Automatic Detection of Deforestation Using Machine Learning

Forest Shi
Columbia University
fs2751@columbia.edu

December 16, 2022

Abstract

Deforestation is a significant global problem with far-reaching impacts on the environment, wildlife, and the climate. Early detection of deforestation is crucial for timely intervention and conservation efforts. In this paper, we propose a machine learning approach to automatically detect deforestation from satellite images using a convolutional neural network. This approach allows for efficient and accurate detection of deforestation and hope be used to assist relevant authorities to take prompt action and protect our natural carbon sinks.

I. Introduction

We are currently facing a climate change crisis, with the world producing 52 billion tonnes of CO_2 every year.[1] A significant contributor to climate change is deforestation, with various sources estimating its total contribution of global greenhouse gas ("**GHG**") emissions to be between 10%[2] and 20%[3]. Trees are natural carbon sinks that capture $C0_2$ from the atmosphere, so cutting them down reduces the planet's ability to decarbonize itself and puts more of the burden on us to do so. Deforestation also disrupts the ecosystem and can have potential negative effects on wildlife, food sources, water reserves, and more. One of the ways to combat deforestation is early detection. Satellite images can help us surveil large areas of forests from the sky, but surveillance cameras on their own are not useful unless we can automate the detection progress using machine learning, as it would be costly and inefficient to have humans constantly monitor the footage. In this project, we trained a convoluntional neural network ("**CNN**") on a dataset of satellite images of the Amazon Rainforest and experimented with different network architectures and hyperparamters to achieve a high prediction accuracy on unseen images.

II. Dataset

The dataset we decided to use is from Kaggle titled: Understanding the Amazon from Space (the "**Amazon Dataset**")[4]. A few other datasets were also considered[5], but ultimately the Amazon Dataset was chosen for its diversity, large size, and high quality, which made it an ideal candidate for training machine learning models on. Excluding the test set for which labels Kaggle did not provide, the dataset consists of over 40,479 sample images with 17 independent class labels covering a range of topics including agriculture, mining, urban infrastructure, natural landscapes, and weather. The images were derived from Planet's full-frame analytics scene products using 4-band satellites in sun-synchronous orbit (SSO) and International Space Station (ISS) orbit. Each image is 256x256 pixels with 3 color bands: red, green, and blue. Additionally, we wanted a dataset that was diverse enough to be able to have high prediction accuracy on unseen images outside of the training dataset. The Amazon Rainforest spans multiple countries in South America and covers a wide range of terrain types. This will allow the model trained on the Amazon Dataset to generalize better to detect deforestation in other regions of the world.

III. Methodology

i. Data Preprocessing

The Amazon Dataset included 17 classes, as shown in Figure 1. The number of examples in each class are as follows:

- haze - 2697
- primary - 37,513
- agriculture - 12,315
- clear - 28,431
- water - 7,411
- habitation - 3,660
- road - 8,071
- cultivation - 4,547
- slash and burn - 209
- cloudy - 2,089
- partly cloudy - 7,261
- conventional mine - 100
- bare ground - 862
- artisinal mine - 339
- blooming - 332
- selective logging - 340
- blow down - 101

To simplify the problem and make it a binary classification task, we categorized each label as either representing deforestation ("1") or non-deforestation ("0"). Of the 17 labels, the 8 that were considered deforestation included agriculture, habitation, road, cultivation, artisinal mine, and selective logging. This allowed us to easily identify whether deforestation was occurring within the region depicted in the image. After categorizing the labels into 1s and 0s, we found that there were a total of 15,899 instances of deforestation and 24,580 instances of non-deforestation, giving us a dataset that is approximately 39% deforestation. This is a fairly balanced dataset for binary classification.

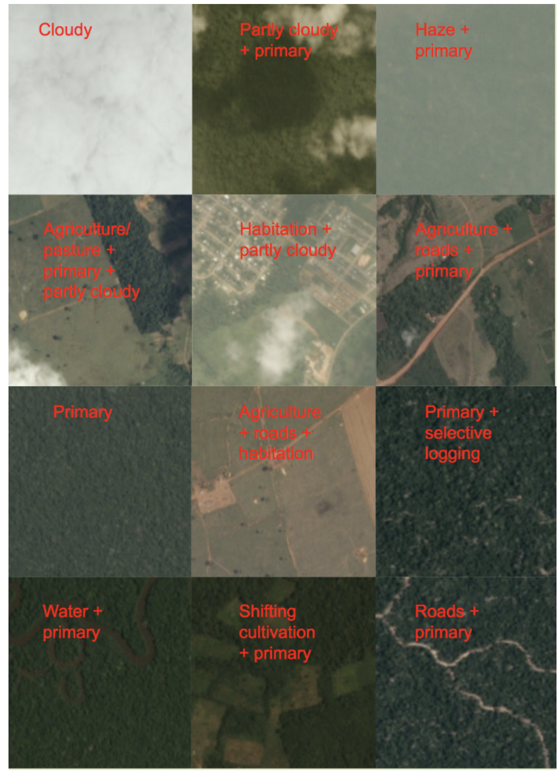

Figure 1

ii. Data Sampling

One of the challenges of working with the Amazon Dataset was the time and resources required to load images from the disk and into memory to prepare them for training. The Keras model requires that the images be converted to the correct RGB color space using the PILLOW library, which turned out be a very time-consuming process when working with a large dataset. To address this bottleneck, we implemented a technique called random sampling to select a subset of the images from the dataset to use for training. By carefully selecting a representative sample of the images, we maintained the same positive and negative class ratios while reducing the number of images that needed to be processed. This allowed us to train the model more efficiently, using less time and resources.

To begin with, a sample of 2,000 images were taken from the 40,479. Using more images took an excessive amount of time to run and caused the Google Colaboratory Notebook ("**Colab**") to time out. The sampled dataset was split into a training set of 1,500 images and a validation set of 500 images, with a ratio of 75/25. Later, with an improved data loading pipeline running on a faster computer's Jupyter Notebook ("**Jupyter**") instead of Colab, we were able to process and train with a dataset of 16,000 images in a reasonable time. The results of both experiments were compared in the section IV of this paper.

iii. Model Selection

To learn the features of the images, we decided to go with a convolutional neural network. CNNs are better for training on datasets of images because they are designed to work with the spatial structure of the data. CNNs use convolutional layers and pooling layers to automatically extract features and reduce dimensionality, respectively. They are also shift-invariant, meaning they can recognize patterns in different locations in the input data. In contrast, dense layers treat the input data as a flat vector and are less effective for image data.

We attempted to improve the performance of our custom-built CNN by using transfer learning with pre-trained models like ResNet50 and EfficientNetB0. However, both models overfitted on the training data and performed poorly on the validation set. This suggests that transfer learning may not be an effective approach for this particular problem due to the dataset being highly specialized.

After experimenting with various approaches, we settled on a simple custom-built CNN for our final model. It consists of a few convolutional layers, max pooling layers, and a dense layer at the end with a sigmoid activation function for binary prediction. The model architecture is shown in Figure 2.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 178, 178, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 89, 89, 64)       0
 )

 conv2d_1 (Conv2D)           (None, 87, 87, 128)       73856

 max_pooling2d_1 (MaxPooling  (None, 43, 43, 128)      0
 2D)

 flatten (Flatten)           (None, 236672)            0

 dense (Dense)               (None, 128)               30294144

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 30,369,921
Trainable params: 30,369,921
Non-trainable params: 0
_________________________________________________________________
```

Figure 2

IV. Results

A sample dataset consisting of 2,000 images was used to train the first model. The training was performed over 15 epochs, during which Keras' built-in ModelCheckpoint was utilized to save the model with the lowest validation loss on any given epoch. This helped to prevent overfitting by effectively implementing early-stopping. The accuracy and loss on both the training and validation sets are shown in Figure 3.

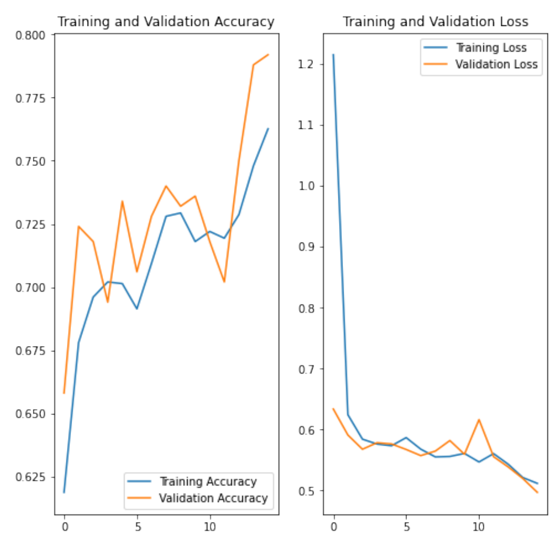

Figure 3

The best accuracy achieved with the our CNN trained on this small dataset of 2,000 images was 80% accuracy on the validation set.

Next, we tried training the CNN on a larger dataset of 16,000 images, with 12,000 images in the training set, and 4,000 images in the validation set. The hope is that this larger dataset will allow the model to learn more nuanced patterns and features, potentially resulting in improved performance. After training for 20 epochs, we achieved 84% accuracy on the best model. This is a significant increase from the smaller dataset, but not as high as we had expected given that the dataset is now 8 times larger. This suggests that we may be reaching the limits of how much the model can improve just by increasing the number of data samples. There is diminishing returns to model accuracy from increasing dataset size, and at a certain point adding more data may not significantly improve the model's accuracy. However, larger models and datasets require more computational power, are more energy intensive, and have a larger carbon footprint. Therefore, it is important to strike a balance between the need for accurate models and the environmental consequences of their training as proposed in Schwartz's paper on Green AI[6]. Hence, we decided to not increase the size of our dataset further and focused tuning other hyperparameters such as batch size, model architecture, and learning rate using grid search.

We experimented with different model architectures and learning rates to find the best combination. In an effort to save time on training, we performed hyperparameter tuning on the 2,000 image dataset and documented the results in Table 1. We discovered that the best performance of the model on the 2,000 image dataset was with a learning rate of 0.001 and the model architecture shown in Figure 2, achieving an accuracy of 80.0%. We used the same model architecture and learning rate on a dataset of 16,000 images to increase accuracy to 84.3%. This is the best validation accuracy we have achieved and is highlighted in Table 1. We used the best-performing model to make predictions of deforestation on new, unseen images. Some examples of these predictions, along with the corresponding images, are shown in Figure 4.

Table 1: *CNN Model Accuracies*

Data Size	Learning Rate	Val Accuracy
2,000	0.001	80.0%
16,000	**0.001**	**84.3%**
2,000	0.01	60.6%
2,000	0.0001	75.0%

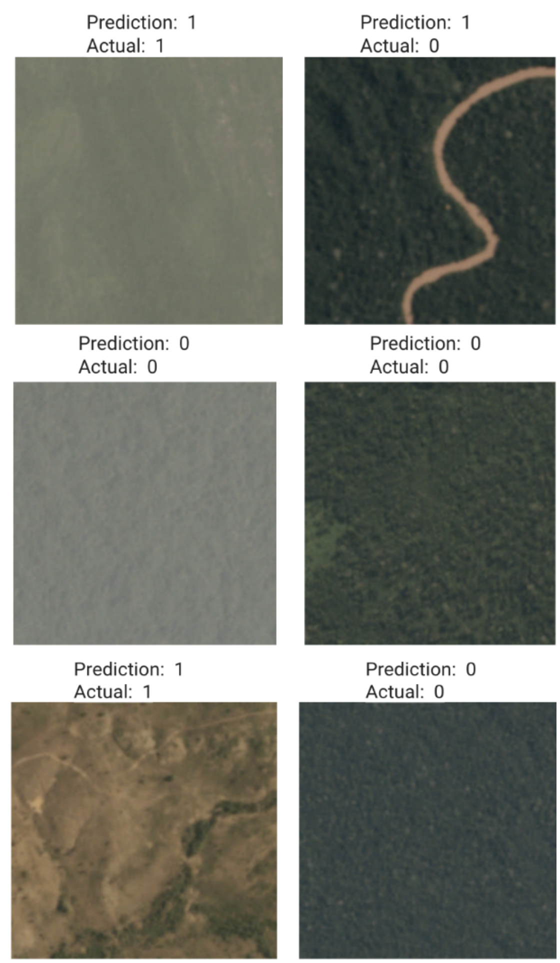

Figure 4

V. Challenges and Future Work

As we can see from Figure 4, our best-performing model does quite well on predicting images outside of the training sample. However, the model has encountered some challenges in accurately predicting on images with cloud cover and images of rivers and other water bodies. For example, as shown in Figure 4, the top right image was incorrectly labeled as deforestation when it should have been classified as no deforestation. Upon closer inspection, it is evident that the river running through the forest is visually similar to a road going through a forest. This suggests that the model may benefit from additional training data featuring both roads and water bodies, in order to improve its ability to distinguish between these features. Cloudy satellite images, as shown in Figure 1, can lead to inaccurate labeling of deforestation, as it is impossible to determine the condition of the forest beneath the clouds. This can result in missed instances of deforestation, which can be detrimental in practice. It is important to consider the quality of the images when making predictions, as it plays a crucial role in the accuracy of the results. For future work, we may consider adding a third label of cloudy or unclear in addition to binary classification of deforestation and non-deforestation.

In selecting a metric for evaluating a machine learning model, it is important to consider the context in which the model will be applied and align the metric with the model's goals and the problem it is trying to solve. In the context of detecting deforestation, it is more important to prioritize recall over precision. This is because correctly identifying all instances of actual deforestation is of greater importance, even if it means occasionally identifying a non-deforestation area as deforestation. In other words, false negatives (missing actual deforestation) are costly, while false positives (incorrectly identifying non-deforestation as deforestation) are trivial. For future work, we could write a custom loss function that prioritizes getting as high of a recall as possible instead of maximizing accuracy.

To further improve the model, future work could include more hyperparameter tuning beyond just learning rate and model architecture. One potential hyperparameter to consider adjusting is batch size. In addition, it would be beneficial to explore alternative optimization methods such as random search and Bayesian optimization to potentially achieve even better performance.

Moreover, to evaluate the generalizability of the model, it would be beneficial to validate it on a dataset featuring images from a different region than the one from the Amazon Dataset. This would allow us to assess the model's performance in predicting deforestation in other parts of the world and gauge its ability to generalize to new environments.

VI. Conclusion

In conclusion, the use of a convolutional neural network to detect deforestation from satellite images is a promising approach that can help identify and monitor changes in land use and vegetation cover. This method allows for the rapid and accurate identification of deforested areas, which can be critical for conservation efforts. However, it is important to note that the accuracy of the model may be affected by factors such as the resolution and quality of the satellite images, as well as the availability and diversity of training data. Despite its limitations, the deep learning model presented in this paper has demonstrated some success and serves as a foundation for future work. Potential avenues for improvement include the incorporation of additional data, the use of more effective metrics, and more thorough hyperparameter tuning. This paper demonstrates that the use of deep learning techniques like a CNN can be a valuable tool for detecting and monitoring deforestation at a large scale and aid in the fight against climate change.

References

[1] Gates, Bill. *Avoiding a Climate Disaster: The Solutions We Have and the Breakthroughs We Need.* Alfred A. Knopf, 2021.

[2] "What Is the Role of Deforestation in Climate Change and How Can 'Reducing Emissions from Deforestation and Degradation' (REDD+) Help?" Grantham Research Institute on Climate Change and the Environment, 14 Jan. 2021, https://www.lse.ac.uk/granthaminstitute/explainers/whats-redd-and-will-it-help-tackle-climate-change/.

[3] Gregory P. Asner, PhD. "Measuring Carbon Emissions from Tropical Deforestation: An Overview." Environmental Defense Fund, 2010, https://www.edf.org/sites/default/files/10333_Measuring_Carbon_Emissions_from_Tropical_Deforestation–An_Overview.pdf.

[4] "Planet: Understanding the Amazon from Space." Kaggle, 2017, https://www.kaggle.com/c/planet-understanding-the-amazon-from-space.

[5] "Towards Detecting Deforestation." Kaggle, 2020, https://www.kaggle.com/c/dsc6232-rwanda-summer2020-hw2/.

[6] Schwartz, Roy, et al. "Green AI." ArXiv.org, 13 Aug. 2019, https://arxiv.org/abs/1907.10597.