2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA)

Estimating Deforestation using Machine Learning Algorithms

Keanu Nichols, Patrick Hosein
Department of Computer Science
The University of the West Indies
St. Augustine, Trinidad
keanu.nichols@my.uwi.edu, patrick.hosein@sta.uwi.edu

***Abstract*—Due to the urgency of the Climate Change phenomenon, it has become important to estimate the rate of deforestation in various countries since forests are essential for oxygen generation. The advent of popular machine learning algorithms, such as those that are applied to the field of computer vision, has led to the use of approaches, such as fully convolutional neural networks (FCN), for performing semantic segmentation on satellite images to determine forested and non-forested areas. However, these models tend to be computationally intensive and, even with the advancement of specialized hardware such as GPU's (Graphical Processing Units), these approaches can still be quite costly especially for Small Island Developing States (SIDS) which are dis-proportionally affected by Climate Change. We consider the use of less computationally intensive approaches such as, logistic regression, linear support vector machine and Naive Bayes to achieve similar performance to a FCN but at a much lower cost. We perform semantic segmentation on satellite images to determine the percentage of forested areas and use this information to determine the rate of change of deforestation over a period of time. We compare the performance, computing requirements, storage requirements and robustness of different Machine Learning techniques.**

***Index Terms*—Deforestation; Machine Learning; Satellite Imagery; CNN; Climate Change; Small Island Development States**

I. Introduction

Deforestation is a global problem [1] that has affected many forests around the world. However, for Caribbean islands, it is quite difficult to track the rate at which this is occurring. Caribbean islands are typically not within the footprints of popular satellites such as Landsat or Sentinel since these satellites typically survey places in the United States of America and the European Union. Hence, Caribbean Islands are forced to use other means of obtaining deforestation data such as the one described in [2], [3], and these forms of gathering data can be time consuming and are not easily accessible to the public. Additionally, it can become dated and more recent satellite images must then be labeled.

Hence, being able to determine a way to track deforestation is a useful tool since it would give smaller Caribbean islands an idea of how the rate of deforestation is occurring. For the area of remote sensing, computer vision has been able to solve several problems ranging from crop and soil segmentation [4], to cloud detection [5]. However, for many of these problems a large dataset is available and easily accessible but for Caribbean islands this is not the case since it can be costly to obtain.

The Normalized Difference Vegetation Index (NDVI), [6], is typically used to monitor drought, forecast agricultural production and assist with managing fire zones. It is calculated by the following formula:

$$\mathsf{NDVI} = \frac{\mathsf{NIR} - \mathsf{RED}}{\mathsf{NIR} + \mathsf{RED}}$$

where NIR is reflection in the near-infrared spectrum and RED is reflection in the red range of the visible spectrum.

These images can therefore be used to distinguish between live vegetation and dead vegetation or no vegetation such as in cities. A few studies [7], [8], [9] have used NDVI sensors in order to look at live vegetation to determine forested areas such as in [10] which looked at tracking the rate at which forests were being degraded in Pahang, Malaysia. Satellites such as Landsat 8 contain the near-infrared and red channels and hence NDVI images are easily accessible for most regions. However, when it comes to Caribbean islands which have few RGB images taken from these satellites, NDVI images are consequently also uncommon.

Convolutional neural networks (CNN) have become the state-of-the-art technique to be used in problems in the realm of Computer Vision such as object recognition and image segmentation. Specifically, fully convolutional neural networks (FCN) [11] looked at the problem of having to label full scale images by summarizing the features of an image by using convolutions and later performs transposed convolutions to return the fully labelled image. This work was then built upon by [12] which proposed a model called "U-Net" which looked at creating a "u-like" architecture which was symmetric and uses convolutions in the first half of the model and in the other half uses transposed convolutions. U-Net became quite popular since the model was able to be adapted to a number of different problems from cloud-segmentation [13] to multi-classification problems used in UAV's (unmanned aerial vehicles) [14], [15].

One problem with CNN's and, by extension FCNs, is that they are computationally intensive because convolutions try to summarize the features of an image. It tends to be expensive to summarize this data and then learn the correct filter to best summarize this information. Additionally, using transposed convolutions further increases the memory required to solve

978-1-6654-2180-5/21/$31.00 ©2021 IEEE

this problem since it also requires the learning of the correct filters to reevaluate the images in a labelled format. Dedicated hardware such as GPU's (Graphical Processing Units) [16] tend to perform better on these problems since a number of these convolutional operations can be performed in parallel. These, however, tend to be quite expensive and typically require dedicated desktops and proper cooling to work on problems with copious amounts of data.

This becomes a problem when looking at small island countries such as those in the Caribbean. As opposed to first world countries, these islands do not easily have access to GPU desktops. There have been strides with the advent of cloud computing platforms, such as Google Colab, which has allowed machine learning models to be trained on these platforms. However, when it comes to performing relevant testing and comparison of training methods it becomes tedious and, at times, unusable since the training of models are on a distributed platform with usage constraints.

The aim of this study is to compare a deep learning approach to an ensemble of traditional methods of training a classifier to track the rate of deforestation in a Caribbean island. In our paper we define deforestation to be a decrease in the number of pixels that are labelled as forests in a consecutive set of years. In this regard we trained the models on the same desktop computer and compared the training time of each model. Additionally, in order to ensure that the models were performing similarly we looked at using several different metrics but specifically the Jaccard Index. The Jaccard Index was specifically chosen since it is a common technique used to determine the similarity between an original ground truth image and a predicted image [11], [17], [18], [19]. Since these models will be tested on an unlabeled Google Earth image, these metrics help justify the validity of each of the models used to predict the forested areas in the image.

II. Related Work

The idea of using low powered alternatives to deep learning approaches is a problem that is still quite relevant at the present time. This is because many developing countries around the world still have limited access to equipment such as GPU's which are more easily accessible for first world countries. Additionally, some problems have limited access to resources which requires exploiting different attributes about the data it is being implemented on. Take for instance the problem of on-board semantic segmentation in the area of UAV's for aerial images. In this area, persons are restricted by the on-board device they have on the drone. Exploiting the idea that plants have a different NDVI which can be used to distinguish between the soil and plants on a farm, as was done by [20], [21], they used this concept to help label the images on a farm, but they focus on the problem of distinguishing between crops, weeds and background.

THe authors in [20] implemented the deep convolutional encoder-decoder model Segnet [19]. They utilized an experiment field with varying levels of herbicide hence they had a row of crops alone, a row of weeds alone and a row of crops and weeds. They trained the Segnet model on the crop only and weed only rows and then did testing on the mixed crop and weed plot. However, for [21] they exploited the fact that most crops are planted in a line and utilized this idea to distinguish between weeds and crops. They used a line detection algorithm to determine the row of the crops, they then used the distance of the crops and weeds from the line detected and fed this information into a random forest algorithm to label the plants as either weed or crops. We can see how depending on the problem certain features about a dataset can be utilized to help reduce the complexity of the model being used. In this case instead of using a deep learning algorithm, a traditional machine learning algorithm can be used instead.

Additionally [22] looks at comparing popular segmentation algorithms such as U-Net, FCN and Fmask [12], [11], [23] to name a few and proposed smaller put still highly accurate models that use the concepts of depth-wise separable convolutions and deconvolutions to help with the segmentation of the images used for the problems of cloud detection and forest segmentation. We propose that you can go further and use traditional machine learning models such as linear SVMs, Naïve Bayes and Logistic Regression.

III. Analysis

A. Dataset

We first gathered images from EO-Browser [24] and specifically used the Landsat 8 USGS satellite. We selected countries that have similar climates to the Caribbean or ensured that the images were generally taken during the summertime since most Caribbean islands are warm and humid. Some of the locations from which we took images ranged from New Zealand, Belize, Jamaica and New York. One of our goals was to have a mixture of forested areas and urban areas. However, since larger Caribbean islands have a larger number of forests, a few of the training images have larger amounts of forest labels.

The images that were used were based on bands 4,3,2 which are full color RGB images, and we also used the Normalized Difference Vegetation Index (NDVI) which served as our basis for determining what we would classify as a forest and not a forest. Using a similar approach as given in [25] we used the NDVI to label our images with any NDVI value above 0 being a forest and any value less than or equal to 0 to be labeled as not a forest. Hence cities and houses were labeled as not a forest. These NDVI images, using the threshold just described, was then used to create the ground truth for the training and test images.

The dataset contains 37 RGB images and 37 corresponding NDVI images, the NDVI images were used to create the ground truth images. Due to the input dimensions required for the FCN model we converted the training images to patches of size 384 x 384 pixels which gave us 3000 images. We implemented 5-fold cross validation which resulted in using 2400 images for training and 600 images for testing for each fold.

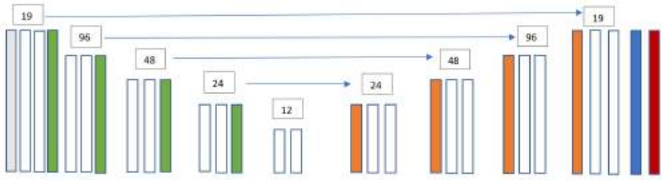

Fig. 1. Grey:input, white: conv3x3 + ReLU, Green: 2x2 max pool, orange:conv tranpose 2x2, blue: conv1x1 sigmoid, red: output

TABLE I
LABEL STATISTICS FOR FOLDS

Fold	Forest Labeled	Non-Forest Labeled
1	83	17
2	90	10
3	68	32
4	83	17
5	84	16

B. Fully Convolutional Neural Network

Because satellite images tend to be quite large, we had to go about creating smaller patches of the images to feed into our FCN. This involved creating patches of size 384 x 384 pixels and resizing them to 192 x 192 pixels which is based on the approach from [13]. The FCN that we used was based upon [12] but we changed the input size of the images to 192, since the original size of 512 would be too computationally expensive to train on our current GPU. The data augmentation techniques used for training our U-net model is used from [13] which involved using techniques such as rotation, flipping and zooming of the images. After training the model and then testing it on each fold we then output the predictions. This was then fed into a script which would produce each of the 192 x 192 patches of the predicted labels of the images and then resize them to 384 x 384.

We can see from Figure 1 the FCN model used has an encoding part with 10 convolutions each having a kernel size of 3 x 3 and using a Rectified Linear Unit (ReLU) activation function. For 8 of the convolutions, they were followed by a 2 x 2 max pooling operation and dropout operation with a value of 0.1, while for the final 2 convolutions there was no max pooling or dropout. For the decoder section of the FCN, we used 4 transposed convolutions with a 3 x 3 kernel, strides of 2 x 2. The transposed convolutions are then followed by concatenation, dropout and convolutions like the ones used in the encoding part of the FCN. After the 4 transposed convolutions, the output is then fed into a final convolution with a 1 x 1 kernel with a sigmoid activation function.

C. Traditional Approaches

Next, we looked at using common machine learning algorithms such as, Logistic Regression, Linear Support Vector Machine (SVM) and Naive Bayes. Since this is a binary classification problem we used Stochastic Gradient Descent to optimize our objective function. Because we wanted to do similar testing, we ensured that the images that were loaded into the traditional models were the same sized images as the FCN. Since the dataset can be quite large, we decided to use "`partial_fit`" in our training since this allowed us to load a portion of the dataset into memory but at the same time the model could be trained on this partial dataset.

For the traditional models, we used input patches of size 384 x 384. However, since the models are less computationally expensive than the FCN we do not need to resize the input. Hence, we go about combining around a thousand patches and then feed it into the model for one epoch, which helps reduce the amount of training time.

For the training of the FCN and traditional models, we made sure to record the time it took to train each model and this was done on a desktop computer with a Nvidia TITAN RTX with a Intel i9-9900K CPU. Additionally, we wanted to determine if the models would be able to run on a cheaper laptop device which involved using a Radeon R7 M260/M265 with a Intel i7-7500U CPU. But when trying to train the FCN the laptop was not able to train the model. However, when training the traditional models, they were able to get results quite similar to the results we obtained using the desktop computer. As discussed earlier, we used a number of metrics to evaluate our results namely, accuracy, precision, recall, specificity, F-score and the Jaccard Index.

IV. RESULTS AND DISCUSSION

Table I shows the composition of the test-set and we can see that the dataset is mainly composed of forests. Hence, when looking at the results we focused on the Jaccard Index [11], [13], [22] which is a common metric that is used for semantic segmentation problems.

Table II shows the results after testing the different models across the 5-folds where the numbers are the average values across the different folds. We can see that overall, for the Jaccard Index, the FCN model performs best. However, the other models results are quite close to the FCN result. Furthermore, if we consider the large imbalance in the dataset, we can see that with respect to the specificity (True Negative Rate) and the recall (True Positive Rate), that the FCN is better at determining the forests with a recall of 97.6%. However when it comes to detecting not forests, for example buildings, the linear SVM tends to perform better with a specificity of 79.4%.

Table III looks at the different models and compares the amount of storage and average training time across the 5-fold validation. We see that the FCN requires 14.3 MB which is significantly larger than the space required by the traditional models. Additionally, if we were to increase the input size of the FCN and, by extension the convolutions and transposed convolutions, in order to achieve better results the training time and memory required would be further increased. Hence, a model such as the Logistic Regression model is presented as a much more efficient solution since it requires significantly less space on a personal computer, and it may be trained on either a desktop or a laptop as the results show.

TABLE II
PERFORMANCE RESULTS

Model	Accuracy	Recall	Specificity	Precision	F-Score	Jaccard Index
Fully Convolutional Neural Network	92.6	97.6	70	93.4	95.6	91.8
Logistic Regression Model (Desktop)	90.2	93.6	76.6	94	93.8	88.6
Logistic Regression Model (Laptop)	91	93.8	77.6	91.6	94	90
Linear Support Vector Machine (Desktop)	90.8	92.7	79.4	95.6	94.1	89
Linear Support Vector Machine (Laptop)	89.6	93.7	75.4	93.5	93.5	88
Naive Bayes (Desktop)	89.7	96.1	60.5	91.3	93.6	88.2
Naive Bayes (Laptop)	89.7	96.1	60.5	91.3	93.6	88.2

TABLE III
COMPUTATIONAL RESOURCE COMPARISON

Model	Storage (MB)	Average Training Time (Hours)
Fully Convolutional Neural Network	14.3	01:32
Logistic Regression Model (Desktop)	0.00105	00:03
Logistic Regression Model (Laptop)	0.001101	00:09
Linear Support Vector Machine (Desktop)	0.000945	00:02
Linear Support Vector Machine (Laptop)	0.0011	00:07
Naive Bayes (Desktop)	0.000645	00:02
Naive Bayes (Laptop)	0.000751	00:06

A review of the training time of the FCN model showed that it took, on average, 1 hour and 32 minutes to train each fold resulting in training taking around 5 hours to be completed. Additionally, the GPU overheated resulting in the model being stopped and restarted several times, indicating that the duration for the exercise was longer than the reported training times. When we compare this to the traditional machine learning models such as the logistic regression model, we see that the average times are 3 minutes on the desktop and 9 minutes on the laptop which is significantly shorter.

As we can see from Figure 2 the FCN tends to perform better than the traditional machine learning models since it is able to better understand the smaller details in the full color images. However, the difference with traditional methods tend to be small as seen in Table II.

Figure 3 shows the predictions made by the logistic regression models and the FCN model on a Google Earth image taken of Trinidad in 2016. As we can see, the models obtained similar shapes in terms of where the deforested areas are such as major towns. We cannot however automatically label the full color image and obtain the ground truth since Google Earth images do not have the NDVI labels of these islands. But when looking at the full color images against the predictions we can see that the models are able to get a good approximation of the non-forested areas. This can be quite useful for Caribbean islands which do not have easy access to Landsat images of their islands but want to have an idea of how deforestation has been progressing over time. Our method of using a logistic regression model is a cheap and easy solution to implement and proves to be quite reliable.

Figure 4 indicates the percentage of pixels that are labelled as forests for Trinidad from 2010 to 2016 using images taken from Google maps where the images were inputted into our different machine learning models. We can see from the graph that the FCN detected the most amount of forests and the logistic regression and linear SVM had similar results. However, the Naive Bayes models method performed poorly. Note the gradual reduction in the percentage of forested areas.

V. CONCLUSION

In this paper, we looked at comparing two different types of methods of labelling satellite images of countries, the popular fully convolutional neural network U-net and an ensemble of traditional machine learning models, to label images from Google Earth images for a Caribbean island. We saw that after training both types of models the overall performance of the models were quite similar but the training time of the U-net model was significantly higher and it required significantly more storage. These models were then used to label images of the Caribbean Island Trinidad and it was shown that each of the models had similar classifications with the Naive Bayes model being the only outlier. Furthermore, because smaller islands have either limited or no access to specialized hardware for training complex machine learning models, using a simpler approach that has quite similar performance results would be more practical. For the Trinidad example we computed the percentage of forest over time to see how deforestation was taking place over time. We believe that this approach can be used for periodic monitoring of the island.

Fig. 2. From left: Full Color Image, Ground Truth Image, FCN Prediction, GPU Logistic Regression Prediction, Laptop Logistic Regression Prediction

Fig. 3. From left: Full Color Image, Fully Convolutional Network Prediction, GPU Logistic Regression Prediction, Laptop Logistic Regression Prediction

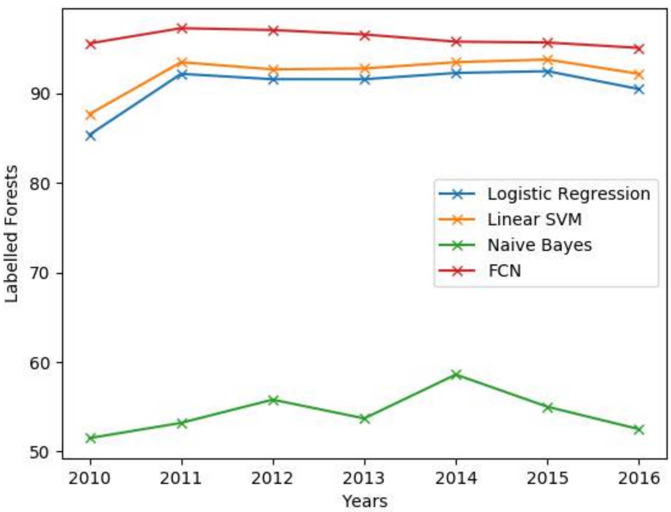

Fig. 4. Predicted percentage of forested areas for Trinidad from 2010 - 2016

References

[1] C. Nunez. Deforestation explained. (accessed: 03.11.2021). [Online]. Available: https://www.nationalgeographic.com/environment/article/deforestation

[2] T. M. Aide, M. L. Clark, H. R. Grau, D. López-Carr, M. A. Levy, D. Redo, M. Bonilla-Moheno, G. Riner, M. J. Andrade-Núñez, and M. Muñiz, "Deforestation and reforestation of l atin a merica and the c aribbean (2001–2010)," *Biotropica*, vol. 45, no. 2, pp. 262–271, 2013.

[3] M. L. Clark and T. M. Aide, "Virtual interpretation of earth web-interface tool (view-it) for collecting land-use/land-cover reference data," *Remote Sensing*, vol. 3, no. 3, pp. 601–620, 2011.

[4] J. Dyson, A. Mancini, E. Frontoni, and P. Zingaretti, "Deep learning for soil and crop segmentation from remotely sensed data," *Remote Sensing*, vol. 11, no. 16, p. 1859, 2019.

[5] Z. Li, H. Shen, Q. Cheng, Y. Liu, S. You, and Z. He, "Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 197–212, 2019.

[6] G. P. Bhattarai, R. B. Schmid, and B. P. McCornack, "Remote sensing data to detect hessian fly infestation in commercial wheat fields," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[7] C. Meneses-Tovar, "Ndvi as indicator of degradation," *Unasylva*, vol. 62, no. 238, pp. 39–46, 2011.

[8] L. C. Alatorre, S. Sánchez-Carrillo, S. Miramontes-Beltrán, R. J. Medina, M. E. Torres-Olave, L. C. Bravo, L. C. Wiebe, A. Granados, D. K. Adams, E. Sánchez *et al.*, "Temporal changes of ndvi for qualitative environmental assessment of mangroves: shrimp farming impact on the health decline of the arid mangroves in the gulf of california (1990–2010)," *Journal of Arid Environments*, vol. 125, pp. 98–109, 2016.

[9] M. M. Aburas, S. H. Abdullah, M. F. Ramli, and Z. H. Ash'aari, "Measuring land cover change in seremban, malaysia using ndvi index," *Procedia Environmental Sciences*, vol. 30, pp. 238–243, 2015.

[10] M. Othman, Z. Ash'Aari, A. Aris, and M. Ramli, "Tropical deforestation monitoring using ndvi from modis satellite: A case study in pahang, malaysia," in *IOP Conference Series: Earth and Environmental Science*, vol. 169, no. 1. IOP Publishing, 2018, p. 012047.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks

for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[13] S. Mohajerani and P. Saeedi, "Cloud-net: An end-to-end cloud detection algorithm for landsat 8 imagery," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 1029–1032.

[14] V. Parmar, N. Bhatia, S. Negi, and M. Suri, "Exploration of optimized semantic segmentation architectures for edge-deployment on drones," *arXiv preprint arXiv:2007.02839*, 2020.

[15] Y. Liu, Q. Ren, J. Geng, M. Ding, and J. Li, "Efficient patch-wise semantic segmentation for large-scale remote sensing images," *Sensors*, vol. 18, no. 10, p. 3232, 2018.

[16] A. Kayid, Y. Khaled, and M. Elmahdy, "Performance of cpus/gpus for deep learning workloads," *The German University in Cairo*, 2018.

[17] S. Mohajerani, R. Asad, K. Abhishek, N. Sharma, A. van Duynhoven, and P. Saeedi, "Cloudmaskgan: A content-aware unpaired image-to-image translation algorithm for remote sensing imagery," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1965–1969.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[20] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. Nieto, and R. Siegwart, "weednet: Dense semantic weed classification using multispectral images and mav for smart farming," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 588–595, 2017.

[21] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "Uav-based crop and weed classification for smart farming," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3024–3031.

[22] G. Bahl, L. Daniel, M. Moretti, and F. Lafarge, "Low-power neural networks for semantic segmentation of satellite images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[23] Z. Zhu, S. Wang, and C. E. Woodcock, "Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images," *Remote Sensing of Environment*, vol. 159, pp. 269–277, 2015.

[24] Sentinel-Hub, "Eo-browser," data retrieved from EO-Browser, accessed: 10.6.2021, https://apps.sentinel-hub.com/eo-browser/.

[25] Y. Liu, W. Gong, X. Hu, and J. Gong, "Forest type identification with random forest using sentinel-1a, sentinel-2a, multi-temporal landsat-8 and dem data," *Remote Sensing*, vol. 10, no. 6, p. 946, 2018.