Understanding the Amazon from Space

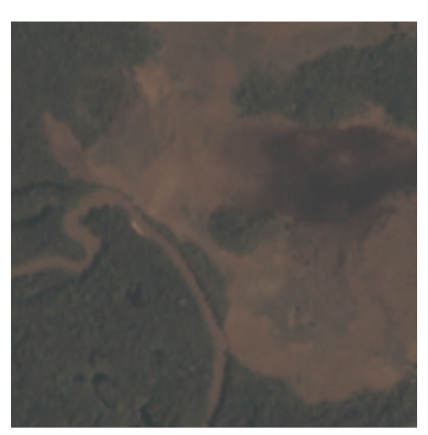
Brent D. Lunghino (lunghino@stanford.edu), Brian A. Rohr (brohr@stanford.edu), Vishal Subbiah (svishal@stanford.edu)

Background and Motivation

The Amazon Rainforest is extremely important to preserve due to its unmatched biodiversity and great capacity to uptake carbon dioxide[1], but it is difficult to make a case for the urgency of its preservation without first quantifying the rate of its destruction. One good way to to track land use over time in the Amazon is to assign land use labels to satellite images of the entire Amazon taken at different points in time. Manually analyzing images of the entire Amazon Rainforest (a total of about 3 million images) is an intractably large task to complete once, and it would need to be completed many times in order to compile temporal land use data. For that reason, we have trained a computational model that can rapidly process satellite images of the Amazon and assign labels that describe land use and atmospheric conditions. With this rapid, automatic image processing technique, temporal land use data will be easily assembled, providing the data necessary to make a case for protecting the Amazon Rainforest.

Problem Statement

The goal is to train a model that can take a satellite image of the Amazon as the input and output whether or not the image has each of 17 labels. Two training examples are below:



agriculture, clear, primary forest, slash and burn, water

conventional mine, partly cloudy, primary forest

Image Data Details

The data is provided by Planet for a Kaggle competition. The training data set is 40,000 images. Each image is 256x256 pixels with 4 bands (RGB + near-infrared). Each image gives a resolution of 3.7 meters per pixel, and the dataset covers 30 million hectares of Amazon Rainforest. Each image has been manually labeled as either having or not having each of 17 labels.

Methods

Data Augmentation: Before each image is fed to the model, it is randomly flipped and rotated then randomly cropped to 224x224.

Model Architectures: We have trained a 6-layer VGG-Net[2] as well as pretrained Resnets[3] with 18, 50, and 101 layers.

Learning Rate Decay: The learning rate is lowered if several mini-batches elapse with no improvement in loss.

Cutoff Optimization: For each image, the model outputs a score between 0 and 1 for each label indicating how strongly it predicts that the label should be assigned to the image. We choose the label assignment cutoff value for each label that maximizes the F2 score on the validation set.

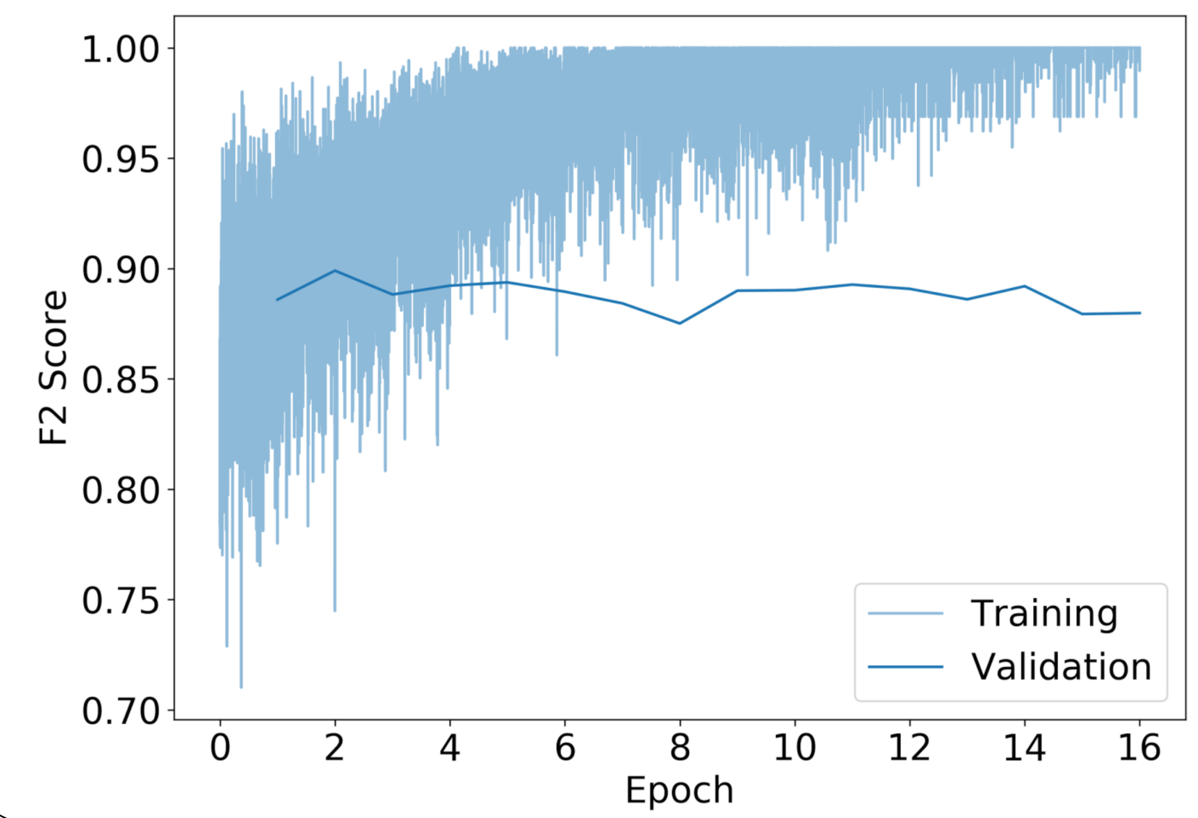
Weighted Ensembling: We trained 3 Res-Nets on the RGB data, IR data, and gradient of the RGB data. The final output was calculated as a weighted sum of the three. The weights were optimized using particle swarm optimization.

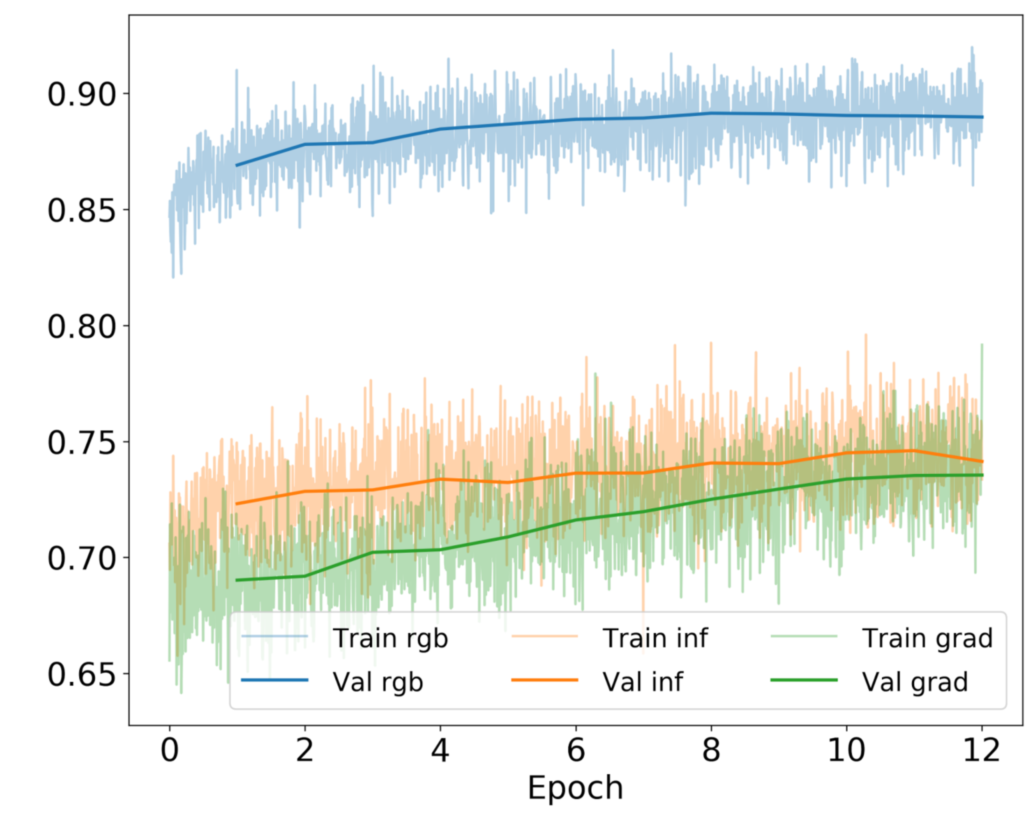
Batch Balancing: During training, each batch of training examples has an equal number of images with each of the 17 labels.

Results and Discussion

Model Name	Validation F2	Test F2	Number of Epochs	Data Augmentation	Batch Balancing
Basic Conv 6-layer	0.830	0.831	10	no	no
Basic Resnet 18	0.887		8	no	no
Basic Resnet 101	0.880		16	no	no
Triple Resnet 18 Tuned F2		0.917	12	yes	no
Triple Resnet 18 Tuned F2 and Model Weights	0.950	0.918	12	yes	no

The table above summarizes the properties and performance of a representative set of our models. The figure on the bottom left shows the learning curve for the basic Resnet (101-layer). To train this model, we fine-tuned a pretrained (on ImageNet) Resnet on the RGB images, but the model suffered from overfitting: the validation accuracy was much lower than the training accuracy. To overcome this, we introduced data augmentation, used a shallower Resnet (18-layer), and used weighted ensembling. The learning curve for this model is shown in the figure on the bottom right. Clearly, the model is not overfit: the training and validation accuracies are nearly identical. After cutoff optimization, this model gives our highest F2 score: 0.917. We tried to further improve on this using batch balancing, but this gave a lower F2 score, likely due to over-predicting rare labels.





Conclusions and Future Work

- Our best model has an F2 score of 0.917, which is about 0.015 lower than the best model in the Kaggle competition.
- To further improve on this result, we plan to:
  - Combine the ensembles by stacking the outputs of the last affine layer and adding one additional affine layer.
  - Train Resnets using Xavier initialization.

References

1. About the amazon. World Wildlife Fund, 2017.
2. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.
3. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015.