Understanding the Amazon from Space with Convolutional Networks

Jeff Pyke
Stanford University
pyke@stanford.edu

Abstract

Satellite data would like to be used to track the human footprint in the Amazon rainforest. This project describes the creation of a convolutional neural network to submit a result to the kaggle competition, 'Understanding the Amazon from Space'. The primary goal is to better track changes in forests with future daily imagery from the Planet constellation. The classifier classifies satellite images of the Amazon rainforest into several classes. Some of the classes differentiate between human causes of forest loss and natural causes, in order to better understand where, how, and why deforestation happens.

1. Introduction

The Amazon Rainforest contains roughly 390 billion trees belonging to 16,000 species. The forest contains over half of the earth's rainforests by land area. The Amazon basin contains 5,500,000 square kilometers of rainforest. However, even with it's massive size, deforestation is a large problem. Because the Amazon basin contains such a large amount of biodiversity, the losses are severe. Traditionally, satellite imagery used to track changes in forests has been provided in coarse resolution. Landsat, a well established satellite imagery program, provides 30 meter pixel images, and MODIS provides 250 meter pixels. In contrast, Planet, the builder of the world's largest constellation of earth imaging satellites, provides 3-5 meter resolution images of earth with their large constellation of CubeSats which were launched in 2016 and 2017. Higher resolution imagery provides a new opportunity to track small scale loss of forest and forest degradation, which were previously not possible with coarse resolution imagery. Forest degradation such as selective logging and small scale mining have only a slight footprint as viewed from above, but are strong indicators of the health of a forest. In addition, the prior two activities are often carried out illegally. Better information about where and how these types of deforestation happen provide better understanding of how the global community can best respond. Because this higher resolution imagery is relatively unprecedented, new algorithms are needed to process the data that is retrieved and help organizations like Planet gain insights from their imagery. This paper seeks to provide a useful algorithm for class discrimination of Planet imagery.

1.1. Competition

Kaggle is hosting a satellite image classification competition to detect small scale deforestation or forest degradation. The competition also seeks to differentiate between natural and human causes of forest loss. The dataset is higher resolution than those typically used from Landsat or MODIS. Details can be found here: `https://www.kaggle.com/c/planet-understanding-the-amazon-from-space`

Submissions to the competition will be evaluated on their mean F2 score, which measures accuracy taking both precision and recall into account. This is shown below in equation (1).

$$(1+\beta^2)\frac{pr}{\beta^2 p + r} \text{ where } p = \frac{tp}{tp+fp},\ r = \frac{tp}{tp+fn},\ \beta = 2. \tag{1}$$

Note that F2 weights recall higher than precision, and that the mean F2 score is equivalent to the averaging the individual F2 score for each test image.

Submissions are submitted in CSV format, with each row containing a test image name and the corresponding classes that apply (there are 17 total classes).

1st, 2nd, and 3rd place will receive prizes of $30,000, $20,000, and $10,000 respectively.

The entry deadline is July 13th, 2017, and the final submission deadline is July 20th, 2017.

1.2. Background/Related Work

In 2014, a group used a deep convolutional neural network to detect small objects such as vehicles in satellite imagery. They extended a traditional DNN to have variable receptive field output sizes, which improved their model. [4] In 2010, a group used neural networks to detct roads in

aerial imagery. [3] In 2017, a group from MIT used convolutional networks to analyze land use in urban neighborhoods. [1]

2. Dataset

This competition uses data from Planet, a company which designs and built the largest constellation of Earth-imaging satellites. The satellite "chips" (image segments) for this competition use 4-band GeoTiff format satellite images in sun synchronous orbit (SSO) and International Space Station (ISS) orbit. They contain red, green, blue, and near infrared bands. Each channel is in a 16-bit format. The imagery has an orthorectified pixel size of 3m. All geotiff information about location has been removed from the chips for the purposes of the competition. The data is available in both tif and jpg format. The JPG chips were produced for reference and practice, by the Planet visual product processor. They refer to the same scene content. The chips are labeled reasonably well. The labels fit into three groups: atmospheric conditions, common land cover/use occurrences, and rare land cover/use occurrences. Sometimes multiple labels occur in a single image, with some exceptions. The most common labels in the dataset are rainforest, agriculture, rivers, towns, and roads. The least common labels are slash and burn, selective logging, blooming, conventional mining, artisinal mining, and blow down. The tif train set contains 12.87 GB of images and the test set contains 19.45 GB. There are 40479 labeled train images. The expected result is a set of labels for each of the 61191 test set images (csv format). The submission will be evaluated on it's mean F2 score, which is a function of precision and recall. Recall is weighted higher than precision.

2.1. Labels

The data was created by first manually collecting an initial set of scenes, with images that had complete four band product. They divided the initial set of 150,000 chips into two sets, "hard" and "easy". Where hard contains the less common scenes such as slash and burn, selective logging, blooming, conventional mining, artisinal mining, and blow down, and easy contains the more common scenes, such as rainforest, agriculture, rivers, towns, and roads. The chips were labeled using crowdsourcing, on the Crowd Flower platform. Planet acknowledges the fact that even groups of experts cannot always agree exactly on what is present in a given image. Furthermore because the labeling was done with crowdsourcing, there is likely even more noise in the labels. They chose not to use ground truth data to label because it would be too time consuming and costly. Planet also felt it was more useful to provide a large crowdsourced data set as opposed to a smaller, more definitive dataset.

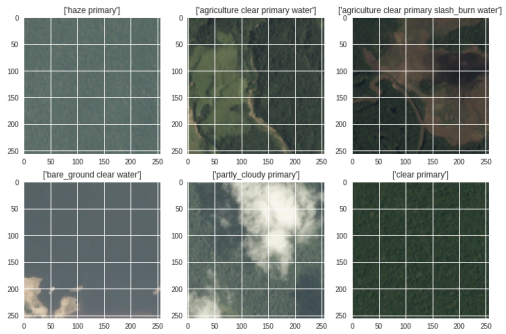

Figure 1. Some example training images

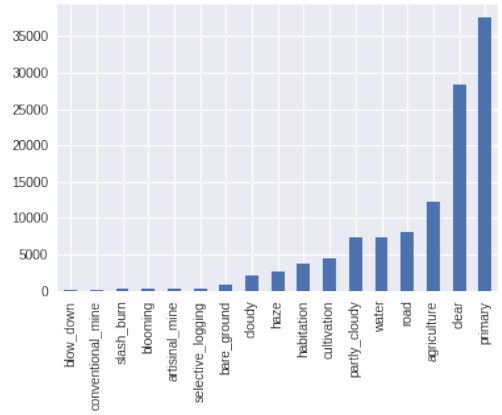

Figure 2. Histogram of label frequency in the training data

3. Exploring the Data

First, some time was spent exploring the data using kaggle provided examples. Heatmap graphs were generated using the python visualization library, seaborn. These show the spread of labels within the data. There are a total of 116,278 tags on the images. And an average number of 2.87 tags per image. It is interesting to note that the vast majority of the labels fall under a couple of labels.

Figure 1 shows the layout of the data. Some classes appear much more frequently in the data than others. In figure 2, it is clear that many of the classes overlap. Figure 3 shows that only one weather label will apply to each image at most. In figure 4, it is shown that multiple labels may apply to land features.

4. Approach

First I created a baseline by computing the F2 score in the case that each scene is labeled clear primary. In the training set, 37,513 images are labeled as primary, or 92.7%. Clear appears in 28,431, or 70.2% of the images. This model scored 0.64640 on the test set. This is a relatively poor score and sets a baseline for how any subsequent

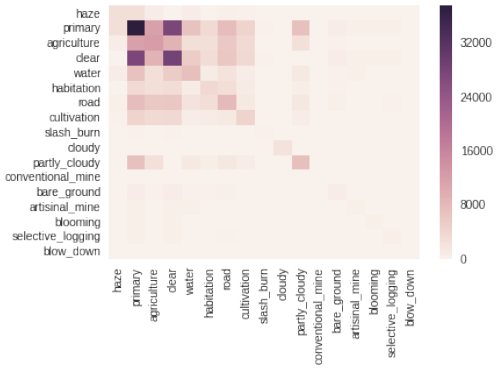

Figure 3. Co-occurance matrix of training data

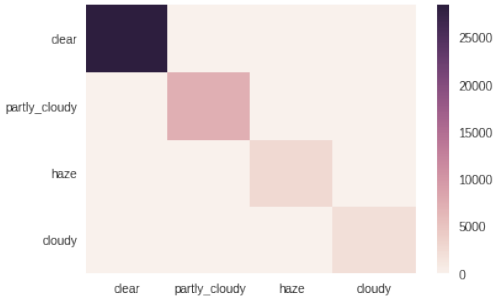

Figure 4. Co-occurance matrix of weather labels

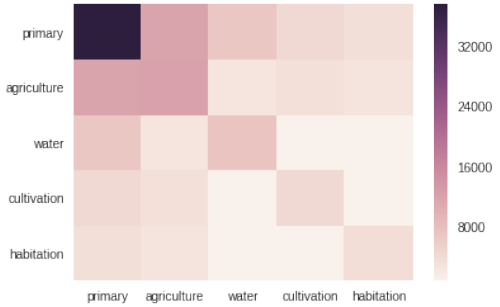

Figure 5. Co-occurance matrix of land labels

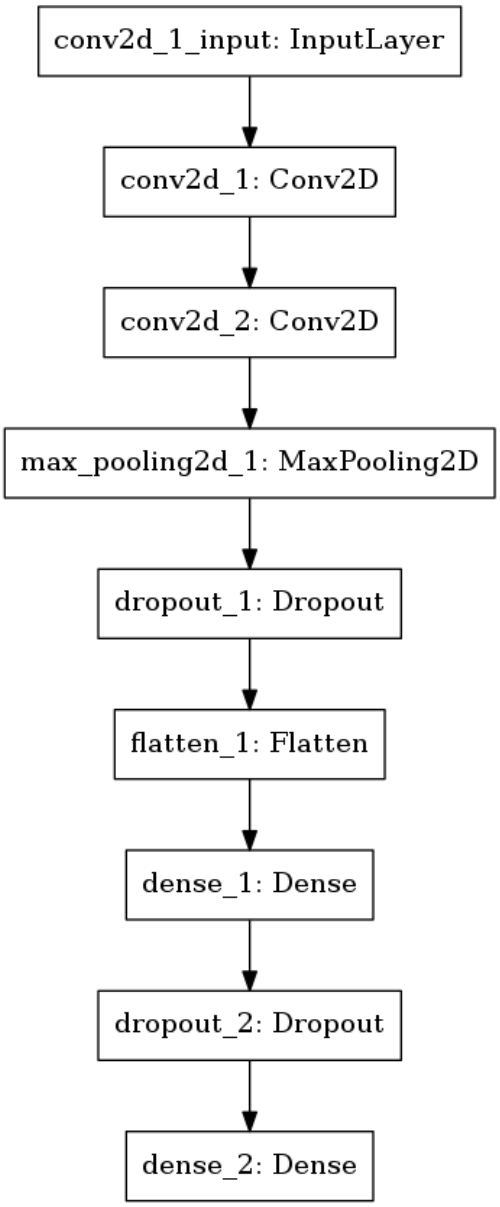

Figure 6. Network topology, minus extra conv and pooling layer

submission should score.

Next, I configured a convolutional network trained from scratch, using keras. Keras is a high level API which is built on top of tensorflow. The basic framework for the neural net is based on a starter kernel provided by the kaggle user anokas. [2] The network is designed as follows: Conv ReLu Layer with 32 3x3 filters, Conv ReLu with 64 filters, 2x2 Max Pool, Dropout, Fully connected ReLu layer, Dropout, Fully Connected layer. A figure of the network topology is included. The NN uses binary cross-entropy loss with the adam optimizer.

5. Experiments

Several small experiments were used to improve the performance of the network topology. Both 32x32 downscaled images and 64x64 images were used. Batch normalization did not seem to improve the score. Using lower levels of dropout was less successful. The most successful dropout levels were .25 after max pooling and .5 after the first dense layer. A reduced number of filters in the convolution layers seemed to perform more poorly (32 vs 64 filters in the first layer, and 64 vs 128 filters in the second layer). When moving from 32 and 64 filters to 64 and 128, the F2 score

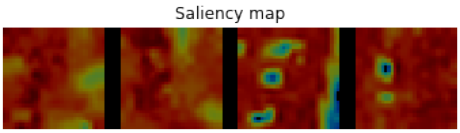

Figure 7. Cam visualization

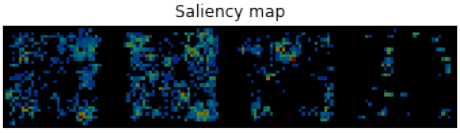

Figure 8. Saliency map visualization

improved from 0.825 to 0.83 on the validation set. As a result of running the above described neural network on a split of 35,000 train / 5479 validation samples over 4 epochs, an F2 score of 0.834176169932 was achieved.

After generating p values for each of the classes, a threshold of 0.2 was initially used. Later, I programmatically found the best cutoff for each class to maximize the F2 score on the validation set. Interestingly, the less common classes generally had very low cutoffs (such as 0.02 for slash burn, 0.03 for artisinal mine, 0.05 for selective logging). The more common classes had higher cutoffs such as 0.27 for primary, 0.17 for clear, and 0.18 for haze, cultivation, and partly cloudy. This improved the F2 score on the validation set to 0.846. Next, I added another set of convolutional layers followed by max pooling. This improved the F2 score to 0.869. Adding a third layer of conv and max pooling did not raise the score. The best score achieved on the test set (uploaded to Kaggle) was 0.86548.

Using the keras-vis library, I was able to generate saliency and cam visualizations for the network on a couple of example images. The results are shown in the figure below.

6. Conclusion

A moderately successful approach was shown. A deep convolutional neural network trained from scratch was able to achieve an F2 score of 0.84 on the dataset, scoring approximately 320th of 428 participating teams. The model is far from perfect, but it was interesting to explore what did and did not work on this type of dataset. I learned that the number of layers is very important, as are the thresholds in the final layer. And, certainly the input size can make a large difference on training time (as can use of GPU code). Additional work could also test on the tif set. However, it's likely that the labels were done on the jpg set, so it's unclear if this would help. Further work could include using VGGnet or Resnet transfer learning, tuning hyperparameters further, and exploring other types of networks. It's quite possible that it would be easier to achieve a better score when using a network like this that is pretrained on millions of regular images.

References

[1] M. G. Adrian Albert, Jasleen Kaur. Using convolutional networks and satellite imagery to identify paerns in urban environments at a large scale, 2017.

[2] anokas. Simple keras starter.

[3] G. E. H. Volodymyr Mnih. Learning to detect roads in high-resolution aerial images, 2010.

[4] C.-L. L. C.-H. P. Xueyun Chen, Shiming Xiang. Vehicle detection in satellite images by hybrid deep convolutional neural networks, 2014.