i = 1$ if the ith image contain label x, and $x_i = 0$ otherwise. Let p_x be the proportion of images that contain label x, and $p_{x|y}$ be the proportion of images that contain label x given that label y is present. Then,

$$p_x = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$p_{x|y} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} y_i} \tag{2}$$

The change in appearance of label x in the presence of y is given by:

$$C_{xy} = \frac{p_{x|y}}{p_x} = \frac{n\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i} \tag{3}$$

Notice that this metric is symmetric, i.e. $C_{xy} = C_{yx}$. We calculated C_{xy} for all label-pairs and applied a transformation to these values so that they lie in [-1,1]. A matrix of these values is presented in Figure 2.

Based on the matrix, we noticed that the labels can be roughly divided into 4 clusters, as indicated by the 4 black boxes within Figure 2. The most top-left box contains the 4 cloud cover labels, and the values corroborate with what we know about these labels (that each image contain exactly one of the 4 labels). For the other 3 boxes, the boxes tend to be positive, whereas the co-appearance values outside the boxes tend to be negative, indicating clustering of the labels. Knowing this information allowed us to verify that our model made sensible predictions. It can also be used to create a better prediction model, which has potential as a future work.

4. Methods

4.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been very successful in recent years for a large number of visual tasks, such as image recognition and video analysis, because of

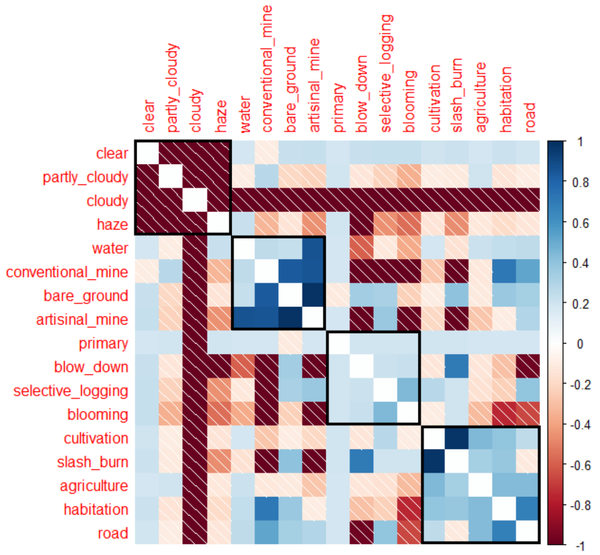


Figure 2. Co-appearance of labels in data. Blue boxes indicate that the presence of a label will make the other label more likely to appear. Red boxes indicate that the presence of a label will make the other label less likely to appear.

their ability to capture structured representation of data. CNNs are in fact an adaptation of a machine-learning model called Neural Networks for visual tasks, and they make use of the special structure present in visual data to the improve efficiency and effectiveness of the Neural Network model.

While vanilla Neural Networks require neurons in successive layers to be fully connected to each other, in CNNs a convolving filter is used across the image and neurons in one layer are only connected locally to the neurons in the preceding layer. For each neuron in the current layer, a dot product is computed between the parameter weights of the convolving filter and the local region in the preceding layer. The parameter weights of the convolving filter are re-used for different neurons as it moves across different parts of the preceding layer. This presents two advantages. First, this drastically reduces the number of parameters that need to be learnt by the model, which allows the model to be trained faster vis-a-vis the vanilla Neural Network. Secondly, because the filter uses the same set of parameter weights while convolving across different parts of the image, this gives CNNs a translational invariance property. The implication is that identical objects appearing in different parts of the image can be recognized as being identical.

The layers described above are known as convolutional layers, because of the convolving filter technique applied to the layers. However, other types of layers are also present in the model, such as the ReLU layer that sets a thresholds for neuron activations, the pooling layer that performs downsampling operations along the spatial dimensions, and dropout layers that randomly deactivates neurons in order to reduce overfitting. These layers can be applied repeatedly to the network. Finally, a fully connected layer similar to the vanilla Neural Network is added at the tail end of the network, and it computes the class scores, which tells us the probability that the image contains a particular label. The CNN architecture that we used for this problem is visualized in Figure 3.

Because our problem is a multi-label one, we had to tweak the output layer to accommodate multiple labels. In usual CNNs, the output layer is usually a softmax function, which squishes all values of a vector onto the range of [0,1], summing together to 1, and is thus only able to express class probabilities for a single-label. To obtain probabilities for multiple labels, we applied a sigmoid function given by

$$\sigma(x) = \frac{1}{1 + e^x} \tag{4}$$

on the logits. We then round the output to generate our prediction, so that each label receives a score of 0 or 1.

Lastly, we noted that our CNN architecture did not exploit the special structure present in the data labels - that each image would have exactly one of the four cloud cover labels, and that images with the ”Cloudy” label would have no other labels. While it is possible to use hierarchical models or even create a special output layer to address this, we decided to first try a simpler model due to limited time and resources. We discovered that the results we obtained were comparable to that of others who were doing the same problem.

4.2. Resnet

Residual Networks, or ”Resnets”, is a variant of CNN developed by He et al. [7] We decided to use a Resnet as a base for our CNN model, as it is a state-of-the-art architecture known for its superior performance. High layer-depths is of central importance for many visual recognition tasks, and Resnets are notable for being able to achieve this at a lower complexity than other architectures. They do so by using special skip connections (see Figure 3), which also helps solve the degradation problem associated with networks with high layer-depth, in which accuracy gets saturated and degrades rapidly. Other notable features of Resnets include the heavy use of batch normalization.

We experiment with this ResNet Architecture as our base, and add on our own prediction layer at the end.

4.3. InceptionNet

Inception Networks were developed by Szegedy et al. [24]. These networks draw their power from being able to better utilize computing resources, allowing researchers to increase the depth and width of the network within the computing contraints. Such a model consists of Inception

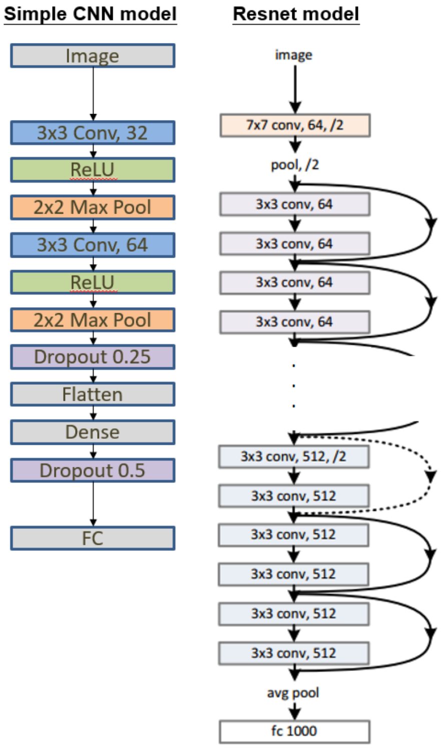
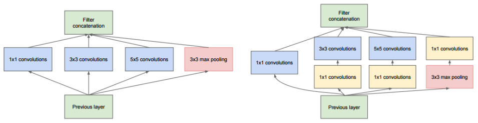


Figure 3. CNN Architectures. (Left) A simple CNN we used. The 2 convolution layers each used 32 and 64 filters of size 3x3. The size of the pooling filter is 2x2. Dropout was applied twice, with $p = 0.25$ and 0.5. (Right) A Resnet architecture with 34 convolution layers, partially shown. The architecture features skip connections.



(a) Inception module, naïve version

(b) Inception module with dimension reductions

Figure 4. Inception Module Architecture

Modules, stacked on each other. An example of the Inception Module can be found in the original paper, and is reproduced here in Figure 4.

In our paper, we use this Inception Network Architecture as our base, and add on Global Average Pooling, Dropout, and Dense Layers to make the prediction specific to our task.

4.4. VGGNet + Transfer Learning

Transfer learning is a learning process in which the domains, tasks, and distributions used in training and testing are different. [19] Knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data-labeling efforts. We start with a CNN model called VGGNet that was pre-trained on the ImageNet dataset and fine-tune it by training it further using satellite images of the Amazon (i.e. transfer learning). A VGGNet is a CNN with 16 layers. [23] ImageNet is an object classification image dataset of over 14 million images with 1000 class labels that are widely used in computer vision tasks. [22] CNN models trained on the ImageNet dataset are recognized as good generic feature extractors, with low- and mid-level features such as edges and corners that are generalizable to many new tasks. [6, 18] ImageNet data comprises object-centric images while satellite images are from a birds-eye view, and therefore the two datasets have different feature distributions. However, the low-level features from an ImageNet-trained CNN are also present in birdseye view images, so using a pre-trained model facilitates the construction of high-level features for satellite images as well. [10]

4.5. Ensembling

Ensembling is a technique that has proven effective in the field of Image Classification. It relies on the predictive power of multiple models in order to make its final prediction. In the past, this has allowed researchers like Krizhevsky et al. [13] to achieve significant improvements in classification accuracy on the ImageNet Dataset. We similarly apply this technique to our problem, collating the different predictions from our different models to make a final prediction.

4.6. Data Augmentation

In order to provide some regularization to the model, and also generate more data, we applied data augmentation to our model. Among the things we experimented with are:

- Horizontal Flips
- Vertical Flips
- Random Rotations within 30 degrees
- Random Zoom

Given that the satellite chips provided a top-down view of the Amazon, these augmentations made sense as they all generated likely perspectives of the subject matter.

4.7. F2 Evaluation

We used the F2 score on the validation set to evaluate our model. The F2 score is based on the F score, which is commonly used in information retrieval [25] and measures accuracy using the precision p and recall r, given by:

$$p = \frac{tp}{tp + fp}, \quad r = \frac{tp}{tp + fn} \tag{5}$$

Essentially, the precision is penalized if the classifier predicts an image to contain a label when in fact it did not, whereas the recall is penalized if the classifier fails to predict that an image contains a label when in fact it actually did. Both metrics are important as we wanted the classifier to sieve out images with labels of interest - key to detection of deforestation - and yet not do it over-zealously.

The actual F2 score is given by:

$$F2 = \frac{5pr}{4p + r} \tag{6}$$

and it weighs recall more heavily than precision. This reflected the fact that the cost of a false negative (failure to detect deforestation) was higher than the cost of a false positive (waste of manpower resources to investigate a potential site).

4.8. Training the CNN

In training our CNN models, we used Keras, with the following hyperparameters.

Hyperparameter	Value
Epochs	20
Optimizer	Adam
Loss	Binary Cross Entropy
Progress Metric	Validation Accuracy
Early Stopping	10 Epochs of no improvement

Table 1. Hyperparameters

We note that as F2 is an aggregate measure, we were not able to directly optimize for it. Instead, we opted to optimize for validation accuracy, which served as a proxy for how well the model was doing.

5. Results and Discussion

5.1. F2 Scores for Each Model

Model	F2
ResNet50	0.89
InceptionNet	0.85
VGGNet	0.89
Simple ConvNet	0.75
Ensembling	0.89

Table 2. Results

We can see that with 20 epochs, ResNet and VGGNet are able to achieve the best F2 Scores, of 0.89. Ensembling was not able to provide significant improvement, although it did manage to match the best result that we achieved. The Simple Convolutional Net that we implemented performed the worst, which showed that the deeper and wider architectures were necessary in order to perform well in this task.

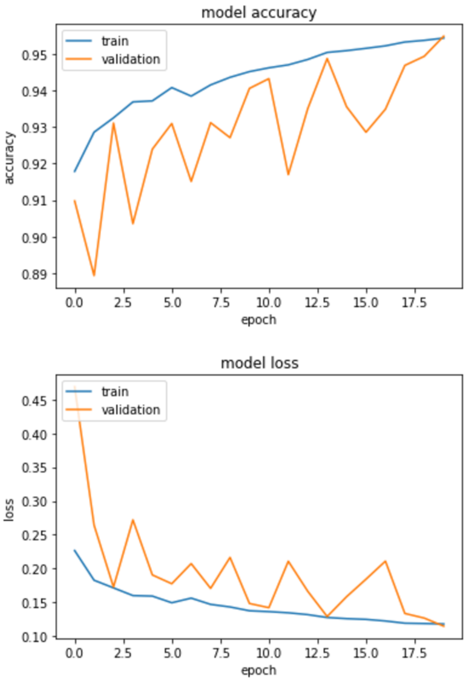


Figure 5. ResNet

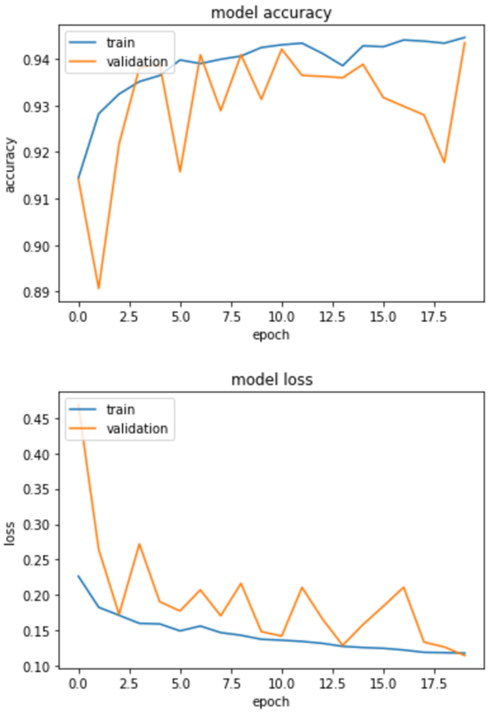


Figure 6. Inception

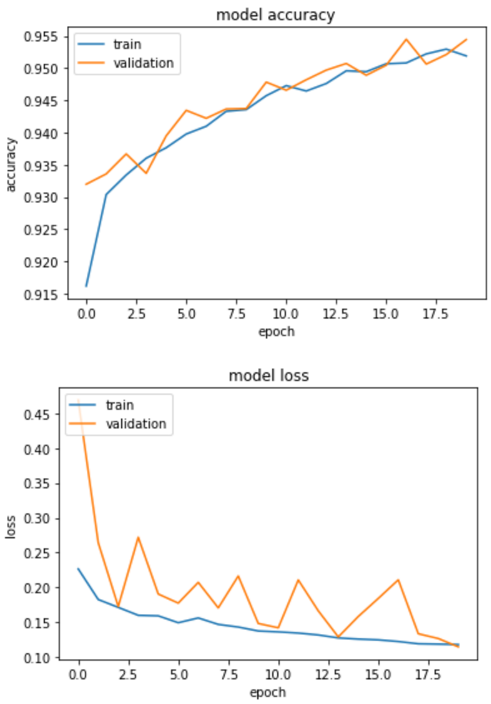


Figure 7. VGG

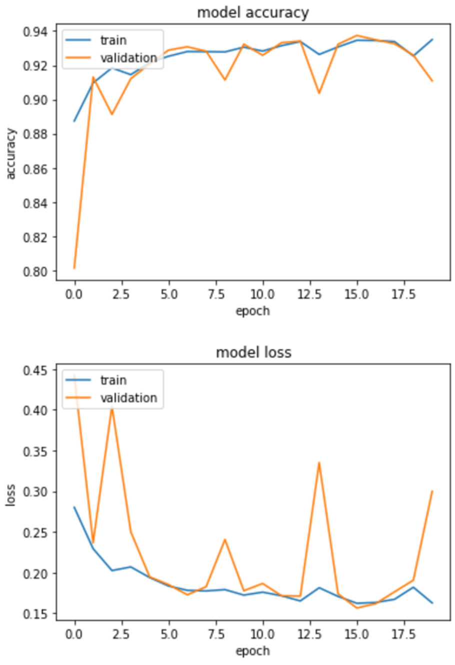


Figure 8. Simple Conv Model

5.2. Monitoring the Learning Process

In order to monitor our training process, we observed the accuracy and loss across each epochs for all of our models. As a sanity check, we saw that the losses went down for all models. Of particular interest is that the validation accuracy of VGGNet 7 tended to track the training accuracy more, and at several times, also beat the training accuracy. In contrast, the accuracy for the ResNet Model 5 and the Inception Model 6 tended to fluctuate a lot more. This may be because we used the pretrained ImageNet weights for the VGG Model, while we used random initialization for the ResNet and Inception Models. The pretrained ImageNet layers, especially the earlier ones, would have learned good representations for image features already, and thus would have been able to provide better validation accuracy.

One other point is that the accuracy for the first 3 models (ResNet, Inception, and VGG) still seems to be improving, and thus training on more epochs may be useful. Unfortunately, due to time and resource constraints, we were only able to train each model for 20 epochs. We do note that the Simple Convolution Model may have reached its limit, with the accuracy appearing to have plateaued towards the later epochs. This is again a reflection of how the ResNet, Inception and VGG architectures are able to learn more due to their advanced features and deeper/wider architectures.

5.3. Confusion Matrices

In order to better understand where our model was failing, we examined the confusion matrices for each of the labels. 3 of the more problematic labels are agriculture in Fig. 9, clear in Fig. 10, and road in Fig. 11.

We note that for agriculture, we are only getting about 75% of the true labels correct, although we are getting most of the false labels correct. This suggests that the model is predicting false too often, and we may need to provide it with more positive examples in order for it to learn how to better recognize true positives. Similarly, the ”clear”, and ”road” labels exhibit this problem, with only 74% and 78% respectively of the true labels being predicted correctly.

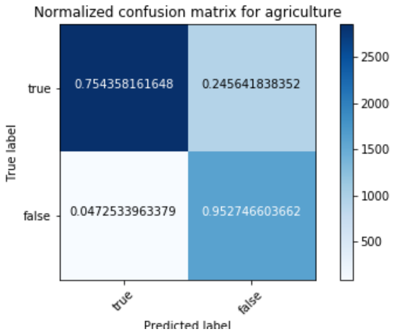


Figure 9. Confusion Matrix for Agriculture

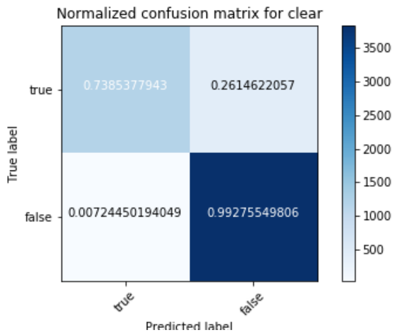


Figure 10. Confusion Matrix for Clear

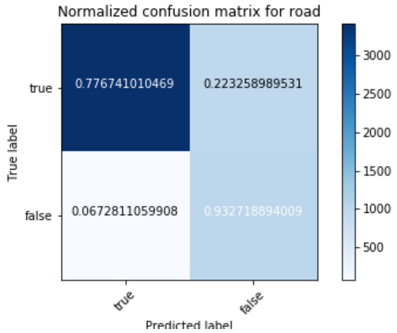


Figure 11. Confusion Matrix for Road

6. Conclusion and Future Work

In our paper, we see that Convolutional Nets are a suitable approach for classifying land use and atmospheric conditions in the Amazon Rainforest, and are able to perform well in a multi-label setting.

Having experimented with multiple models, we see that having deeper and wider networks are critical in achieving a good F2 Score, with our ResNet, Inception and VGGNet models performing much better than a simple convolutional model. However, with these models, it may be necessary to train them for more epochs in order for them to fully converge. Ensembling in our case was unable to provide much of a performance boost, although that may change after we improve each of the models. We may also want to weight the predictions from different models differently.

Lastly, we find that while our model is able to perform well on certain labels, it struggles with labels like ”road”, ”agriculture” and ”clear”. In the future, we could think of better ways for the model to make predictions, perhaps by building separate classifiers specifically for these labels. We also seek to exploit the structure of the data to improve the accuracy of our model, by leveraging on the fact that some of the labels are mutually exclusive, or in order cases, often appear together.

7. Code Citations

We used code from Kaggle User anokas [1] for image data loading as well as the F2 Metric. We also used code from Keras Documentation for Transfer Learning [2].

References

[1] Keras starter code. `https://www.kaggle.com/anokas/simple-keras-starter/comments/code`. Accessed: 2017-06-12.

[2] Transfer learning. `https://keras.io/applications/`. Accessed: 2017-06-12.

[3] Arcgis. Prodes deforestation, 2017. https://www.arcgis.com/home/item.html?id=4160f715e12d46a98c989bdbe7e5f4d6.

[4] W. Cohen, M. Fiorella, J. Gray, E. Helmer, and K. Anderson. An efficient and accurate method for mapping forest clear cuts in the pacific northwest using landsat imagery. *Photogrammetric Engineering and Remote Sensing*, 64(4):293–300, 1998.

[5] E. Diaz. Online deforestation detection, 2017. arXiv:1704.00829.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition, 2013. arXiv:1310.1531.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. arXiv:1512.03385.

[8] T. Hilker, M. A. Wulder, N. C. Coops, J. Linke, G. McDermid, J. G. Masek, F. Gao, and J. C. White. A new data fusion model for high spatial- and temporal-resolution mapping of forest disturbance based on landsat and modis. *Remote Sensing of Environment*, 113(8):1613–1627, 2009.

[9] N. Jean, M. Burke, M. Xie, W. Davis, D. Lobell, and S. Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[10] N. Jean, R. Luo, and J. Kim. Nighttime light predictions from satellite imagery, 2016. http://cs231n.stanford.edu/reports/2016/pdfs/423_Report.pdf.

[11] T. Kehl, V. Todt, M. Veronez, and S. Cazella. Real time deforestation detection using ann and satellite images, 2015. Springer.

[12] S. Kluckner, T. Mauthner, P. Roth, and H. Bischof. Semantic classification in aerial imagery by integrating appearance and height information. *Asian Conference on Computer Vision, Lecture Notes in Computer Science*, 5995:477–488, 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] M.C.Hansen, D. Roy, E. Lindquist, B. Adusei, C. Justice, and A. Altstatt. A new data fusion model for high spatial- and temporal-resolution mapping of forest disturbance based on landsat and modis. *Remote Sensing of Environment*, 112(5):2495–2513, 2008.

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

[16] NASA. The landsat program, 2017. https://landsat.gsfc.nasa.gov/.

[17] NASA. Moderate resolution imaging spectroradiometer, 2017. https://terra.nasa.gov/about/terra-instruments/modis.

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks, 2014. Proceedings on the 2014 IEEE Conference on Computer Vision and Pattern Recognition.

[19] S. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE*, 22(10):1345–1359, 2010.

[20] G. Popkin. Satellite alerts track deforestation in real time, 2016. http://www.nature.com/news/satellite-alerts-track-deforestation-in-real-time-1.19427.

[21] P. Potapov, M. Hansen, S. Stehman, T. Loveland, and K. Pittman. Combining modis and landsat imagery to estimate and map boreal forest cover loss. *Remote Sensing of Environment*, 112(9):3708–3719, 2008.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014. arXiv:1409.0575.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale visual recognition, 2014. arXiv:1409.1556.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[25] C. J. van Rijsbergen. Information retrieval (2nd ed.), 1979. Butterworth.

[26] Z. Zhu, C. E. Woodcock, and P. Olofsson. Continuous monitoring of forest disturbance using all available landsat imagery. *Remote Sensing of Environment*, 122:75–91, 2012.