Capstone 2- Amazon Image Classification

Isadora P. Thisted

Springboard

August 5th, 2020

Introduction

The Amazon is the largest surviving rainforest ecosystem in the world and currently home to 10% the world's species[1]. It plays an important role in regulating rainfall cycles in South America[2] and in thermo-regulating the atmosphere above the region. While the full extent of the forest's role on the global climate is not fully understood according to one modeling study its destruction could have ripple effects in as far-reaching places as the Sierra Nevada mountains in California.[1]

And while complete deforestation would certainly have devastating effects in many fronts (biodiversity, habitat loss, and climate change to name a few) by some projections, a loss of as little as 20 to 25 percent of original forestland could tip the entire system into an unstoppable transition to a drier, savanna-like ecosystem.[3] The WWF estimates that 27 percent of the Amazon biome will be without trees by 2030 if the current rate of deforestation continues.

Climate change, wildfires, cattle ranching, mining, logging, and farming all contribute to deforestation compounded by ineffective or counterproductive government policies and lack of resources that have left many of these activities unchecked. Better data about the location of deforestation and human encroachment on forests can help governments and local stakeholders to respond more quickly and effectively.

Satellite imagery with 3 to 5-meter resolution of the Amazon basin was made available on Kaggle by Planet, designer and builder of the world's largest constellation of earth-imaging satellites. The images were manually labeled and collected by Planet's Flock 2 satellites between January 1, 2016 and February 1, 2017 and cover an area of 30,000,000 hectares. Each image chip can contain one or more labels' associated atmospheric conditions as well various classes of land cover and land use. Two versions of each image are available: .jpg and tif, with the latter containing near infra-red data in addition to RGB. For the purposes of this project and because of computational power limitations, only the .jpg images will be utilized.

There are approximately 40,000 16-bit images in the training set, each one has dimensions 256x256.

For this project, we will utilize these images to create a machine learning model capable of identifying the labels associated with each image. The resulting algorithm can allow organizations and the global community to better understand where, how, and why deforestation happens and take proactive steps to prevent it. It can also aid them in identifying areas for active reforestation so as to build a margin of safety.

Data

- **Source**
    - https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data
- **Description**
    - 34GB of images and a file (train_labels.csv) containing image file names and a list of its associated labels
    - 17 possible labels, each image can contain multiple labels
        - Atmospheric conditions: haze, clear, cloudy, partly_cloudy
        - Vegetation and land use: primary, agriculture, water, road, cultivation, habitation, bare_ground, selective_logging, slash_burn, blooming, blow_down, conventional_mine, artisinal_mine
    - 80,000 images (50/50 split for train and test sets), .jpg and .tif files are provided for each image. The .jpg files are 256x256 and contain 3 layers of data (RGB), the .tif files contain an additional layer of near infra-red data
    - From the images provided,
    - On average .jpgs are ~15KB each and .tifs are 538KB each

Note: the image labeling process was done manually through crowdsourcing and as such is likely to contain some mistakes. Even with these mistakes the data has a reasonably high signal to noise ratio and is sufficient for training.

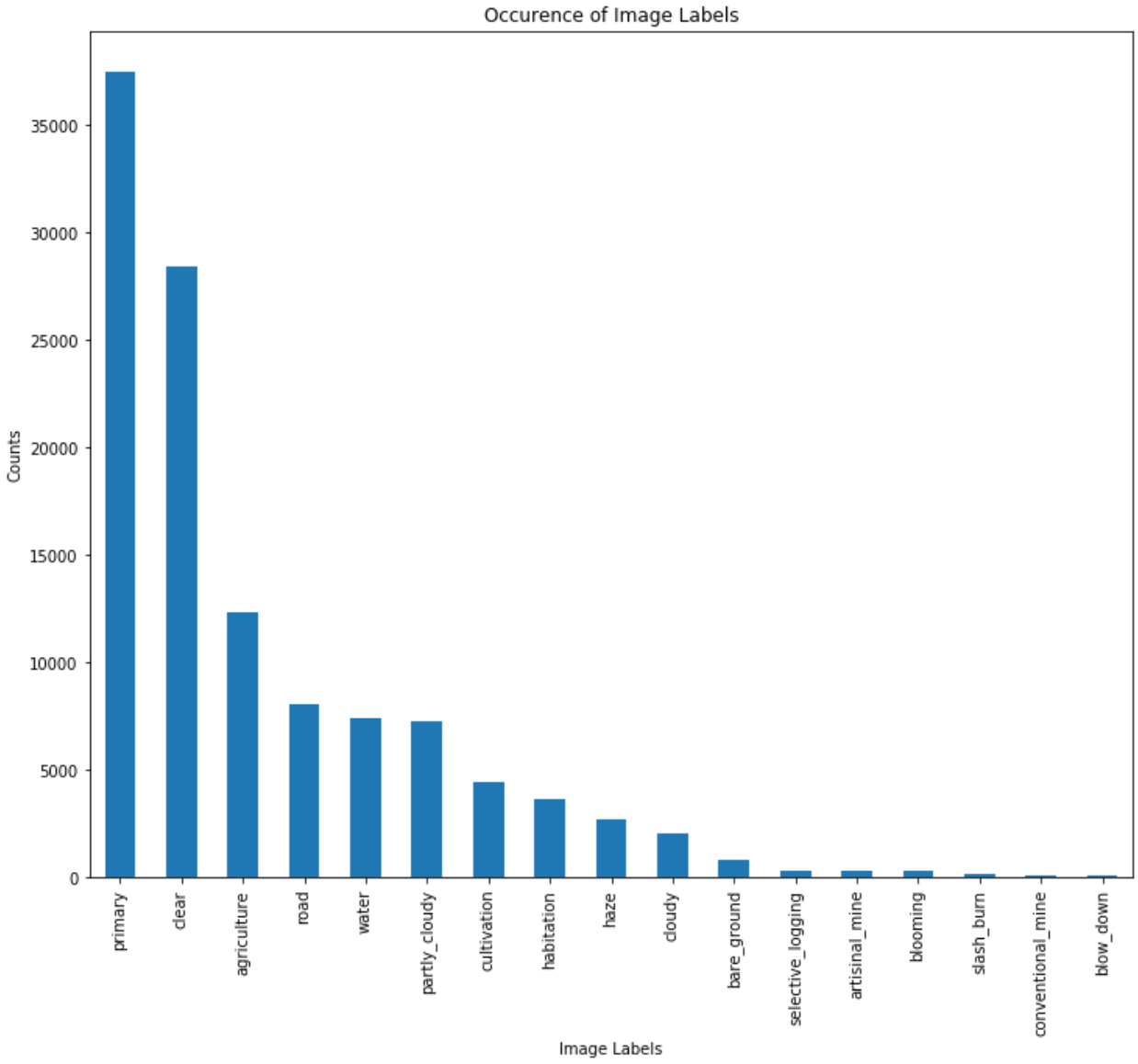
Project deliverables will be a code, an h5 file containing the best performing deep learning model, a report, and a presentation.

Exploratory Data Analysis

Occurrence of Labels

The satellite images dataset contains over 40,000 training images and an additional 40,000 images for testing. Each image is made available both in .jpg and .tif format, with the larger .tif files containing an additional layer of color (infra-red). Due to the size of the files and the limited amount of computing power and time at hand, our project focused exclusively on the .jpg images on the training set.

The distribution of the label instances in the training set can be seen in the histogram below:



The predominant cloud coverage in the images is “clear” (28,431), followed by partly cloudy, haze and cloudy. The majority of images (37,513 images) in the dataset are also labeled as “primary” which represents the primary rainforest and is characterized by dense tree cover. The second most commonly occurring land use label is agriculture with 12,315 occurrences. Road and water (rivers and lakes) are also commonly occurring and appear in approximately 8,000 images each.

Artisanal mine, blooming (blooming of flowering trees), selective logging and slash and burn each have fewer than 500 instances. Blowdown (toppled trees resulting from naturally occuring microbursts) and conventional mines each have fewer than 100 instances.

An example image (train_1.jpg) is shown below. This image has been tagged with the following labels: primary, agriculture, clear, and water. An attempt was made to identify where the labels are even though this information is not provided in the dataset. The clear cloud cover represents lack of clouds and is therefore not shown.

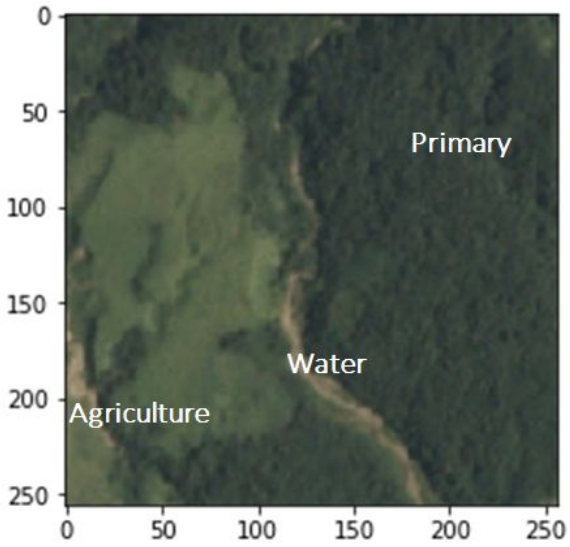
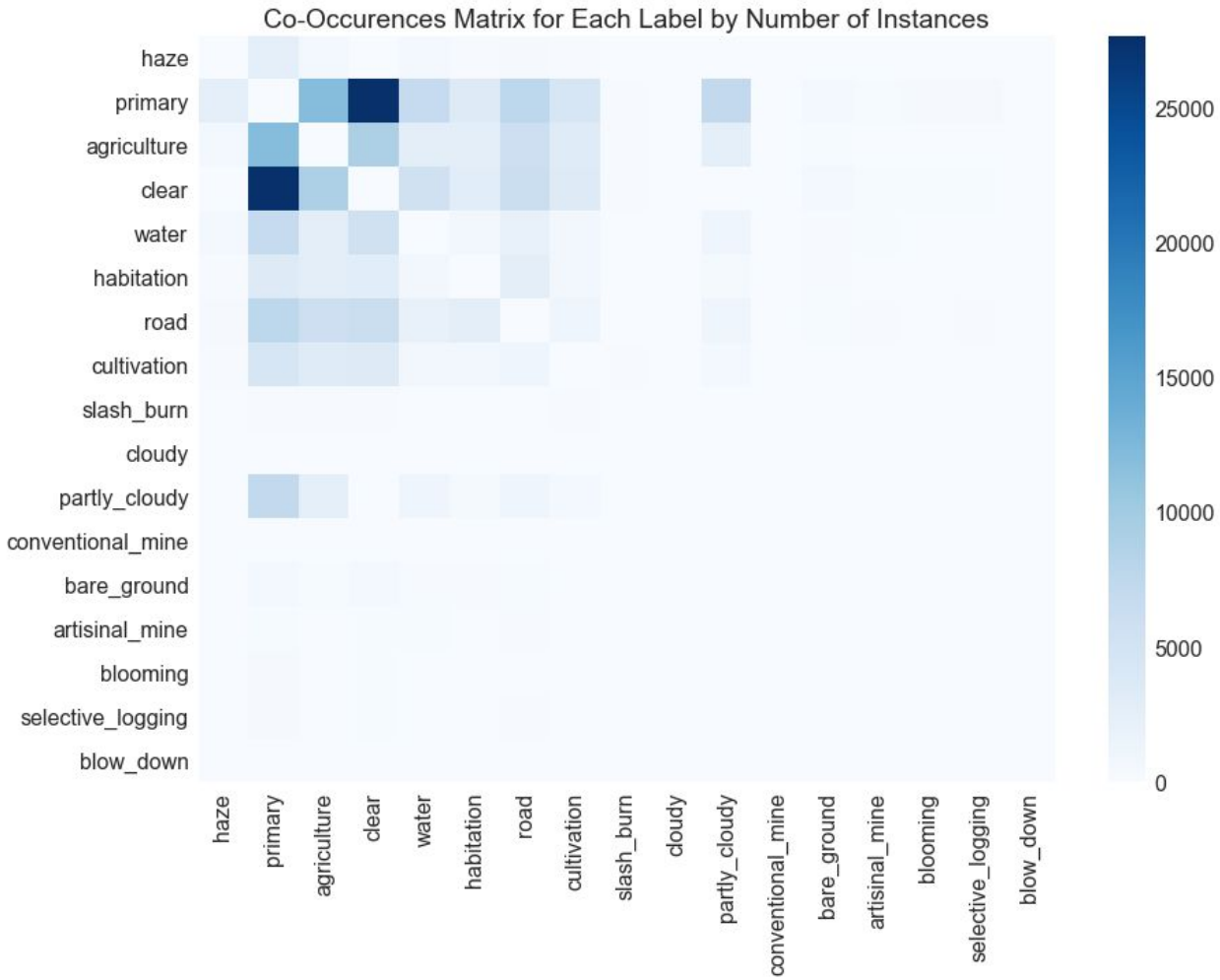


Figure 1 - Sample Image and Labels

Co-occurrence of labels

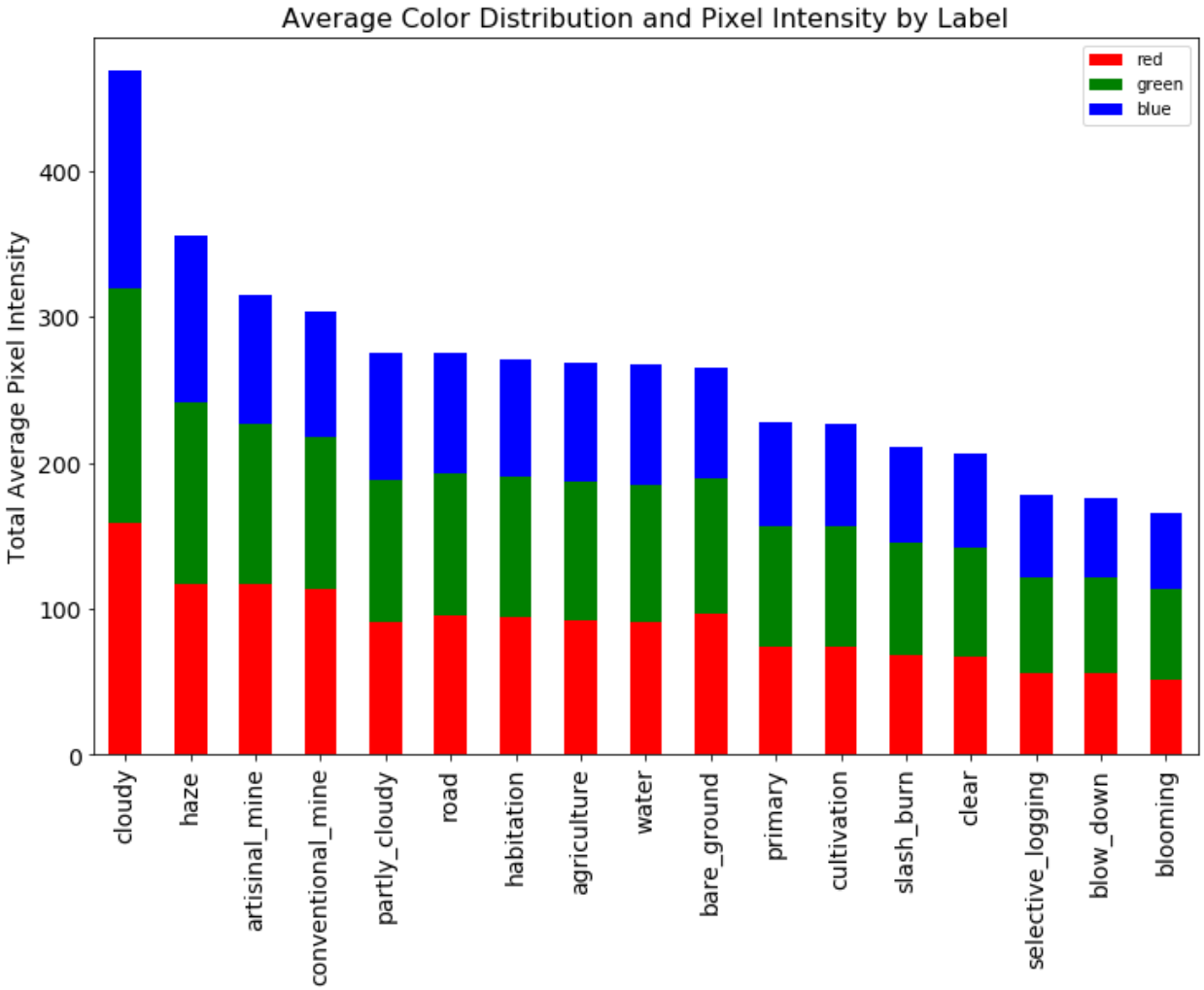
While each of the images can be tagged with a combination of the 17 available labels the cloud cover labels cannot co-occur with each other and can only take one of the possible 4 values per image (clear, partly cloudy, cloudy, or haze), the land use and vegetation labels can co-occur at any rate. To explore the co-occurrence of the labels in the training set we create a co-occurrence matrix shown below:



There are close to 28,000 images with co-occurrence of “clear” cloud cover and “primary” rainforest. “Primary” and “agriculture” represent the second most commonly occurring co-occurring labels with close to 12,000 examples, followed by “agriculture” and “clear” weather.

Decomposing Images

The average RGB values for all of the images tagged with a given label were calculated and are shown in the plot below.



Green is the dominant color for the majority of the labels, with the exception of conventional mines, artisanal mines, and bare ground for which red is the dominant color. Cloudy and hazy are the most saturated labels and this is expected given they are white or gray dominant.

An average image on a pixel by pixel basis was also created for each of the labels using all of the images on the training set.

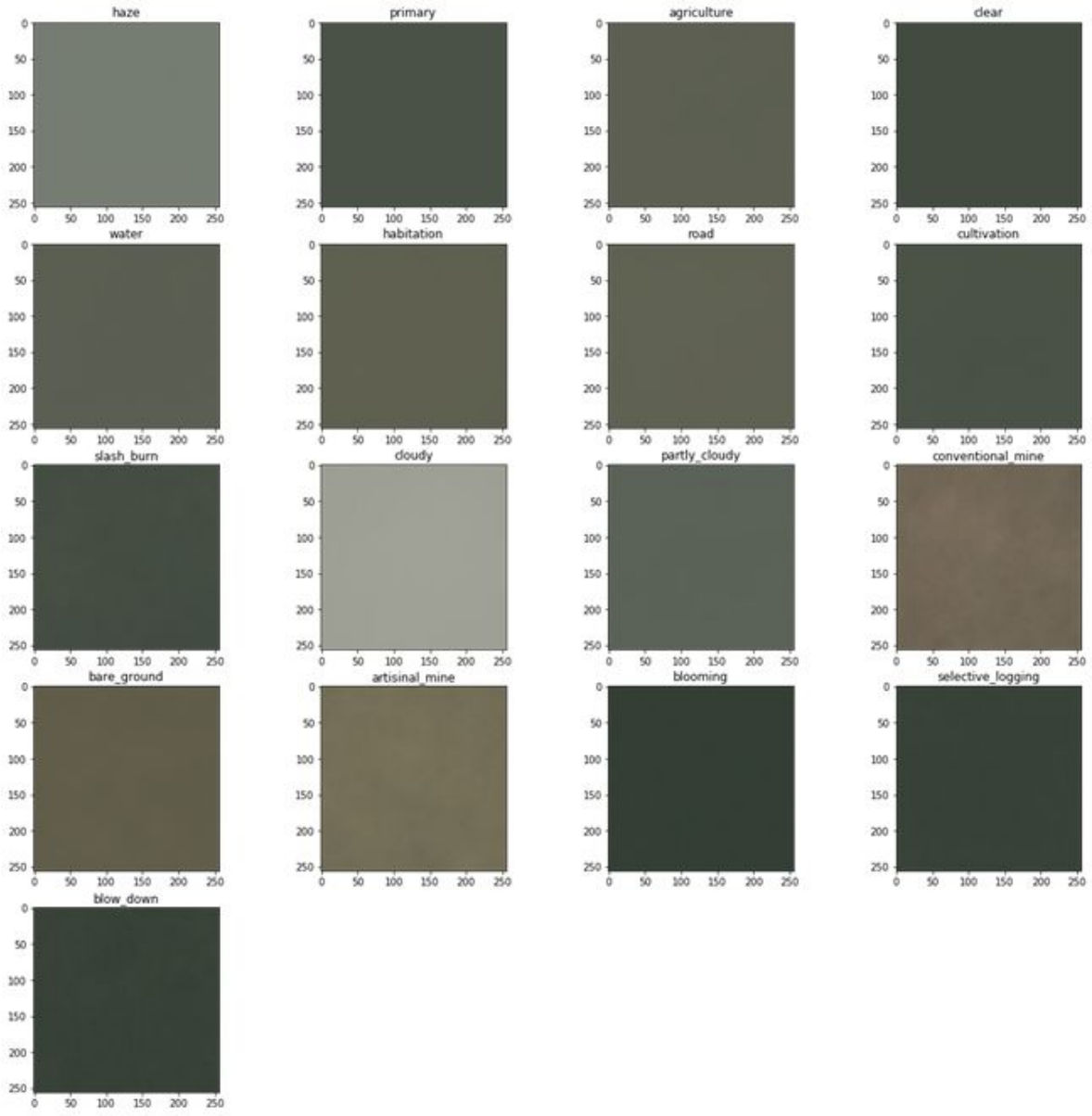


Figure 2 - Average Image for each label, calculated pixel by pixel

The pictures reflect to some extent what we observed in the previous breakdown of the colors per label: artisanal mine, conventional mine, and bare ground appear as shades of brown while all other images appear as shades of green or gray. Both cloudy and haze images have distinct gray colors. Because the images can contain a variety of labels, are not oriented in any specific way, and information regarding the location of the labeled features within the image are not available these images are not particularly helpful.

Machine Learning

Random Forest Model

For this project, we ultimately would like to explore convolutional neural networks for the classification of the various labels in the images. As a starting point, a baseline Random Forest model was created.

The labels for all of the images in the training set were one-hot-encoded and the average R, G, and B values per image were calculated as shown below:

	R	G	B	image_name	haze	primary	agriculture	clear	water	habitation	road	cultivation	slash_bu
0	91.787384	109.347000	103.609146	train_0	1	1	0	0	0	0	0	0	
1	65.639862	76.497375	64.301590	train_1	0	1	1	1	1	0	0	0	
2	43.164474	58.039383	56.862000	train_2	0	1	0	1	0	0	0	0	
3	51.411316	65.253769	53.972961	train_3	0	1	0	1	0	0	0	0	
4	57.342377	45.253403	19.525055	train_4	0	1	1	1	0	1	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	

The metric chosen for model evaluation was F_score (using f1_samples). The training set consisted of only the first 20,000 images with 5,000 images (20%) used for testing. Two parameters were tuned using GridSearchCV with 5 fold cross-validation: n_estimators and max_depth. Max depth values of 1 thru 12 and estimators from 100 to 400 in 100 increments were tried. The best parameters are shown below:

rfm = RandomForestClassifier(max_depth=10, n_estimators=200)

The overall F1 score obtained with this model is 0.704, however the average F1 score is 0.21 this is because the model does not perform well on rare occurring labels. The recall, precision and F1 score values per class were calculated and are shown below:

Table 1 - Recall, Precision and F1 scores for the Random Forest model per label

	Recall	Precision	F1
haze	0.039813	0.548387	0.074236
primary	0.988687	0.956234	0.972190
agriculture	0.494366	0.657329	0.564318
clear	0.938177	0.831550	0.881651
water	0.026211	0.660000	0.050420
habitation	0.014196	0.818182	0.027907
road	0.278067	0.617162	0.383393
cultivation	0.000000	0.000000	0.000000
slash_burn	0.000000	0.000000	0.000000
cloudy	0.520958	0.746781	0.613757
partly_cloudy	0.000844	0.250000	0.001682
conventional_mine	0.000000	0.000000	0.000000
bare_ground	0.014925	0.400000	0.028777
artisinal_mine	0.078125	0.625000	0.138889
blooming	0.000000	0.000000	0.000000
selective_logging	0.000000	0.000000	0.000000
blow_down	0.000000	0.000000	0.000000

Several of the classes are not being predicted: cultivation, slash burn, conventional mine, blooming, selective logging, and blowdown. It is possible that because these are small features on the image, the RGB average simply doesn't provide enough information (resolution) to allow the model to capture them.

A series of deep learning models were generated to see if we could improve upon the random forest model. The results are shown in the sections that follow.

Convolutional Neural Network (CNN) Model

Convolutional Neural Networks are a class of neural networks that have been successfully proven overtime for image classification problems. A series of CNN models were created using Keras with TensorFlow backend. The baseline CNN model architecture and its parameters are shown below:

- Image resolution: 64x64 (3 channels)
- Training set size: 20,000
- Validation set size: 7,000
- Test set size: 5,120 (multiples of data generator batch size of 256)
- Batch size: 256
- Early stopping: monitoring validation loss with patience of 5
- Metric: F1 score

The model was sequential and composed of:

1. CNN layer with 32 filters, kernel size (8,8) and ReLu activation
2. CNN layer with 64 filters, kernel size (8,8) and ReLu activation
3. Max_pooling layer with size (4,4)
4. CNN layer with 64 filters, kernel size (3,3) and ReLu activation
5. Dropout layer (probability 0.5)
6. Flattening layer
7. Fully connected layer with ReLu activation
8. Dropout layer (probability 0.5)
9. Fully connected layer for predictions with sigmoid activation

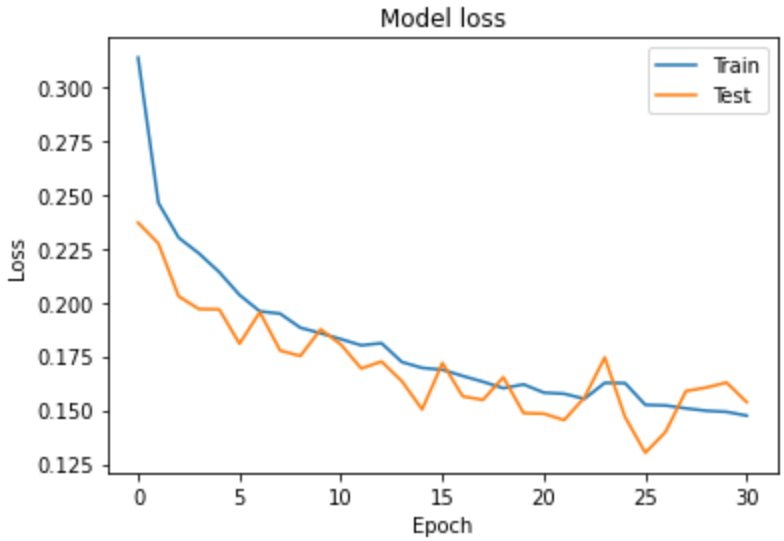
The Adam optimizer was used and loss was set to be measured using “bynary_crossentropy”. The model recorded loss, accuracy, F1 score, precision, and recall for both training and validation sets while training. A maximum of 50 epochs was set but early stopping kicked in at the 31st epoch due to no improvement in the validation loss between the 26th and 31st epochs.

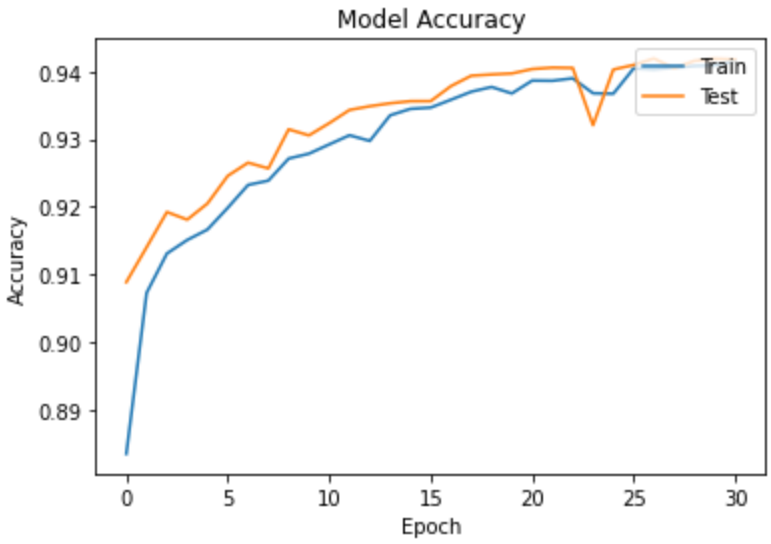
The model’s architecture summary is shown below:

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 57, 57, 32)	6176
conv2d_6 (Conv2D)	(None, 50, 50, 64)	131136
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 64)	0
conv2d_7 (Conv2D)	(None, 10, 10, 64)	36928
dropout_3 (Dropout)	(None, 10, 10, 64)	0
flatten_2 (Flatten)	(None, 6400)	0
dense_2 (Dense)	(None, 128)	819328
dropout_4 (Dropout)	(None, 128)	0
preds (Dense)	(None, 17)	2193

Total params: 995,761
Trainable params: 995,761
Non-trainable params: 0

The model's loss and accuracy during training are shown in the plots below:





The best F1 scores per class for the model were calculated by optimizing thresholds, the results are shown in the table below:

Table 2 - Baseline CNN model -Best threshold and F1 Score for each Label

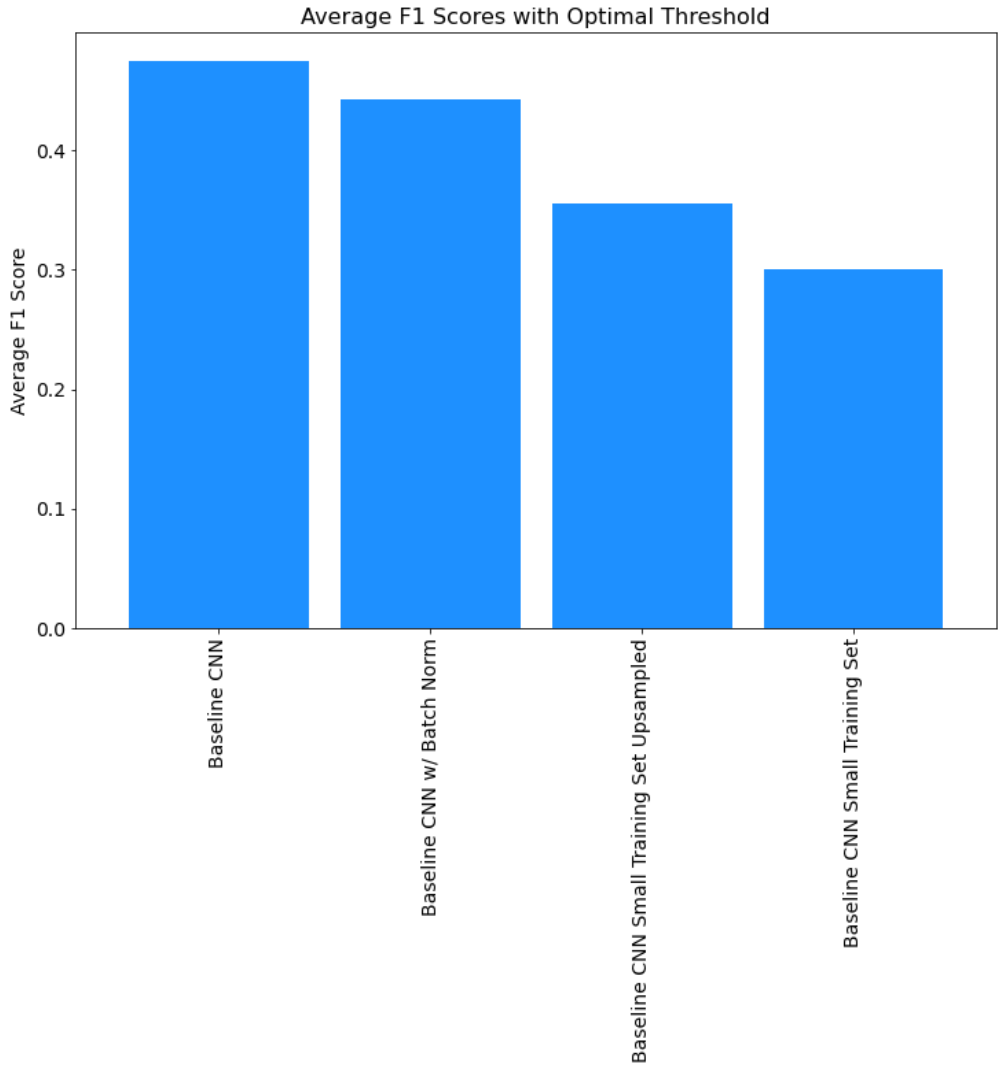
	Best Threshold	F1
haze	0.350	0.665
primary	0.550	0.977
agriculture	0.450	0.762
clear	0.540	0.946
water	0.270	0.465
habitation	0.250	0.479
road	0.400	0.627
cultivation	0.220	0.447
slash_burn	0.030	0.041
cloudy	0.270	0.717
partly_cloudy	0.220	0.851
conventional_mine	0.090	0.222
bare_ground	0.140	0.252
artisinal_mine	0.120	0.414
blooming	0.050	0.125
selective_logging	0.040	0.071
blow_down	0.010	0.012
Average_F1	0.235	0.475

Blowdown, slash and burn, conventional mine, selective logging and blooming only get predicted at very low threshold values and their F1 score is also low.

Several other parameters and models were tested for comparison and in an effort to improve upon the baseline CNN classification performance:

- Batch Normalization: an additional batch normalization layer was added to the model immediately following the first fully connected dense layer. This was done both in an effort to assess impact on training speed and to explore its impact on the networks's classification performance.
- Softmax for cloud coverage prediction: since the cloud coverage labels are mutually exclusive, the same baseline model architecture was used while modifying the target inputs to include only the 4 cloud coverage labels and the last (predictive) layer was changed to have a softmax activation function with 4 nodes (4 possible cloud classes). This was done to assess whether this model would perform better at exclusively predicting cloud cover and if so, utilize it for this purpose, separately from the land use and ground cover labels.
- Upsampling: blowdown images from the entire training set were utilized and usampled to verify whether the model would do a better job predicting that label.
- Grayscale images with higher resolution: the same baseline model was used with higher resolution grayscale images (1 channel) with 128x128 resolution to assess whether a single channel with better image resolution would do better at identifying and classifying the various labels.
- Baseline CNN with small training set: the number of training samples was reduced to 2000 samples to assess the training set size's impact on overall model performance.

From these models, the best performing model, as measured by the average F1 score was the baseline convolutional neural net model. The addition of a batch normalization layer, the increase in image resolution provided by the grayscale images, upsampling of the rare class and the softmax activation layer for cloud coverage prediction either decreased or did not have a significant impact in improving model performance. As expected, reducing the size of the training set had a significant detrimental impact to the average F1 score. The average F1 scores are shown below:



From the plot we can observe the effects of reducing the size of our training set, as the worst performing model differs from the Baseline CNN only in its training set size (2000 vs 20000 training images). Batch normalization slightly reduced the baseline model's performance, and the higher resolution provided by the grayscale model also yielded worse results.

One way to better understand how CNNs work is by visualizing the network's filters. In order to create filter visualizations, a random image is created and fed to the network, the gradients of the loss for the image for a given filter are computed and the image is progressively updated to maximize that filter's activation function. The resulting images represent the inputs that maximize each filter's activation and can give us clues about what the network's filters are "looking for". The below images are filter visualizations from our baseline CNN:

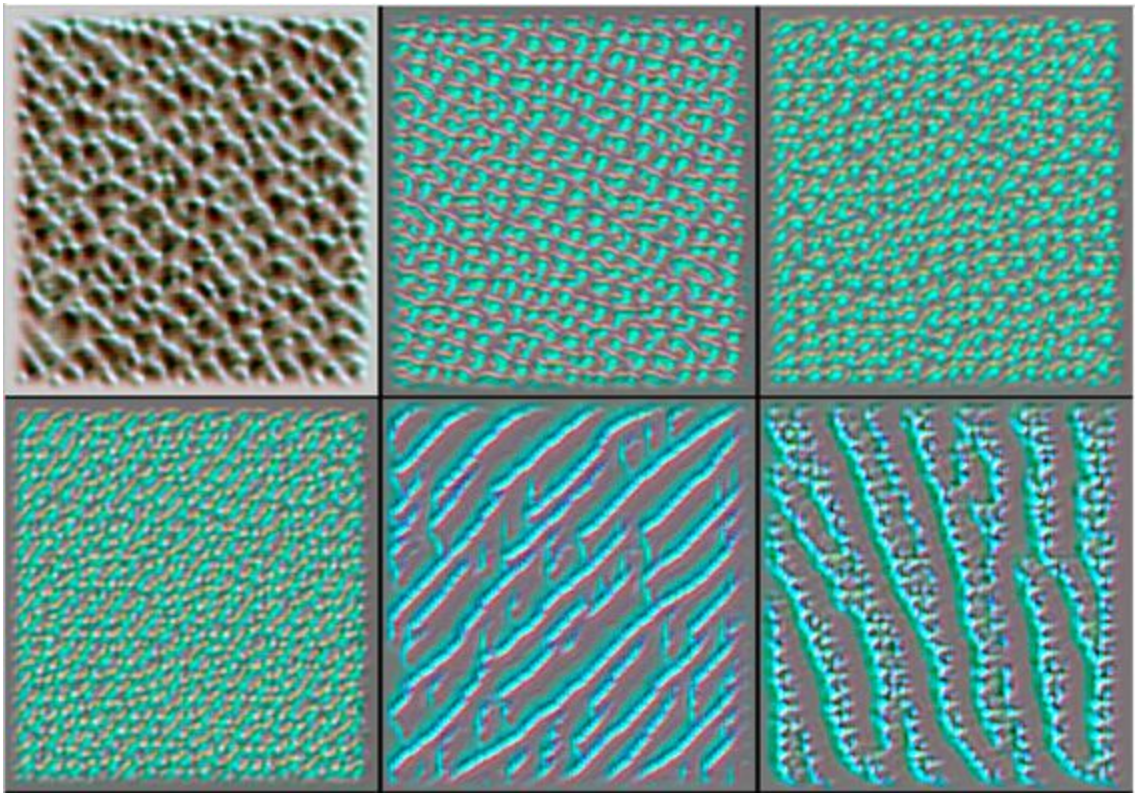


Figure 3 - Filter Visualization for the last convolutional layer (conv2d_7) of the baseline model

The filter's remind us of some of the common patterns observed in our satellite images: tree cover, bare ground, agriculture, rivers and roads. More broadly, these filters are likely extracting specific edge and gradient patterns from the input images. The source code to generate these filter visualizations was adapted from Reference 4.

Transfer Learning

Keras' applications interface offers us the ability to load pre-trained neural networks, thereby allowing us to use the knowledge gained from a problem to a different, related problem - this is known as transfer learning. There are several available models that can be used in image recognition tasks, for our problem we utilized the VGG16 model to try and improve upon our predictions.

The VGG16 model, developed by Oxford's Visual Geometry Group (VGG), consists of a 16 layer convolutional neural network. By specifying the parameter *" weights = 'imagenet' "* when the model is loaded, the network's initial weights will have been pre-trained using the ImageNet dataset. The ImageNet dataset contains ~ 1.2 million images and the network weights were optimized to classify these input images into 1,000 different object categories. Kera's

pre-trained networks are known for having a strong ability to generalize to images outside of the ImageNet set.

Some of the resulting images for the filters from the pre-trained VGG16 network on the ImageNet set are shown below:

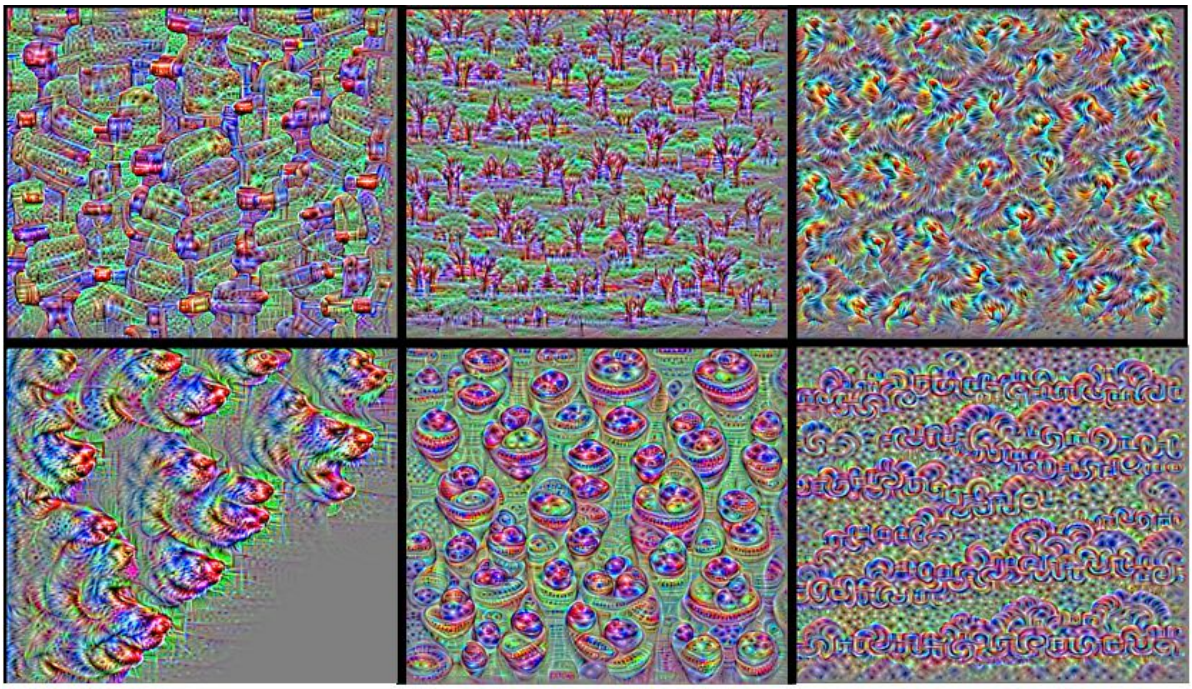


Figure 4 - Filter visualizations for the VGG16 network pre-trained on the ImageNet data

Some of the images displayed resemble dogs, marbles, trees and chains. These are a direct result of the contents of the ImageNet dataset and the network's original purpose of identifying the 1,000 different labels and objects.

To adapt the VGG16 model for our purposes, we integrated it into a new model, for which the VGG16 block essentially functions as a feature extractor. The output from the VGG16 network was flattened and fed into a final classifier type layer and retrained with the satellite images to predict our 17 labels of interest as shown in the summary architecture below.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_1 (Batch (None, 64, 64, 3)         12
_________________________________________________________________
vgg16 (Model)                (None, 2, 2, 512)         14714688
_________________________________________________________________
flatten_1 (Flatten)          (None, 2048)              0
_________________________________________________________________
preds (Dense)                (None, 17)                34833
=================================================================
Total params: 14,749,533
Trainable params: 14,749,527
Non-trainable params: 6
_________________________________________________________________
```

Using filter visualization for various layers of the retrained network we gain a better understanding of how the satellite images and the new classes influenced the network's weights:

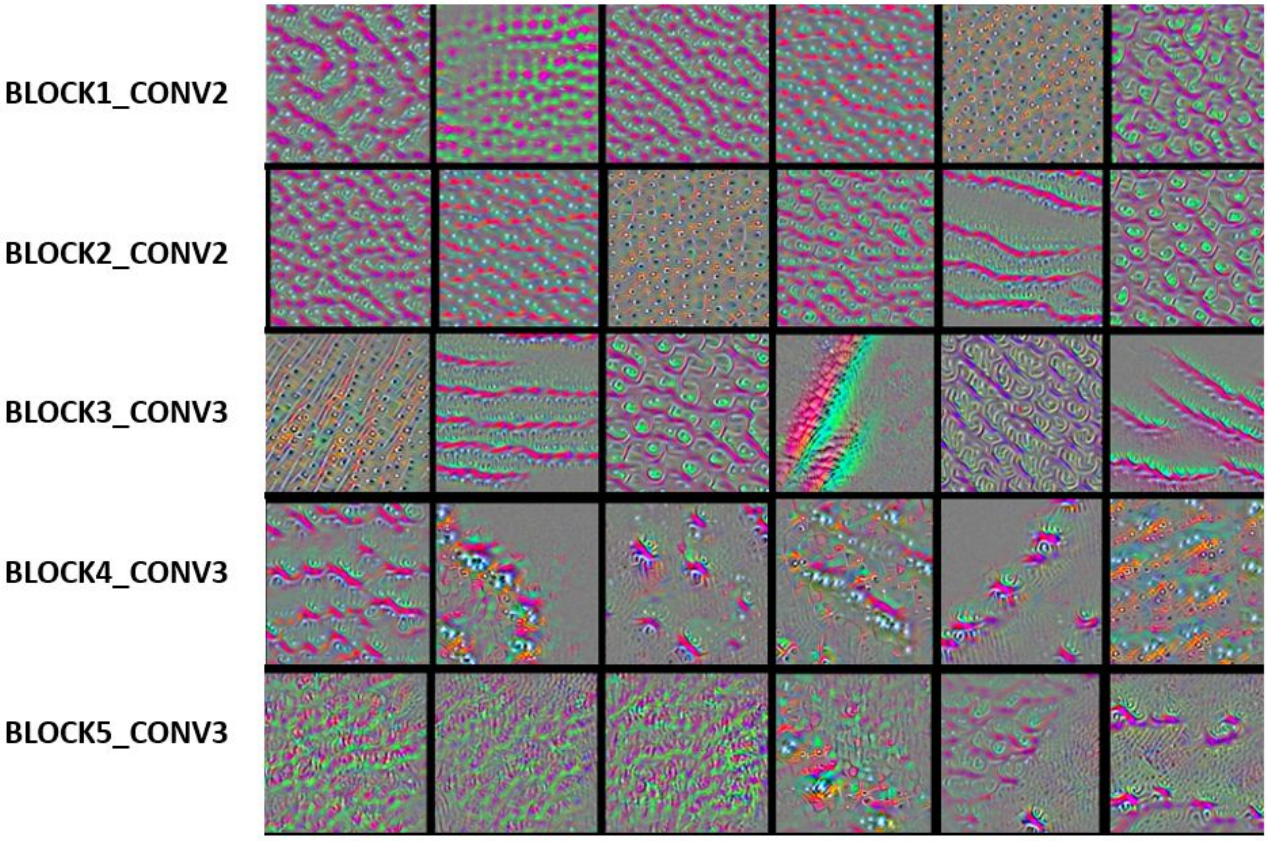


Figure 5 - Filter visualizations for various layers of the VGG16 network, retrained on the satellite images

It can be noted that the filter patterns causing maximum activation have significantly changed from the ones originally loaded in the VGG16 model trained on the ImageNet set. The deeper filters of the layers generally tend to look for more complex patterns while the earlier filters seem to be looking for general edges and patterns. The early filters also resemble the filters observed in our own baseline CNN network in the previous section.

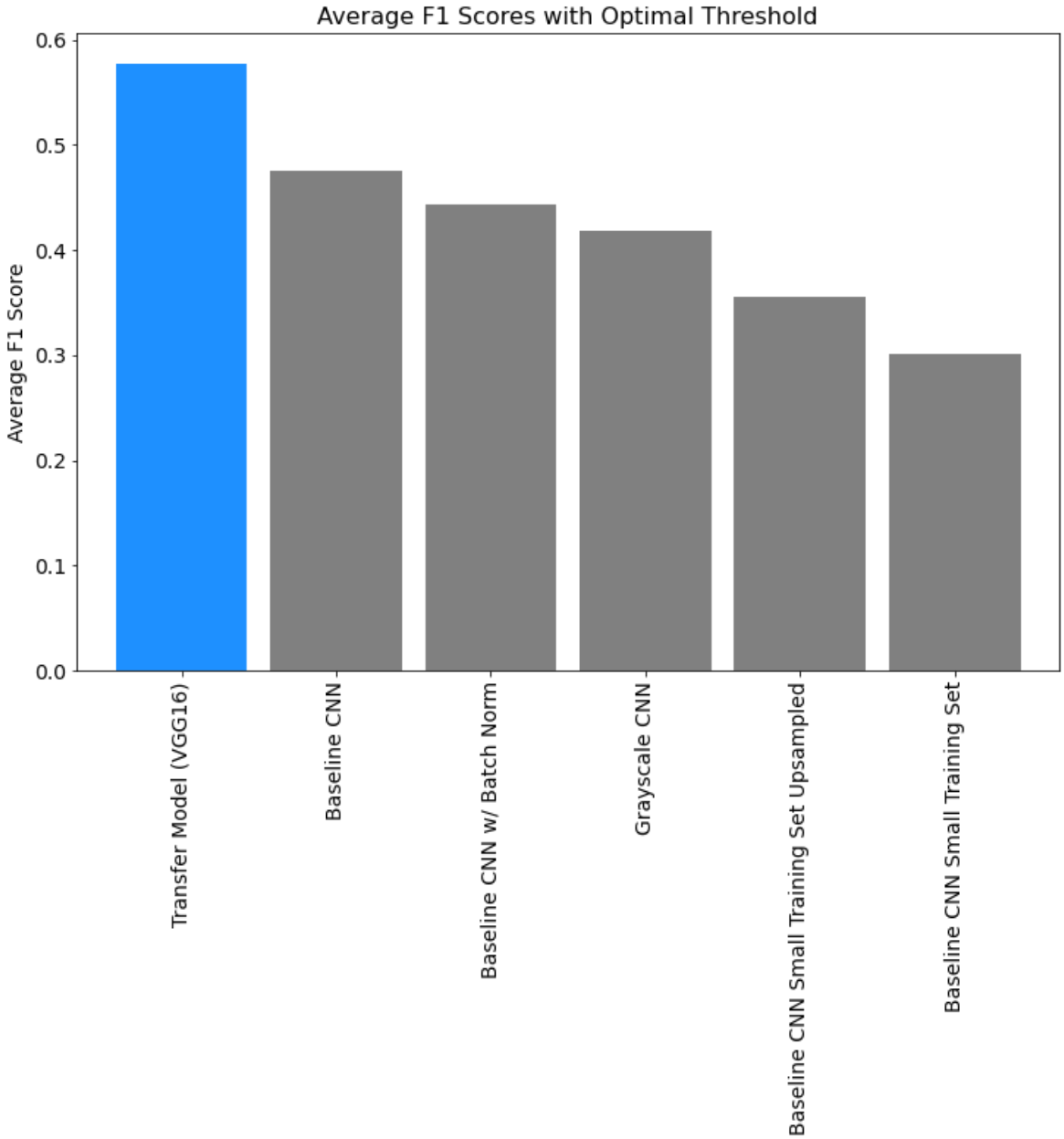
The VGG16 model contains a larger number of layers with a total of 4224 convolutional filters, which is significantly more complex than our baseline CNN with only 160 filters. IT is likely due to the fact that it is able to capture more patterns that it performs so much better.

The transfer model yielded vast improvements in classification performance over the baseline convolutional neural net. The best F1 scores and thresholds per label for the transfer model are shown below:

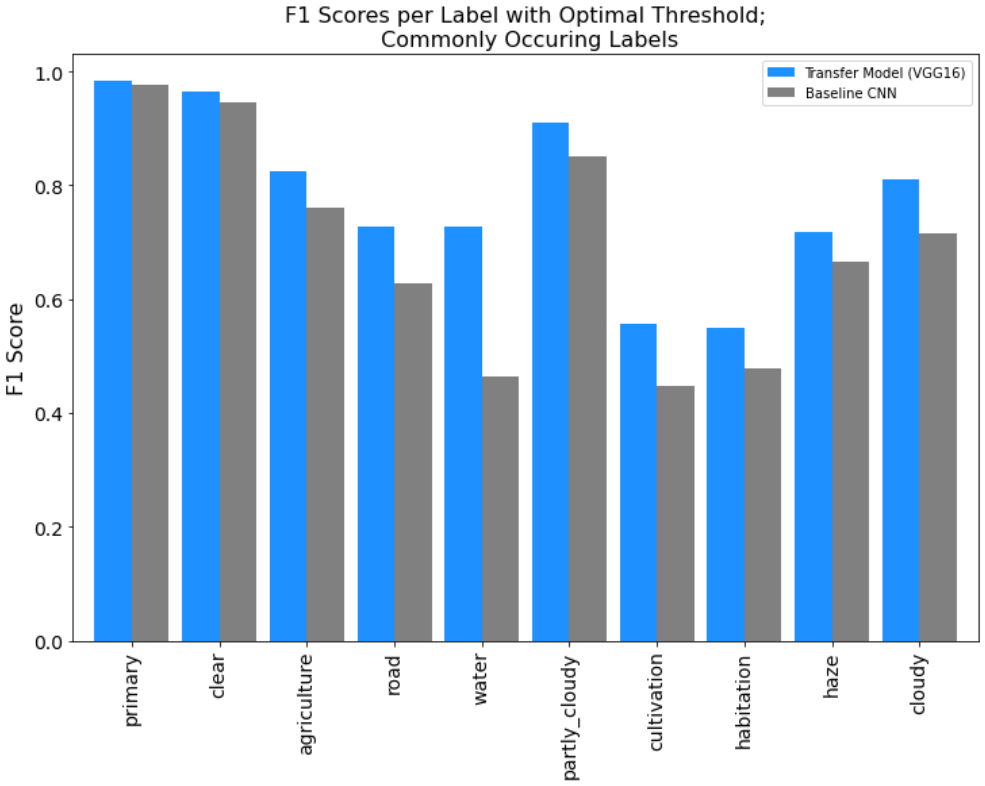
	Best Threshold	Best_F1
haze	0.180	0.719
primary	0.380	0.983
agriculture	0.230	0.824
clear	0.690	0.964
water	0.280	0.727
habitation	0.240	0.549
road	0.280	0.727
cultivation	0.350	0.557
slash_burn	0.050	0.114
cloudy	0.500	0.812
partly_cloudy	0.460	0.909
conventional_mine	0.030	0.207
bare_ground	0.120	0.303
artisinal_mine	0.700	0.701
blooming	0.140	0.220
selective_logging	0.190	0.310
blow_down	0.090	0.190
average	0.289	0.577

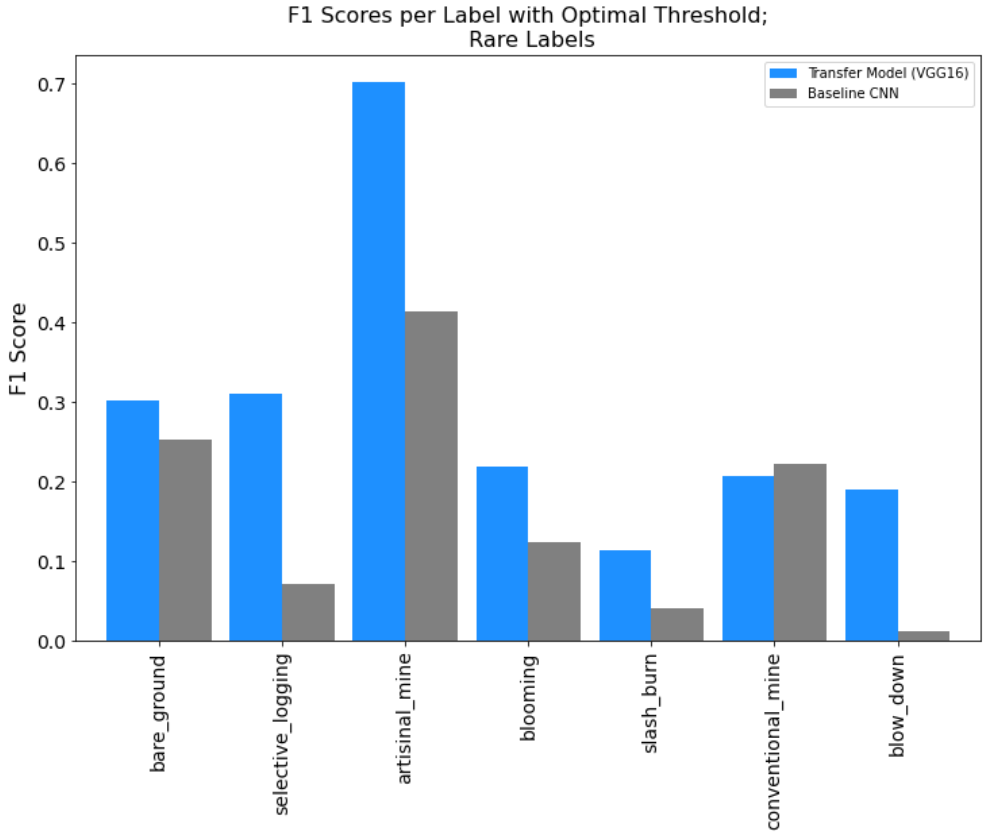
It can be noted that for some labels, especially the ones with low occurrence, the best threshold is quite low (slash and burn, conventional mines and blowdown). There is also a noted difference in F1 scores between the commonly occurring labels and the rare labels, with slash and burn, conventional mines, bare ground, blooming, selective logging and blow down having best scores under 0.32.

The threshold that yielded the best F1_scores was also computed for all of the deep learning models, and the average F1 score for each model was calculated - the results can be found in the plot below:



In the figure above it becomes clear that the VGG16 model has the highest average F1 score with the Baseline CNN model performing second best. Below we break down how much the transfer model improved our results for both the high occurrence and low occurence labels over our Baseline CNN:





The only label for which the transfer model had inferior results to the baseline CNN was “conventional mine”, the difference is small and not enough to justify using the baseline CNN model for the prediction of this class.

Conclusion

The model’s predictions are useful for labeling the data, which in turn could help organizations make more informed decisions in the rainforest’s preservation and management efforts.

The VGG16 model was the best performing model for all labels, with the most drastic improvements observed for the low occurring labels (blow down, selective logging, blooming, slash and burn and artisanal mine). This implies that the depth and complexity of the neural network, as well as the size of the training set play an important role in improving deep learning model performance for computer vision problems. Visualizing some of the filters for the networks helped gain a better understanding of the features being extracted by the model from the input images.

For future work, increasing the size of the training set by using all available images and using higher resolution .TIF images would be recommended, as well as artificially increasing the training set size through the use of image distortion, rotation and cropping, especially for the low occurrence labels.

References:

1) https://www.nationalgeographic.com/environment/2018/11/how-cutting-the-amazon-forest-could-affect-weather/
2) https://www.nationalgeographic.com/environment/2019/08/why-amazon-doesnt-produce-20-percent-worlds-oxygen/#close
3) https://advances.sciencemag.org/content/4/2/eaat2340
4) Simulated Changes in Northwest U.S. Climate in Response to Amazon Deforestation - https://journals.ametsoc.org/doi/10.1175/JCLI-D-12-00775.1
5) Team, K. (2020). Keras documentation: Visualizing what convnets learn. Retrieved 5 August 2020, from https://keras.io/examples/vision/visualizing_what_convnets_learn/