Predicting Land Use and Atmospheric Conditions from Amazon Rainforest Satellite Imagery

Joe Johnson, Rohisha Adke

June 2017

Problem Statement

Governments and local stakeholders do not have enough information to diagnose and address deforestation in the Amazon basin.

We seek to arm officials with information by identifying land use occurring across the Amazon basin from satellite imagery. In addition, we identify atmospheric conditions and type of land cover in each image.

Existing Research

CNN architectures have been used for land use classification on DeepSat (Basu, 2015) and the UC Merced Land Use (Yang, 2010) dataset.

Transfer learning has also been used for land use classification:

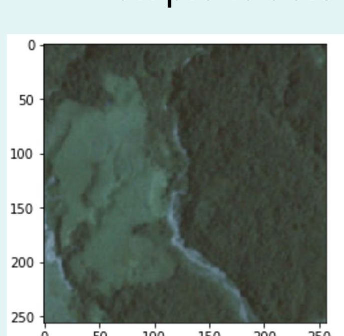
- Trained GoogLeNet + fine-tuning on UC Merced Land Use, Brazilian Coffee Scenes datasets: 97.10% accuracy (Castelluccio, 2015)
- Trained Overfeat (based on AlexNet) + custom CNN component to classify images in the UC Merced Land Use dataset accuracy of 92.4% (Marmanis, 2016)
- AlexNet and VGGNet: > 99.9% accuracy on DeepSAT (Papadomanolaki, 2016)
- DCNN based on Inception modules (inspiration from GoogLeNet), hyperparameters chosen using genetic algorithm: 98.4% accuracy on SAT-4, 96.0% on SAT-6 (Ma, 2016)

SatCNN (Zhong, 2017) is a CNN architecture specifically for high-spatial-resolution remote-sensing images.

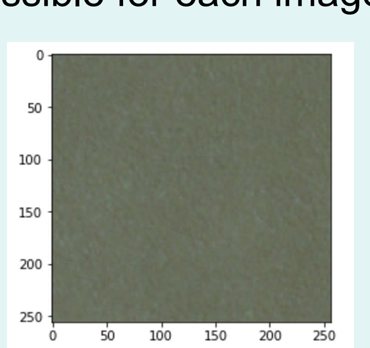
- Deeper convolutional layers with smaller filters
- 99.65% and 99.54% accuracies on DeepSat [8]

Dataset

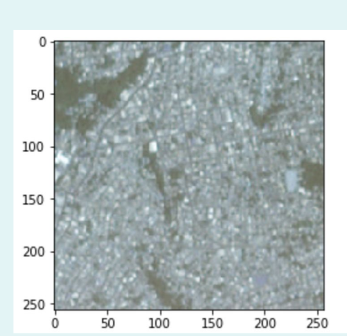
- 150,000 256x256 pixel satellite images of Amazon Rainforest
- 17 labels describing atmospheric condition (clear, cloudy, etc.), land use, and forest cover (primary forest, agriculture, etc.)
- Multiple labels possible for each image



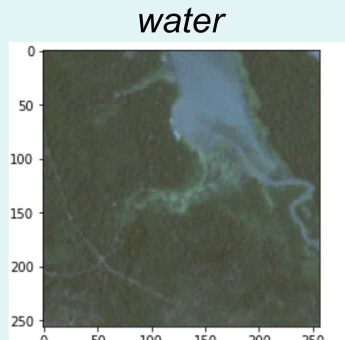
clear primary road selective_logging water



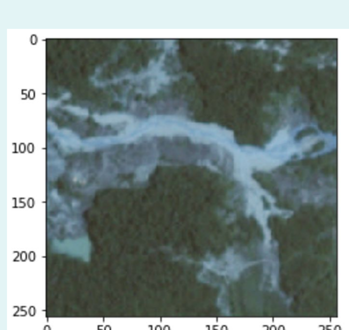
haze primary



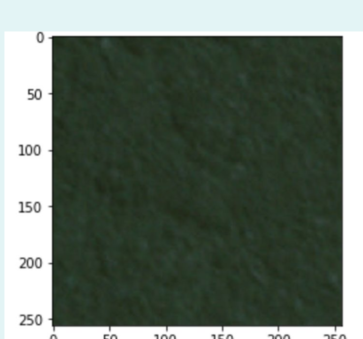
clear habitation primary road



agriculture clear primary water

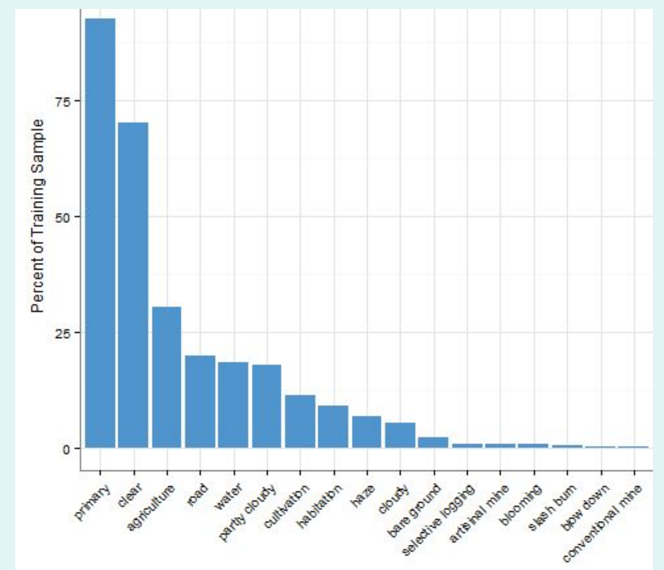


artisinal_mine clear primary



clear primary

Highly imbalanced label distribution



Evaluation Metric

For imbalanced data it is easy to get high accuracy. Instead of accuracy we evaluate with F2 score =

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

r =recall, p =precision, t =true positive, f =false positive

F2 score weights recall higher than precision

Baseline CNN

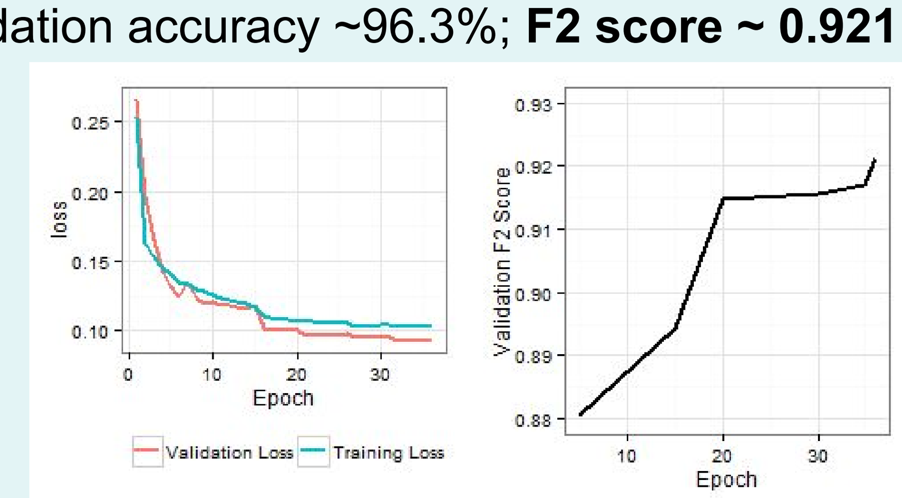
CNN with 4 convolutional layers

validation accuracy ~ 94.43%

F2 score ~ 0.86

Deeper CNN - 8 layers

Deeper CNN model with 8 convolutional layers, larger filters, learning rate decay, and early stopping rule validation accuracy ~96.3%; **F2 score ~ 0.921**



ResNet-18

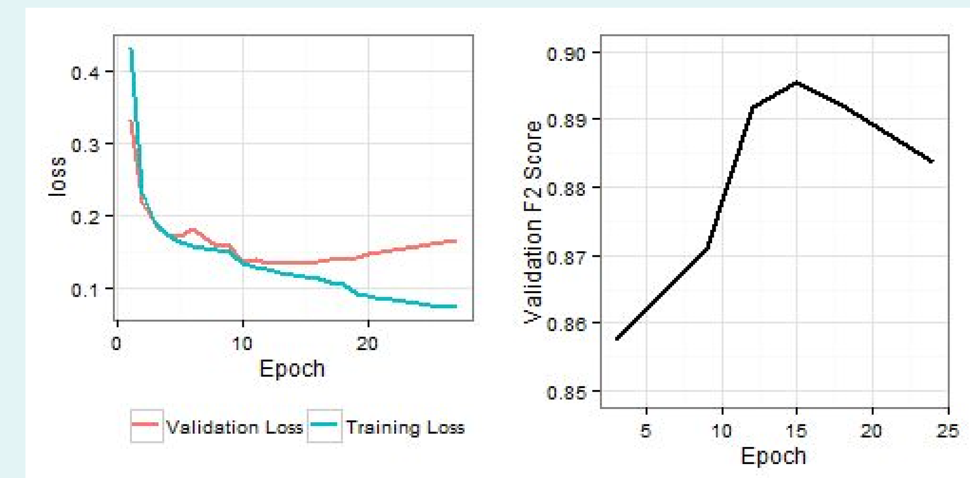
State of the art model. Allows us to train deeper neural network models than standard CNNs by implementing bottleneck layers. This allows the gradient to flow backwards through very deep networks.

Our model has 18 layers

validation accuracy ~ 95.44%

F2 score ~ 0.895

ResNet-18 (continued)



VGGNet-16 Transfer Learning

Tried various architectures involving transfer learning:

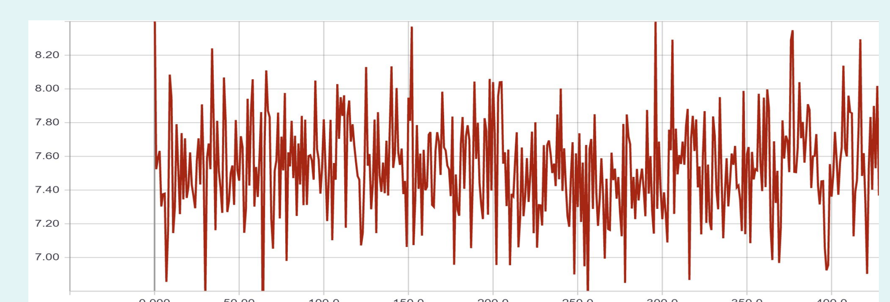
1. VGGNet with final fully-connected layer replaced with layer to our 17 classes - trained last layer, then entire network for a few epochs

F2 score ~0.55-0.65

2. VGGNet with final two layers replaced with fully-connected layers, along with two additional fully-connected layers of decreasing size - trained four layers, then entire network for a few epochs

F2 score ~0.64 - 0.69

Loss vs. Iteration



3. VGGNet with final two layers removed, extra layers added: fully-connected 4096-unit layer, batch normalization, convolutional layer with 32 5x5 filters, leaky relu activation, max pool, convolutional layer with 64 5x5 filters and leaky relu, max pool, fully connected layer, then fully connected layer with sigmoid activation

F2 score ~0.66-0.71

Loss vs. Iteration

Analysis

- Our best performing model was the CNN with 8 layers
  - Proper network training through early stopping, hyperparameter search, and learning rate adjustment make big improvements
- Relatively simple model outperformed more complex models like ResNet, and transfer learning
- Complex models may not be necessary because we have a smaller classification problem
  - Only 17 classes
  - Within-class images that are very similar to one another
  - Images that often do not have distinct features in different parts of the image
- Complex models may in fact hurt performance because the models are too flexible, and learn noise in images if not trained for enough time
- Transfer learning may not perform well because data set is dissimilar from ImageNet images VGGNet was trained on. Thus, doing transfer learning requires training for many epochs, or adding additional layers on top of the preexisting net and training for many epochs

Future Work

In future work we plan to explore feature augmentation to increase training data size through:

- Rotating images
- Image cropping
- Efficient data-loading/memory management

Continuing to optimize hyperparameters, including logit score probability threshold.

Adding best performing CNN model as component after VGGNet.