Applications of computer vision to population dynamics: detecting flowering trees in high-resolution cube-sat imagery

Milla Shin
Brown University
15th May 2020
Advisor: James Tompkin

Abstract

The recent advances and widespread availability of high-resolution satellite imagery and other remote sensing data have provided new avenues and applications for image analysis techniques. Satellite imagery is particularly useful in biological contexts, such as quantifying population dynamics and assessing biodiversity for ecosystem conservation. However, remote sensing imagery analysis still poses many challenges, and more research must be done to create effective and efficient computer-aided tools to best assist biologists. The present study focuses on the automatic mapping of flowering trees in the Amazon rainforest to analyze population dynamics. The method uses a convolutional neural network (CNN) to classify flowering trees and a sliding window approach to make individual pixel predictions on whole images. Due to the lack of sufficient labelled data, two approaches are taken to evaluate the CNN. The first involves training, validating, and testing on two 8000x8000 pixel satellite images. The second involves training on only one satellite image, and testing on the other. The results show that the proposed CNN currently does not have enough training data to generalize on other satellite images, but it has high potential for efficiently automating the process of tree mapping if more labelled data is supplied. These findings also demonstrate the possibility of applying deep learning to satellite imagery analysis in general.

1. Introduction

Recent advances in remote sensing technologies have greatly advanced our understanding of Earth's surface and ecosystems across large spatial gradients [13]. Satellite sensors, for example, can track a variety of information, including panchromatic, optical/infrared, thermal infrared, and radar signals, and are now able to generate data at scales of time and space aligned with biological processes [15, 13]. Remote sensing has thus been applied to a variety of biological contexts, such as population dynamics, ecosystem and biodiversity conservation, and high-spatial-resolution phenology [13]. Manually analyzing these large and complex datasets is often infeasible, so computer vision and machine learning techniques are needed to efficiently and automatically annotate these images [26, 28].

The Kellner Lab at Brown University is currently investigating the use of satellite imagery to quantify population dynamics. The images are obtained from constellations of miniature satellites called CubeSats, which work together to capture Earth's entire land surface at 1-3m resolution every day [13]. They are specifically working on mapping trees in the genus *Handroanthusmap* in the Amazon rainforest. These trees are particularly interesting and useful because they exhibit conspicuous flowering patterns that can be captured from space. Only for a few days each year, these trees produce vibrant floral displays that indicate the individual is alive and ready to reproduce. By tracking these flowering displays over successive years and across vast geographic areas, we can better understand how forests are changing over time.

The goal of this project is to help analyze these high-resolution image time series by automating the process of individual tree detection. Currently, they have explored several methods; the two they took the furthest were: 1.) generating polygons around tree objects, and then classifying each object based on the minimum spectral angles of the collection of pixels, and 2.) running a principal component analysis (PCA) on the 4-band satellite image, then using a decision tree based on the PCs. However, both approaches require prior knowledge for detection and manual work by the researcher, which can get labor intensive and infeasible when working with larger amounts of data.

In this project, I explore a deep learning approach, where I train a classifier for tree detection with a convolutional neural network (CNN). Training was conducted in two ways: using a single 8000x8000 pixel satellite image and using both satellite images. Evaluation was conducted by comparing predictions to ground truths marked by individuals from the

Kellner Lab. The lack of training data significantly limited the performance and evaluation methods of the CNN, but the results present a preliminary classifier that shows potential for efficiently automating the process of tree mapping.

In addition to the biological applications, this project also makes contributions to the field of computer vision and image analysis. The use of deep learning and CNNs on satellite imagery is still a relatively new field, and a lot of current research focuses on land coverage analysis through object detection, semantic segmentation, and image classification [20, 9, 18, 16]. However, there are many challenges associated with using a deep learning approach on satellite imagery; for example, algorithms must take into account the high resolution and spatial complexity of images, and unlike traditional datasets like ImageNet, where objects take up the majority of the image, objects in satellite images are small and often densely grouped. There is also a lack of sufficiently annotated images for training, especially those that are labelled pixel-by-pixel [14, 20, 6]. Lastly, images are often affected by different atmospheric conditions like cloud cover [17]. Thus, satellite imagery is an exciting new avenue for computer vision and biology research alike [15, 26].

2. Related Work

Various machine learning algorithms have been used to classify satellite images and produce feature maps of land use. More recently, however, deep learning and CNN-based approaches have shown great potential to outperform these traditional techniques. The types of deep learning methods applied to satellite imagery can largely be divided into three main categories: land surface classification, semantic segmentation, and object detection. Classification aims to assign labels to entire scenes, while segmentation aims to produce feature maps that assign a class to each pixel. Object detection aims to create "bounding boxes" around parts of the image that correspond to different labels [2].

2.1. Satellite Image Object Detection

Object detection in satellite imagery remains a very difficult task, and existing object detection methods cannot be directly applied. The large input sizes of satellite images often make computation too slow for practical use, tiny objects are difficult to detect, and complex backgrounds cause a significant amount of false alarms [21, 6]. Etten, for example, showed that performance is extremely poor when applying YOLO, a standard object detection network architecture, to 416x416 pixel cutouts of satellite images of cars [6]..

Thus, new architectures must be created for accurate object detection. Pang et al. prposed R^2-CNN, a unified and self-reinforced CNN, which joins a classifier used to predict the existence of targets in each patch with a detector used to locate these targets accurately [21]. Etten also created a new network, YOLT, which is optimized for small, densely packed objects. The pipeline was used to detect both smaller objects, such as boats and airplanes, and larger objects, such as airports and roads. Etten found that the pipeline yields an object detection F1 score of approximately 0.6 - 0.9 if the model is trained separately for small and large objects [6].

2.2. Satellite Image Classification

Classification tasks in satellite imagery analysis involve labelling images based on land cover types, such as "agriculture", "water", and "road." Other tasks involve analyzing atmospheric conditions, sorting images into categories such as "partly cloudy", "hazy", or "clear" [22, 16].

Numerous studies have found that CNNs can be used to classify satellite images with high accuracy. For example, Rakshit et al. achieved a testing accuracy of 96.71% by adapting the VGG model architecture to classify images from the Amazon rainforest. The images were 128x128 pixels with 3 color bands and could be labelled into multiple categories that described the land cover type, as well as the atmospheric condition [22]. Kussul et al. compared an ensemble of multilayer perceptrons, a random forest approach, and a CNN to classify land coverage and crop types, and found that the CNN performed the best. For the CNN, they used a sliding window approach with a 1-pixel step size and 7×7 pixel window size to assign classes to the central pixel of each sliding window. They attributed the CNN's success to its ability to "build a hierarchy of local and sparse features" as opposed to a "global transformation of features" [16].

2.3. Satellite Image Segmentation

Previous studies show two main approaches to using CNNs for image segmentation: a patch-based approach and a pixel-to-pixel semantic segmentation approach [12].

The patch-based approach first creates smaller patches from the input images. The classifier is trained to label the center pixel of each patch. Then, a sliding window approach is used to make predictions on each pixel of the entire image [12]. The task is similar to the image classification technique described above, but requires additional pre-processing to generate training data and post-processing to combine pixel-based predictions. The previously described study by Kussul et al. is an example of this technique [16].

The second is based on fully convolutional networks (FCNs) [14, 12]. This approach replaces fully-connected layers at the end of a neural network with convolutional layers, so that the output has the same shape as the original input image. The result is a feature map with category predictions for each pixel [24]. Napiorkowska et al. demonstrated that a VGG network, combined with FCN layers, can be used to detect roads, palm trees and cars in images from Deimos-2 and Worldview-3 satellite images. They were able to achieve accuracies as high as 98-99%, outperforming more common techniques in remote sensing such as Random Forest

or Support Vector Machines [20]. Other papers have also tackled satellite image segmentation with a FCN approach [14, 2, 9]. Khryashchev et al. compared three different FCN architectures, U-Net, SegNet, and LinkNet, to compare image segmentation performance for distinguishing between classes such as "forest", "crops", and "water". They found that all models displayed high accuracy results [14].

2.4. General Approaches for Object Detection, Classification, and Semantic Segmentation

Deep learning has many applications outside of satellite imagery and tree mapping problems, and some of these techniques can be applied to the problems addressed in this project. Jimenez and Racoceanu used two deep learning approaches to detect and classify mitosis in histopathological tissue samples for breast cancer diagnosis. The first method, a classification based method, involved a pre-processing step of creating a blue ratio image to detect potential mitosis and then extracting them as 71x71-pixel patches. These patches were used as inputs to a fine-tuned version of AlexNet for binary classification. The second approach, a segmentation based method, used the U-Net architecture. Both methods outperform classical image processing techniques. They argue that the U-Net approach requires further analysis to improve border detection, but has advantages over AlexNet in that it eliminates the need for pre/post-processing [10].

Similar to the AlexNet technique used by Jimenez and Racoceanu, Haehn et al. extracted smaller 75x75 pixel patches from a larger image, and used these patches as inputs to a CNN to perform a binary classification task. The goal of the project was to reduce boundary errors generated from automatic segmentation and classification of brain tissue. The patches were created over the center of an existing boundary and were labelled as having either a correct or erroneous boundary. Jimenez and Racoceanu raised the issue of high computational cost involved in the post-processing step of patch-based classification approaches [10], but Haehn et al. avoid this issue by only running the CNN on cell boundaries, rather than analyzing every pixel [7].

2.5. Tree Mapping Using Satellite Imagery

Most of the current tree mapping approaches rely on hand-crafted features and manual work by the researcher [23, 1, 27]. For example, Rizvi et al. used an object based image analysis (OBIA) method for agroforestry mapping, which involved an in-depth understanding of the spectral information of trees [23]. Alganci et al. used a similar method as the techniques proposed by the Kellner Lab to determine the spatial distribution of olive trees. Their method involved using geometric correction and a decision-tree classification approach that integrated spectral properties of the image [1].

Deep learning approaches have only recently been used for tree mapping. Most of these studies have taken a patch-based classification approach: smaller samples are collected using a sliding window technique, and detection results are obtained by merging the coordinates of the trees from individual predictions. Both Li et al. and Bhattacharyya et al. demonstrated high accuracy results using this technique for detecting and counting oil palm trees and shade trees, respectively. In both of these studies, predictions were more difficult because the study area was densely populated, and the tree crowns often overlapped [18, 3]. Sylvain et al. used CNNs to detect and map tree health status and functional type, evaluating the effect of window size, spectral channel selection, and ensemble learning on classification accuracy. The researchers found that channel size had a limited effect, but larger window sizes led to better predictions. Aggregating multiple predictions using the ensemble approach also increased classification accuracy [25].

2.6. Summary

Deep learning approaches have shown to outperform traditional machine learning techniques, but more work still must be done to efficiently and accurately analyze satellite imagery. All three of the main satellite imagery analysis techniques mentioned above–object detection, classification, and segmentation–can be associated to tree mapping, but based on the works discussed, image segmentation methods seem the most promising and relevant. Previous papers have explored two main approaches for semantic segmentation: a patch-based classification approach and a pixel-to-pixel FCN approach. These techniques have not only been used for satellite imagery analysis, but also for other areas like boundary correction in connectomics and mitosis detection in tissue samples [10, 7]. Many of the recent tree mapping studies have shown successful results using the patch-based technique. This project attempts to build on these previous studies and apply deep learning to flowering tree detection in the Amazon using the patch-based semantic segmentation approach.

3. Data

In this study, two analytic Ortho tile images acquired on August 17, 2016 from the PlanetScope Satellite are used. Each image covers a single 25x25km (8000x8000 pixel) grid cell from Rondônia, Brazil and comes with 4 multispectral bands (blue, green, red, near-infrared) 1. These images were chosen because they are part of a time series during which some of the flowers emerge and disappear. In addition, the grid cell covers non-forested areas, which the model can learn to distinguish from the forested areas.

Planet Labs [2] creates orthorectified tile images by collecting a series of overlapping consecutive scenes from a single

[2] https://assets.planet.com/docs/Planet_Combined_Imagery_Product_Specs_letter_screen.pdf

satellite in a single pass. These images are radiometrically-, sensor-, and geometrically-corrected and aligned to a cartographic map projection.

The dataset is also extremely imbalanced. Flowering tress make up a very small proportion of the overall image, so there are significantly more negative than positive samples. The imbalance ratio (IR), or skew, is often used to measure the level of imbalance. However, Luque et al. proposed a new measure, the imbalance coefficient, which is more intuitive as values lie within the range of $[-1, 1]$, with $\delta = 0$ indicating a balanced dataset. The imbalance coefficient, denoted δ, is calculated as follows:

$$\delta = 2 * \frac{m_p}{m} - 1$$

where m_p is the number of positive samples and m is the total number of all samples [19]. The first satellite image has 12,119 positive and 63,987,881 negative pixels, and the second image has 6,518 positive and 63,993,482 negative pixels. Both result in $\delta \approx -1$, indicating an extreme imbalance toward the negative class. This has several implications for the creation and evaluation of the CNN, which will be discussed later on.

4. Method

I use a patch-based segmentation approach to detect individual trees in the images. I first build a CNN trained to classify the center pixel of each patch as either positive (containing a flowering tree) or negative (not containing flowering tree).

Due to the lack of labelled data, I experiment with two approaches for training the CNN. First, I combine patches from both satellite images and train the CNN on 70% of this dataset, validate on 15%, and test on the remaining 15%. The second experiment uses patches only from the first satellite image, then tests on the patches generated from the second image, in order to determine how well the first image can generalize to the second. For the second experiment, the CNN is used to predict labels for each pixel using a sliding window approach.

4.1. Data Preprocessing and Labels

The two satellite images came with corresponding labels created by individuals from the Kellner Lab. The labelled images contain green pixels at locations with flowering trees and black pixels everywhere else; this was converted to an array of 1s and 0s, representing flowering trees and non-flowering trees, respectively.

To generate training samples to feed as inputs to the CNN, all of the coordinates of the green pixels were identified from the labelled images, and 25x25 pixel patches were created from the corresponding analytic satellite image, with each green pixel at its center. This resulted in 12088 positive samples for the first image. To create a more balanced dataset, 12088 pixel locations were chosen at random to create the 25x25 negative sample patches. For the second image, 6518 positive samples were obtained and 6518 negative samples were chosen at random. An undersampling approach was chosen due to the extreme class imbalance. Buda et al. found that oversampling performs better in all cases except for when there is an extreme class imbalance ratio, in which case undersampling performs on par with oversampling while significantly reducing training time [4].

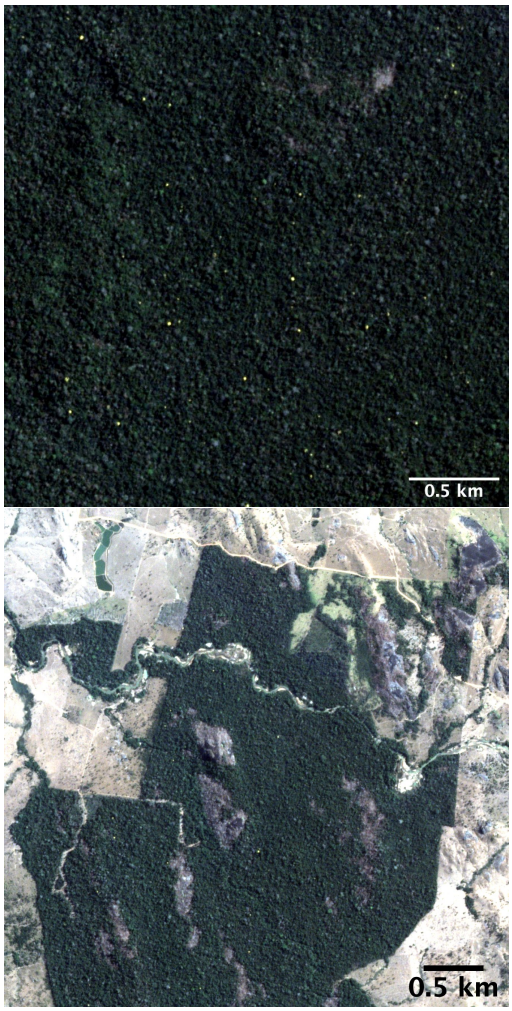

Figure 1. Zoom up examples of the high-resolution Ortho tile images from the PlanetScope Satellite. The images capture individual flowering trees in the Amazon forest (yellow objects) and cover both forested and nonforested areas. The examples shown are the visual (RGB) version of the analytic (RGB and near-infrared) image used for training.

The 25x25 pixel dimension was chosen to be large enough to encompass an entire tree (most of the trees only span about 15x15 pixels), as recommended by Sylvain et al.'s study [25]. This chosen patch size allowed the network

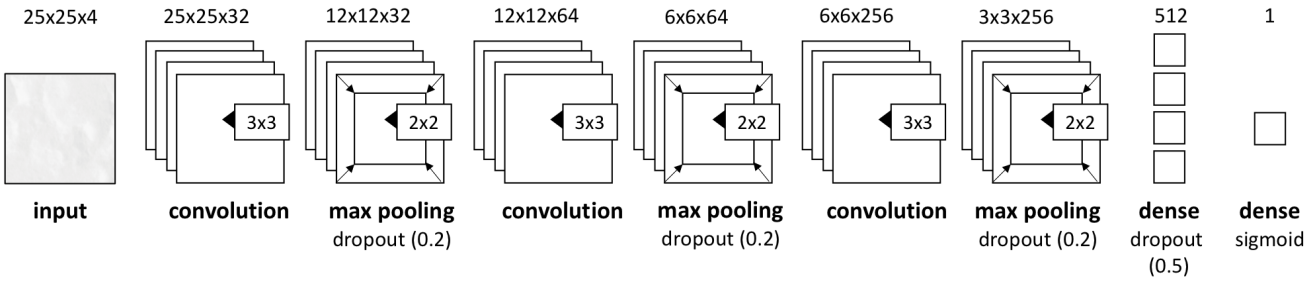

Figure 2. The CNN architecture consists of three convolutional layers with max pooling and dropout regularization. The output is the probability of finding a flowering tree in the central pixel of the input image patch.

to obtain enough contextual information around the trees while still maintaining an efficient computing time. I also tested a 75x75 pixel patch size, but found that this only increased training and testing time while decreasing classifier performance.

Due to the lack of sufficient training data, augmentation was performed on the training set. Three different augmentations were applied: 1.) rotating the patches randomly in four different angles (0°, 90°, 180°, and 270°). 2.) flipping the images randomly up/down or left/right 3.) applying both rotation and flip.

4.2. CNN Architecture

Transfer learning is the process of using weights from a network pre-trained on a larger dataset, and applying it to a smaller dataset by fine-tuning. Although this approach has shown to be very successful, the pre-trained networks generally only accept 3 band (RGB) images as inputs, which differs from the 4 band (RGBI) satellite images used in this study. In addition, the networks are trained on common image datasets like ImageNet, which may have significantly different features from satellite images [12, 8]. Thus, this study implements a CNN from scratch, inspired by previously studied models.

I explored several different architectures. The final CNN configuration has three convolutional layers, each followed by max pooling with dropout regularization to prevent overfitting 2. Each convolutional layer, except the last one, is batch normalized with a leaky rectified linear activation (ReLU). The final layer uses a sigmoid function to generate binary predictions. The CNN was implemented using Keras and Tensorflow.

4.3. Classifier Training

The Adam optimizer was used to minimize loss with a learning rate of 0.001. Loss was measured by the binary cross-entropy loss function. A batch size of 124 was used.

For the first experiment, 25x25 pixel patches from both

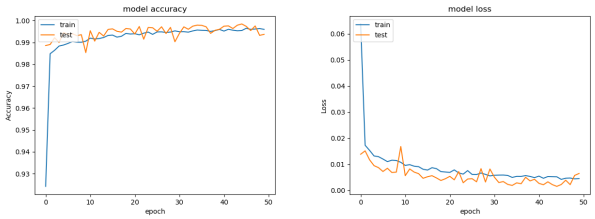

Figure 3. Performance curves from experiment 1, using both satellite images for training and testing. *Left:* training and testing accuracies. *Right:* training and testing loss.

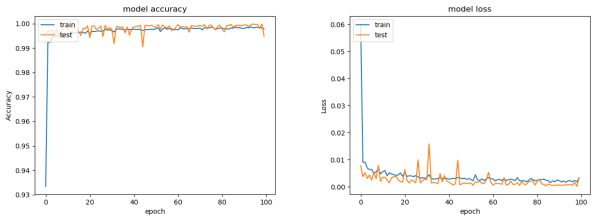

Figure 4. Performance curves from experiment 2, using only one satellite image for training and testing. *Left:* training and testing accuracies. *Right:* training and testing loss.

satellite images were used to train the CNN. The dataset was shuffled and split into a ratio of 70-15-15 for training, validation, and testing. The model trained for 50 epochs 3.

For the second experiment, 25x25 pixel patches from only one satellite image were used for training and testing the CNN. The training dataset was shuffled and split into 70% for training and 30% for testing. The model trained for 100 epochs but achieved a high accuracy and low loss already from around epoch 10 4.

4.4. Classification Task

The resulting CNN is able to provide a class label for every 25x25 pixel patch. Thus, in order to obtain a full map of predicted locations of flowering trees for a given satellite image, a sliding window approach is used with a

step size of 1 pixel. The decision of whether to label the center pixel of each patch as a flowering tree or not is taken by comparing the output of the CNN to a threshold of 0.5; a value greater than or equal to 0.5 represents a positive prediction, indicating the existence of a flowering tree.

The classification task was only performed on the second satellite image using the CNN trained in experiment 2. This step was not performed for the first experiment, since both satellite images were used for training and testing the CNN.

Ideally, if more labelled data were available, the classification task would be performed on more images to better evaluate the performance of the CNN.

5. Results

5.1. Evaluation Method and Metrics

To quantitatively evaluate the performance of the CNN, the following metrics are calculated: accuracy, precision, sensitivity, specificity, F1 score, geometric mean (GM), and Matthews correlation coefficient (MCC).

The choice of an appropriate metric was an important consideration, especially due to the extreme imbalance of the dataset. While most authors use accuracy and F1 score, recent papers have shown that these performance metrics are highly biased and often show overoptimistic inflated results, especially on imbalanced datasets [19, 5]. Consider accuracy, for example, on an image where a very small percentage of the pixels have positive labels–a prediction of all 0's would still result in a very high accuracy. F1 score is similarly biased; it varies if the majority and minority classes are swapped, and is also independent of the true negatives [5].

Luque et al. argue that the best performance metrics are sensitivity, specificity, and geometric mean, because they do not have bias due to imbalance. However, these measures only focus on the classification successes as opposed to the errors, so if errors must also be considered, MCC is the next metric with the lowest bias [19]. MCC incorporates both dataset imbalance and invariance for class swapping, taking into account all four values in the confusion matrix [5]. While the first experiment avoids testing on an imbalanced dataset using undersampling, the second experiment involves making predictions on the entire satellite image. Thus, I provide results for all of these metrics, defined as follows:

$$\text{Total Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2}$$

$$\text{Recall / Sensitivity} = \frac{TP}{TP + FN}, \tag{3}$$

Metric	Value
total accuracy	0.9918
precision	0.9837
recall/sensitivity	1.0000
specificity	0.9836
F1 score	0.9918
GM	0.9918
MCC	0.9837

Table 1. Evaluation metrics tested on 15% of the dataset for the first experiment. Training and testing were done using 70% and 15% of the dataset, respectively, which consisted of 25x25 pixel patches from both satellite images.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

$$\text{F1 Score} = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

$$\text{GM} = \sqrt{sensitivity * specificity} \tag{6}$$

$$\text{MCC} = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{7}$$

For the first experiment, the performance of the CNN was tested on 15% of the dataset, which consisted of patches from both satellite images. Testing was done at the end of training, using weights from epoch 50 1. Examples of predictions versus real labels are shown in Figure 5.

For the second experiment, the entire second 8000x8000 pixel satellite image was used for evaluation. This image contains 6,518 positive pixels and 63,993,482 negative pixels. The model converges from very early on, as shown in Figure 4. Through testing on a small region of the image, epoch 50 was found to produce the best results; thus, the weights saved from epoch 50 were used to make predictions. Predictions were made on each 25x25 pixel patch using the classification method described earlier. As the CNN requires a 25x25 patch, the borders of the image were not used. Table 2 shows the results of the calculated metrics 2. Examples of predictions versus real labels are shown in Figure 6.

6. Discussion

The model is able to fit very well in both experiments; the performance curves indicate that the model achieves a low loss and high accuracy from very early on in the training phase 3, 4. In addition, the evaluation metrics from testing in the first experiment are all extremely high 1.

However, the performance metrics from the second experiment are much lower 2. Total accuracy is high, but this tells us little about the performance of the CNN because of

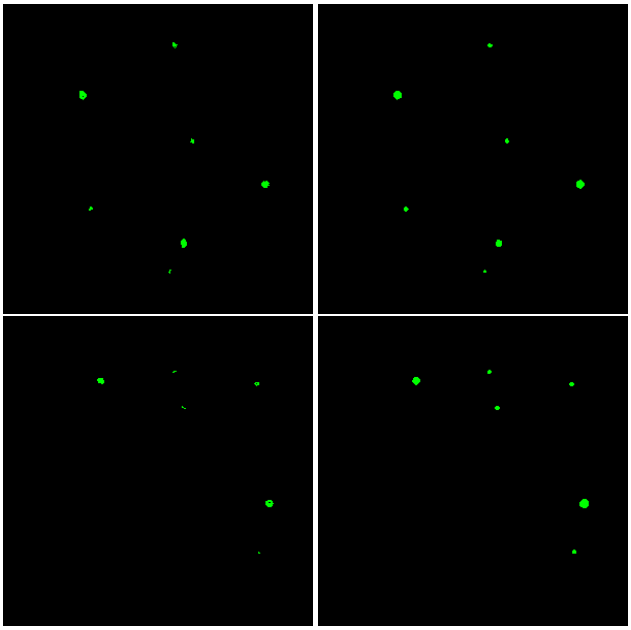

Figure 5. Examples of predicted tree locations from small 300x300 pixel regions in the second satellite image, generated from experiment 1. The left images are the actual labels, and the right are predicted locations. *Top:* locations of trees are well predicted, with sensitivity = 0.98, specificity = 1.00, GM = 0.99, MCC = 0.85. *Bottom:* locations are similarly well predicted in another region, with sensitivity = 0.97, specificity = 1.00, GM = 0.98, MCC = 0.74. The MCC scores are not as high due to the lower precision scores, but overall the model predicts locations of the trees almost perfectly.

Metric	Value
total accuracy	0.9998
precision	0.2662
recall/sensitivity	0.6563
specificity	0.9998
F1 score	0.3788
GM	0.8100
MCC	0.4179

Table 2. Evaluation metrics tested on the second satellite image for the second experiment. Training and validation were done using 70% and 30% of the dataset, respectively, which consisted of patches only from the first satellite image.

the extreme imbalance of the dataset. The high specificity shows that the classifier is able to identify true negatives well, but the lower sensitivity indicates that the classifier failed to identify the positive patches well.

These results can be attributed to the lack of sufficient labelled data. The first experiment shows that using all of the data from both satellite images can make predictions with very high accuracy, but using only one satellite image is not enough to generalize to other images. There is simply not

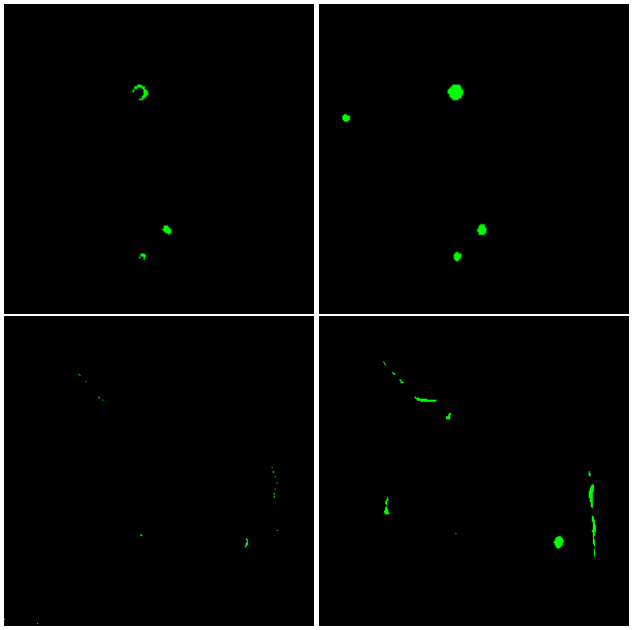

Figure 6. Examples of predicted tree locations from a small 300x300 pixel region in the second satellite image, generated from experiment 2. The left images are the actual labels, and the right are predicted locations. *Top:* a better predicted region, with sensitivity = 0.99, specificity = 1.00, GM = 0.99, MCC = 0.65. *Bottom:* a badly predicted region, with sensitivity = 0.64, specificity = 1.0, GM = 0.80, MC = 0.20. The general locations of the trees are not predicted poorly, but the CNN over-classifies the regions surrounding the trees.

enough complexity and variety in a single image, although it generated more than 12,000 positive samples. Despite this lack of data, however, the model did not perform too poorly, with a geometric mean of 0.81 and a MCC of 0.42 when testing on the second image 2. The example results of the classification task in Fig. 6 also show a well-predicted and poorly predicted region. Surprisingly, many of the general tree locations were actually predicted correctly.

These results reflect an underlying challenge with the application of deep learning to satellite imagery. Deep learning requires very large datasets, but there is a lack of sufficiently annotated satellite images, especially those that are labelled pixel-by-pixel [14, 20, 6]. Another issue is the extreme imbalance when detecting sparse, tiny objects or rare events, like the flowering of trees that only occur for a few days each year. Juba and Le found that in highly imbalanced datasets, the only way to achieve high precision and recall is to use a large amount of data. None of the tested imbalance-correcting methods, such as oversampling or undersampling, were effective in increasing precision and recall. If a large enough training set is not available, they recommend "exploiting some kind of prior knowledge about the domain" to create an effective classifier [11].

Thus, the most promising way to achieve a higher accu-

racy and correctly predict the locations of flowering trees using a deep learning approach would be to train on a large enough dataset. This might be challenging, however, as labelling images takes a lot of manual labor and time, and the amount of labelled data needed to generalize to other areas in the Amazon, or even other satellite images in general, could be too large to be feasible. Thus, unless more labelled data can be obtained, a combined approach of using hand crafted features along with the features extracted from CNNs could be useful.

7. Conclusion

In this project, I have explored a deep learning approach to detect flowering trees from satellite imagery. A patch-based segmentation approach is used. First, a CNN is trained to classify the center of a 25x25 pixel patch as either containing a flowering tree or not. Then, a sliding window approach with a step size of 1 pixel is used to generate predicted tree locations for an entire image. Due to the lack of sufficient data, two experiments were performed to evaluate the performance of the CNN. The first method used both of the 8000x8000 pixel satellite images and a 70-15-15 split for training, validation, and testing to train and evaluate the CNN. The second method used only one of the satellite images to train the CNN, and performed the classification task on the entirety of the second image.

The results indicate that the CNN was able to fit the data well–the loss and accuracy of the model converged quickly, and all performance metrics for the first experiment, including geometric mean and MCC, achieved values close to 1.00, indicating almost perfect performance. However, the results for the second experiment are significantly worse, with a geometric mean of 0.81 and MCC of 0.42. This suggests that with more labelled images, the presented CNN has potential for accurately predicting the presence and location of flowering trees, but a single satellite image does not have the complexity required for a CNN to generalize and make predictions on other images.

The challenges faced in this study reflect an underlying issue of using deep learning on satellite imagery: the lack of sufficient labelled training data. Obtaining enough data, especially those that are labelled pixel-by-pixel, is a very labor intensive task. Thus, to make the most out of the currently available satellite images, either more labelled images need to be generated, or perhaps classical image-processing methodologies and deep learning approaches can be combined with hand-crafted features to create more efficient classifiers.

Another challenge faced in this study was extreme imbalance of the dataset–the number of negative samples significantly outweighed the positive samples. This project presents an undersampling approach and discusses the ramifications that such an imbalance has on the choice of appropriate evaluation metrics.

This project presents a preliminary CNN-based classifier that shows potential for automating the process of tree mapping using satellite images. Further analysis is needed on more labelled data in order to improve accuracy and fully evaluate the performance of the CNN. In addition, for future study, the time complexity of the classification task could also be considered. The sliding window approach must loop through every pixel and make predictions on each 25x25 pixel patch, so it is not very time-efficient. Perhaps an alternative segmentation approach can be considered, such as the use of fully connected networks to generate entire feature maps more efficiently.

References

[1] U. Alganci, E. Sertel, and S. Kaya. Determination of the olive trees with object based classification of pleiades satellite image. *International Journal of Environment and Geoinformatics*, 5(2):132–139, 2018. 3

[2] V. Alhassan, C. Henry, S. Ramanna, and C. Storie. A deep learning framework for land-use/land-cover mapping and analysis using multispectral satellite imagery. *Neural Computing and Applications*, 2019. 2, 3

[3] A. Bhattacharyya and R. Bhattacharyya. Crown detection and counting using satellite images. In *Emerging Technology in Modelling and Graphics. Advances in Intelligent Systems and Computing*, volume 937, pages 765–773. Springer, 2020. 3

[4] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018. 4

[5] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(6), 2020. 6

[6] A. V. Etten. You only look twice: Rapid multi-scale object detection in satellite imagery. *ArXiv*, abs/1805.09512, 2018. 2, 7

[7] D. Haehn, V. Kaynig, J. Tompkin, J. Lichtman, and H. Pfister. Guided proofreading of automatic segmentations for connectomics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9319–9328, 2018. 3

[8] Z. M. Hamdi, M. Brandmeier, and C. Straub. Forest damage assessment using deep learning on high resolution remote sensing data. *Remote Sensing*, 11(17):1976, 2019. 5

[9] V. Iglovikov, S. Mushinskiy, and V. Osin. Satellite imagery feature detection using deep convolutional neural network: a kaggle competition. *ArXiv*, abs/1706.06169, 2017. 2, 3

[10] G. Jimenez and D. Racoceanu. Deep learning for semantic segmentation vs. classification in computational pathology: Application to mitosis analysis in breast cancer grading. *Frontiers in Bioengineering and Biotechnology*, 7:145, 2019. 3

[11] B. Juba and H. S. Le. Precision-recall versus accuracy and the role of large data sets. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, volume 33, pages 4039–4048, 2019. 7

[12] M. Kampffmeyer, A. Salberg, and R. Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 2016. 2, 5

[13] J. R. Kellner, L. P. Albert, J. T. Burley, and K. C. Cushman. The case for remote sensing of individual plants. *American Journal of Botany*, 106(9):1139–1142, 2019. 1

[14] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, and A. Rubtsov. Comparison of different convolutional neural network architectures for satellite image segmentation. In *2018 23rd Conference of Open Innovations Association (FRUCT)*, pages 172–179. IEEE, 2018. 2, 3, 7

[15] C. Kuenzer, M. Ottinger, M. Wegmann, H. Guo, C. Wang, J. Zhang, S. Dech, and M. Wikelski. Earth observation satellite sensors for biodiversity monitoring: potentials and bottlenecks. *International Journal of Remote Sensing*, 31(18):6599–47, 2014. 1, 2

[16] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017. 2

[17] M. A. LaRue, S. Stapleton, and M. Anderson. Feasibility of using high-resolution satellite imagery to access vertebrate wildlife populations. *Conservation Biology*, 31(1):213–220, 2016. 2

[18] W. Li, H. Fu, L. Yu, and A. Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9(1):22, 2017. 2, 3

[19] A. Luque, A. Carrasco, A. Martin, and A. de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019. 4, 6

[20] M. Napiorkowska, D. Petit, and P. Marti. Three applications of deep learning algorithms for object detection in satellite imagery. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4839–4842. IEEE, 2018. 2, 3, 7

[21] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng. $\mathcal{R}^2$ -cnn: Fast tiny object detection in large-scale remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5512–5524, 2019. 2

[22] S. Rakshit, S. Debnath, and D. Mondal. Identifying land patterns from satellite imagery in amazon rainforest using deep learning. *ArXiv*, abs/1809.00340, 2018. 2

[23] R. H. Rizvi, R. Newaj, S. Srivastava, and M. Yadav. Mapping trees on farmlands using obia method and high resolution satellite data: a case study of koraput district, odisha. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 423:617–621, 2019. 3

[24] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional betworks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 2017. 2

[25] J.-D. Sylvain, G. Drolet, and N. Brown. Mapping dead forest cover using a deep convolutional neural network and digital aerial photography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:14–26, 2019. 3, 4

[26] B. Weinstein. A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3):533–545, 2017. 1, 2

[27] D. Wen, X. Huang, H. Liu, W. Liao, and L. Zhang. Semantic classification of urban trees using very high resolution satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4):1413–1424, 2017. 3

[28] Y. Xue, T. Wang, and A. K. Skidmore. Automatic counting of large mammals from very high resolution panchromatic satellite imagery. *Remote Sensing*, 9(9):878, 2017. 1