The Amazon Rainforest is subject to aggressive deforestation. Automatically identifying areas of human encroachment could aid in conservation. We built a family of neural network classifiers which input a satellite image and output 17 distinct labels per image. We achieved an F2 score of 89% using transfer learning (Resnet) on a labeled dataset. Models were implemented in PyTorch and run on a Google Cloud GPU instance.

The dataset consists of over 110K 256 x 256 images, augmented during training with random jitter, cropping, scaling, and transposing. The most common label is “primary rainforest”, followed by “water”, “habitation”, “cultivation”, and “cloudy”. 40K images for training; 60K images for testing. We achieved an F2 score of 89% using transfer learning (Resnet) on a labeled dataset.

Baseline CNN
- 3 layer sequential Conv-net + 2 FC layers
- Adam Optimizer ($\gamma=0.9$, $\beta=0.999$)

U-Net [2]
- Recursive convolutional neural network.
- Double-conv blocks that are successively downsampled and then recombined through upsampling.
- Observed to perform well on image segmentation.
- SGD with Momentum ($\alpha=0.9$)

Pyramid Net [4,5]
- Similar recursive structure to U-Net
- Uses residual connections in addition to max pooling on the recursive “down”.
- All spatial levels are fed to aggregation and prediction layers.
- SGD with Momentum ($\alpha=0.9$)

Above neural models trained with multi-label soft-margin loss and a sigmoidal predictions.

\[
\text{loss}(x, y) = -\frac{1}{K} \sum_{k=1}^{K} \left( y_k \log\left( \frac{e^{x_k}}{1+e^{x_k}} \right) + (1 - y_k) \log\left( \frac{1}{1+e^{x_k}} \right) \right)
\]

VGG16 [3], trained last two FC layers
ResNet50 [4], trained last Conv block + FC
ResNet50, trained last Conv block + FC
ResNet50 and last two Conv blocks + FC

Transfer learning performs the best amongst all approaches. Specifically, ResNet50 performed best. Recursive neural networks seem to perform well on possibly related tasks (i.e. image segmentation). However, a standard U-Net implementation was not as performant as a U-Net with modified residual connections such as Pyramid Net.

XGBoost is a formidable baseline given the simplicity of feature extraction and training.

The imbalanced dataset caused per-class sensitivity to vary significantly.

Using SGD-Momentum ($\alpha=0.9$) was crucial to lowering loss below 0.2. LR Decay with Adam halted progress for early results.

Per-class loss weighting did not improve performance on unbalanced dataset.

Upsampling minority classes using a Least Squares Projection caused immediate overfitting that couldn’t be solved with dropout.

Modifying the loss function to include Jaccard distance seems to have been used effectively in other cases, but requires more tuning.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>445,560</td>
</tr>
<tr>
<td>XGBoost</td>
<td>4,452,384</td>
</tr>
<tr>
<td>Pyramid Net (ResNet)</td>
<td>4,452,384</td>
</tr>
<tr>
<td>ResNet50</td>
<td>4,452,384</td>
</tr>
<tr>
<td>U-Net</td>
<td>4,452,384</td>
</tr>
</tbody>
</table>

REFERENCES
[3] Very Deep Convolutional Networks for Large-Scale Image Recognition
- Simonyan, K.
- Zisserman, A.


DISCUSSION
- Train multiple sub-networks that specialize for label type.
- Condition neural architectures on statistical features.
- Ensemble all trained models.
- Train models on TIF infrared channel data.
- As an experiment, feed Resnet features into XGBoost
- Train models on entire dataset and compete on Kaggle
- Continue to tune hyperparameters, new LR schedules