An Automated and Scalable ML Solution for Mapping Invasive Species: the Case of the Australian Tree Fern in Hawaiian Forests

Ovidiu Dan Iancu
Amazon Web Services
440 Terry Ave N, Seattle, WA 98109, United States
iovidiu@amazon.com

Kara Yang
Amazon Web Services
440 Terry Ave N, Seattle, WA 98109, United States
karayang@amazon.com

Han Man
Amazon Web Services
440 Terry Ave N, Seattle, WA 98109, United States
hanman@amazon.com

Theresa Cabrera Menard
The Nature Conservancy
4245 North Fairfax Drive, Suite 100 Arlington, Virginia 22203-1606
theresa_menard@tnc.org

Abstract

Biodiversity loss and ecosystem degradation are global challenges demanding creative and scalable solutions. Recent increases in data collection coupled with machine learning have the potential to expand landscape monitoring capabilities. We present a computer vision solution to the problem of identifying invasive species. The Australian Tree Fern (Cyathea cooperi) is a fast growing species that is displacing slower growing native plants across the Hawaiian islands. The Nature Conservancy organization has partnered with Amazon Web Services to develop and test an automated tree fern detection and mapping solution based on imagery collected from fixed wing aircraft. We utilize deep learning to identify tree ferns and map their locations. Distinguishing between invasive and native tree ferns in aerial images is challenging for human experts. We explore techniques such as image embeddings and principal component analysis to assist in the classification. Creating quality training datasets is critical for developing ML solutions. We describe how semi-automated labeling tools can expedite this process. These steps are integrated into an automated cloud native inference pipeline that reduces localization time from weeks to minutes. We further investigate issues encountered when the pipeline is utilized on novel images and a decline in performance relative to the training data is observed. We trace the origin of the problem to a subset of images originating from steep mountain slopes and riverbanks which generate blurring and streaking patterns mistakenly labeled as tree ferns. We propose a preprocessing step based on Haralick texture features which detects and flags images different from the training set. Experimental results show that the proposed method performs well and can potentially enhance the model performance by relabeling and retraining the model iteratively.

1. Introduction

In recent decades the rate of environmental degradation and habitat loss has accelerated [19]. Traditional landscape monitoring procedures and methodologies require in many cases substantial resource and time investments, limiting ecological forecasting at short time scales [4]. The advent of satellite and drone technologies has expanded ecological data collection capacity [12], while ecological data analysis at scale is still an area of active research and development

[5]. High-throughput machine learning (ML) solutions are emerging as one of technologies with the biggest impact on ecological monitoring and forecasting [14].

Hawaiian forests are essential as a source of clean water and for preservation of traditional cultural practices [11], however they face critical threats from deforestation, species extinction, and displacement of native species by invasive plants [21]. The Nature Conservancy (TNC) is helping to address the invasive plant problem through "Citizen Science" initiatives [1] such as the Hawaii Challenge [6], which allows anyone with a computer and Internet access to participate in tagging invasive weeds across the landscape. AWS has partnered with TNC to augment these efforts by contributing expertise in computer vision and cloud resources.

Among the most aggressive species invading the Hawaiian forests is the Australian Tree Fern (ATF), originally introduced as an ornamental but now rapidly spreading across several islands by producing numerous of spores that are easily transported by the wind. The ATF is fast growing and outcompetes other plants, smothering the canopy and affecting several native species, resulting in loss of biological diversity.

Prior to the deployment of the automated pipeline, detection of the tree ferns was accomplished by capturing images from fixed wing aircraft surveying the forest canopy. The imagery is subsequently manually inspected by human labelers, a process that takes significant effort and time, potentially delaying the mitigation efforts by ground crews by weeks or longer. One of the advantages of utilizing an CV algorithm is the potential time savings as the inference time requires only minutes to hours.

The goal of the AWS-TNC partnership was to evaluate the potential of computer vision (CV) algorithms to reliably detect and potentially classify tree ferns by species. A performant CV can then form the basis of a fully automated AWS cloud native solution that enhances the capacity of TNC to efficiently and in a timely manner detect invasive tree ferns and direct resources to highly affected areas.

The availability of labeled data is critical for the performance of any CV algorithm. However, in many cases such data is either not available, of insufficient volume, or the presence of labeling errors hinders the performance of the algorithm. We encounter and address these issues in the present project, leveraging AWS tools such as Amazon SageMaker [13]. At the inception of the project annotation data on the tree fern species (ATF versus native) was not available and furthermore it was questionable whether a human expert could reliably distinguish between tree fern species based solely on overhead images. We investigate the classification problem by utilizing a hybrid semi-supervised approach based on image embeddings and Principal Component Analysis (PCA).

After model deployment, it is often necessary to monitor both the distribution of the input data as well as the inference performance. A common issue encountered in many practical applications has been described as 'data drift', or differences emerging between the distribution of the training data and the data utilized for inference. When TNC conducts surveys in new forests and/or new islands, it can be expected that there will be differences in the season, time of day, cloud cover, presence of coastline, rivers or lakes, and the density and image texture of the forest canopy. We monitored inference performance on new images, detecting the emergence of false positives in some instances. We propose an automated monitoring and mitigation approach based on examining the Haralick texture characteristics [10] of the inference images.

The paper is organized as follows. In Section 2 we present the data processing and analysis tools utilized and their rationale. In Section 3 we present and discuss the results, and in Section 4 we summarize the conclusions.

2. Data processing and analysis

Aerial footage is acquired by TNC contractors (Resource Mapping Hawaii) by flying fixed winged aircraft above affected areas within the Hawaiian Islands. Heavy and persistent cloud cover prevents use of satellite imagery. The data available to TNC and AWS consists of raw images and metadata allowing the geographical localization of the inferred tree ferns.

2.1. Image and geographical coordinates

Images are available in the JP2 format in the range of 100K by 100K pixels. In addition to the raw footage, the metadata associated with each JP2 also includes meta-information that allows each pixel to be associated to specific Universal Transverse Mercator (UTM) geospatial coordinates. The UTM coordinate system divides the world into north-south zones, each 6 degrees of longitude wide. The first UTM coordinate (northing) refers to the distance between a geographical position and the equator, measured with the north as the positive direction. The second coordinated (easting) measures the distance, in meters, towards east, starting from a central meridian that is uniquely assigned for each zone. By convention the central meridian in each region has a value of 500,000, and a meter east of the region central meridian has therefore the value of 500,001. To convert between pixel coordinates and UTM coordinates we utilize the affine transform as outlined in Equation 1 below, where x', y' are UTM coordinates and x, y are pixel coordinates. The parameters a, b, c, d, e, f of the affine transform are provided as part of the JP2 file metadata. We implemented these image operations utilizing the *affine* and *rasterio* Python packages [18]

$$\begin{bmatrix} x\prime \\ y\prime \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

For the purposes of labeling, training and inference the raw JP2 files are divided into non-overlapping 512 by 512 pixels jpg files. The extraction of smaller subimages from the original JP2 necessitates the creation of an individual affine transform directly from each individual extracted jpg file. These operations were performed utilizing the rasterio and affine Python packages and facilitated the reporting of the position of inferred tree ferns in UTM coordinates.

2.2. Data labeling

Visual identification of tree ferns in the aerial images is complicated by several factors. Most of the information is aggregated in the green channel and there is a high density of foliage with frequent partial occlusion of tree ferns by both nearby tree ferns and other vegetation, as illustrated in Figure 1. The information of interest to TNC is the relative density of tree ferns per acre and therefore it is important to count each individual tree fern even in the presence of occlusion. Given these goals and constraints, we chose to utilize an object detection framework.

To label the data, we set up an Amazon SageMaker GroundTruth [16] job. Each bounding box was intended to be centered in the center of the tree fern, and to cover most of the tree fern branches, while at the same time attempting to minimize the inclusion of other vegetation. The labeling was performed by the authors following consultation with TNC domain experts; an example is provided in Figure 1. The initial labeled dataset included 500 images, each typically containing several tree ferns. In this initial labeled set we did not distinguish between native and invasive tree ferns.

2.3. Model training

The labeled data was utilized to train an object detection model by leveraging the Amazon SageMaker Object Detection algorithm [16]. We utilized the Single Shot Multibox Detector (SSD) framework and base network 'vgg-16' [17]. In SageMaker this comes pre-trained on millions of images and thousands of classes from the ImageNet dataset [3]. After training the first iteration of the model, we utilized the predictions from this model to expand the training set. All predicted tree ferns with confidence greater than 0.3 were aggregated into a new set of 3888 images, of which 3142 (80%) were used for training and the rest for validation. For both the first and second iteration of the model the algorithm reports mean absolute precision (mAP) [20]. This measure is dependent on the overlap threshold and is related to the area under the precision-recall curve.

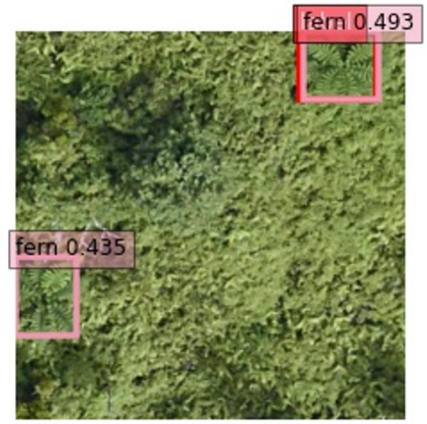

Figure 1. Example of image with labeled (red) and predicted (pink) tree ferns. Australian tree ferns have a characteristic radial shape when viewed from above.

2.4. Model performance evaluation

The utilization of predictions from a first version of the model as training data for a second version model greatly expands the size of the training data without the need for additional manual labeling. However, performance metrics are difficult to interpret. It is conceivable that if the first model produces numerous false positives, which are then utilized as training data by the second model, spurious results could be generated even though the mAP performance metric appears high. To preclude this possibility, we set up a second SageMaker GroundTruth job, which examined the prediction results of 300 images. Here we examined the images with the predicted tree fern bounding boxes overlayed as in Figure 1, and we asked the labeler to decide whether the number of tree ferns detected was correct, an undercount or an overcount. This procedure was intended as an independent and unbiased check on the final model prediction.

2.5. Classification of tree fern species

The object detection algorithm aims to identify all tree ferns within an image, regardless of whether they are natives or invasive. While the invasive tree ferns can have a stereotypical appearance, there are several native tree fern species with subtle differences in appearance. Distinguishing between the tree fern species and obtaining a labeled set of moderate size was considered difficult and time-consuming by TNC.

To determine whether it is possible to distinguish between ATF and native tree ferns without the substantial effort of labeling a large set of images, we implemented an unsupervised image analysis procedure. For each predicted tree fern, we extracted the region inside the binding box and saved it as a separate image. Next, these images were embedded in a high dimensional vector space, generating a

2048 long vector for each input image. These vectors were analyzed by utilizing Principal Component Analysis (PCA). We retained for further analysis the top three components, which together accounted for more than 85% of the variance in the vector data.

For each of the top three components, we extracted the associated images with the highest and lowest scores along the component. These images were visually inspected by the authors and TNC domain experts, with the goal of identifying whether the highest/lowest scores are associated with native/invasive tree ferns. We further quantified the classification power of each principal component by manually labeling a small set of 100 tree fern images as either invasive or native and computing metrics such as area under the precision-recall curve for each of the top three PCA components.

2.6. Creation of inference pipeline

The overall goal of the TNC-AWS partnership is the creation of an automated pipeline that takes as input the JP2 files and produces as output UTM coordinates of the predicted tree ferns. There are three main tasks. The first is the ingestion of the large JP2 file and its division into smaller 512 by 512 jpg files, each of these with an associated affine transform that can generate UTM coordinates from the pixel coordinates. The second task is the actual inference and detection of potential tree ferns and their locations. The final task assembles the inference results into a single csv file which is delivered to TNC. An outline of the pipeline is presented in Figure 2.

Upon a user uploading a JP2 image to a specified S3 location, an automated triggering is activated which starts the AWS StepFunctions [7] workflow. As the JP2 image can potentially be very large (100k by 100k pixels) and its conversion can be time consuming; therefore, it was decided that AWS LambdaFunctions [7] time and storage limitations preclude its usage for image conversion and instead this functionality is performed by StepFunctions.

For the inference procedure we utilized SageMaker BatchTransform [16], which allows the allocation of computing resources on an as-needed basis and importantly reduces costs because EC2 instances are only utilized and charges incurred only during the inference process.

The report generation Lambda function is triggered upon completion of the BatchTransform job. For each jpg image the output consists of a json file which preserves the original filename – for example img_a_b_c_d_e_f.jpg.json.out. The affine transform associate with each jpg inference is constructed by parsing the filename, and tree fern prediction bounding box UTM coordinates are compiled into a csv report. Upon completion SNS notifications are sent to stakeholders. The stakeholders can upload the csv report into their Geographic Information System and visualize the predicted tree fern locations and densities on their map tiles.

The orchestration of the pipeline was implemented using StepFunctions. As is the case for the inference, this choice automates many of the aspects of provisioning and releasing computing resources on an as-needed basis. Additionally, the pipeline architecture can be visually inspected which enhances dissemination to the customer. Finally, as updated models potentially become available in the future, they can be swapped in with little or no disruption to the workflow.

3. Results and discussion

3.1. Model training and validation metrics

The first model training iteration utilized a set of 500 labeled images, of which 400 were in the train set and 100 in the validation set. This model only achieved a mAP score of 0.49. We next used this initial model to produce predictions on a larger set of 3888 jpg images extracted from the available JP2 data. Using this larger image set for training achieved a mAP score of 0.87. This marked improvement illustrates the value of automated labeling and model iteration.

A set of 300 images were randomly selected for an additional round of validation. As outlined in Section 2, we set up a GroundTruth image classification labeling workflow. The labeler examined the images and their predicted tree ferns and decided whether the algorithm correctly counted the number of tree ferns, or whether there were over/undercounts. This procedure revealed that when utilizing a threshold of 0.3 for detection confidence, 84% of the images were deemed by the labeler to have the correct number of predicted tree ferns, with 6.3% being overcounts and 9.7% being undercounts. In most cases , the over/undercounting was off by only one or two tree ferns out of 5-6 present in an image and is not expected to significantly affect the overall estimation of tree fern density per acre.

These results complement and confirm that the relatively high mAP score of 0.87 achieved by the second model iteration is not a spurious or biased result. Based on these metrics, taken together with the positive feedback received from TNC regarding model performance, we conclude that that CV algorithm can quantify tree fern density from aerial imagery without major errors or biases.

3.2. Tree fern classification

To distinguish between invasive and native tree fern species, we implemented the unsupervised procedure outlined in Section 2.5. This procedure was tested against a manually labeled set of 100 images. When the PC scores were used as inputs to a binary classifier, we found that PC2 was the most discriminative, followed by PC3, with

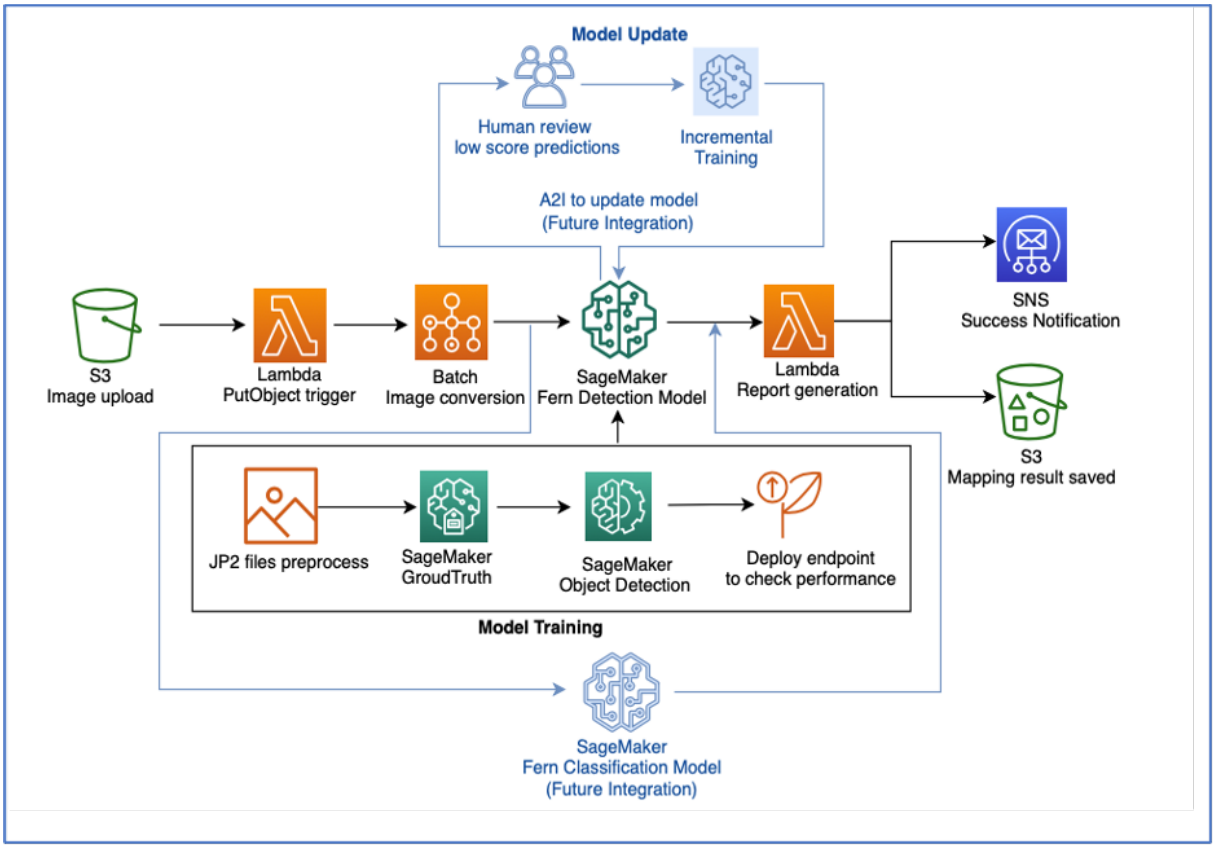

Figure 2. High level architecture of the labeling and training workflow. Bottom: refinement of model utilizing Amazon SageMaker Ground Truth. Middle: inference pipeline upon object upload, processing within the AWS cloud, and success notification.

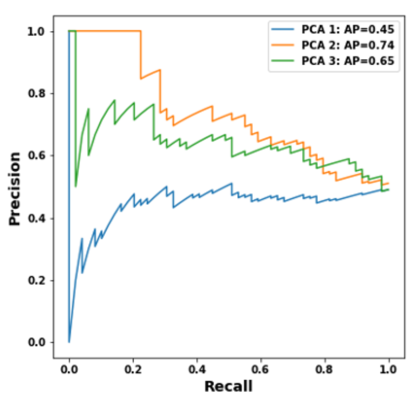

Figure 3. Performance of top 3 PCs as binary classifier inputs. AP: average precision.

PC1 displaying only modest performance in distinguishing between native and invasive tree ferns (Figure 3).

3.3. Inference performance

When the inference pipeline was used in batch mode on a source image of 10K by 10K pixels, and allocating an m4.large instance to the SageMaker BatchTransform, the whole inference workflow executed within 25 minutes. Of these, 10 minutes was taken by the BatchTransform and the rest by Step and Lambda functions execution time. TNC expects sets of up to 24 JP2 images at one time, about twice a year. By adjusting the size and number of instances to be used by the BatchTransform, the inference pipeline can complete the inference in a time that we estimate to be between 30 minutes to 2 hours.

3.4. Evaluation on novel images

The model was built utilizing training data which was collected in 2017 from the Wainiha Valley on the island of Kaua'i. TNC subsequently utilized the algorithm on images collected at later times from additional locations within the Hawaiian archipelago. In the course of evaluating the inference results, we detected spatially clustered patterns of false positives, which upon closer inspection turned out to be originating from very steep slopes and from riverbanks. Visual examination revealed that the images con-

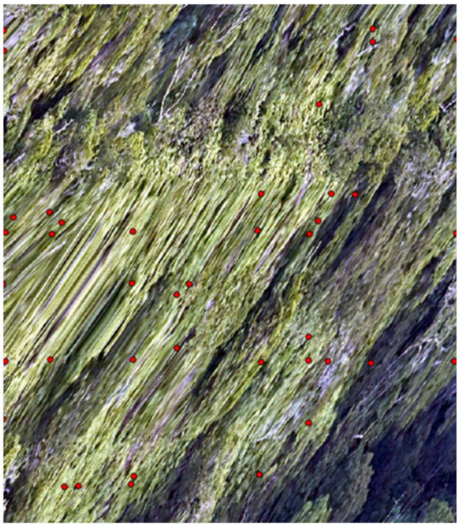

Figure 4. Example of streaked and blurred image collected from steep mountain slope, together with false positive tree ferns (red dots).

tained streaking and blurring patterns (Figure 4) for which, at moderate thresholds (0.3), the object detection algorithm incorrectly identified the presence of tree ferns. To evaluate whether the inference pipeline could automatically detect blurred/streaked images without the need for manual inspection, we selected 100 images from the blurred/streaked regions and a similar number from the original training dataset. For both image sets we computed the 13 Haralick texture features [10]. These features are based on second-order statistics calculated from the Gray Level Cooccurrence Matrix (GLCM). First, the image is converted to grayscale, and subsequently the gray values are digitized into N distinct values. Computing the Haralick features consists of counting the co-occurences of gray levels in neighboring pixels and results in an *N by N* square matrix.

While in principle is possible to associate individual Haralick features with specific types of image alterations such as streaking, at present we do not pursue this and we utilize the features simply as means to detect images that are outliers in reference to the training set. To compute the features we utilized the *mahotas* Python package [2]; each feature value was averaged over the 4 directions. The results of the comparison are displayed in Figure 5.

4. Conclusions and future directions

4.1. Benefits to TNC

The major benefits to TNC from adopting the inference pipeline proposed here are twofold. First, there is the possibility to almost completely eliminate manual labeling efforts because the performance of the model is comparable or better to that of the human labelers, as determined by TNC. However, we do expect a small amount of manual effort in the future in case we observe model drift and the need for additional labeling and model retraining. Second, there are substantial time savings. Currently manual labeling for a dozen large JP2 files take several weeks to complete, whereas the inference pipeline is expected to take in the order of hours, depending on the number and size of inference instances allocated. A faster turnaround time would impact the capacity of TNC to plan routes for the crews responsible for treating the invasive tree ferns in a timely manner, and potentially find appropriate treatment windows considering the seasonality and weather patterns on the islands.

4.2. Potential model enhancements

At the present time the pipeline includes an object detection model that does not distinguish between natives and invasive tree ferns. This is also the case for the manual labeling efforts. Whether based on human labeling or model inference, TNC uses the predicted position of the tree ferns to direct resources based on tree fern density per acre. Upon arrival, ground crews determine whether the tree fern clusters are invasive and require treatment. The results from the PCA procedure are very encouraging and we have plans for further testing to quantify reliability. In the near future TNC still intends to direct ground crews to examine and potentially treat all regions with high tree fern density.

There are two possibilities for integrating the tree fern classification into the inference workflow. The first is to train a linear classifier based on the values of the PC1 and PC2, which as we demonstrated in Section 3 have predicted value. In this case, the architecture of the workflow would be altered as follows. After the SageMaker BatchTransform completes, bounding boxes for predicted tree ferns are extracted and embedded into a vector space [9], followed by a PCA SageMakerBatchTransform. The resulting PC1 and PC2 values are in turn fed into a LinearClassifier BatchTransform, resulting in a probability, for each predicted tree fern, of being native or invasive. The main advantage of this approach is that the requirements for manual labeling are low.

The second approach would require more labeling effort, but it would have the benefit of leaving the current inference pipeline in place. The object detection model would simply be expanded to include two classes, for native and invasive tree ferns respectively. Being based on a deep neural net architecture, this solution likely requires more labeled data. To acquire this data, TNC could leverage services such as AmazonMechanicalTurk [15]. The second possibility is to create a labeling pipeline utilizing AugmentedAI [8]. We are currently evaluating these options.

A second enhancement to the model regards increasing the robustness in the face of likely changes to the image

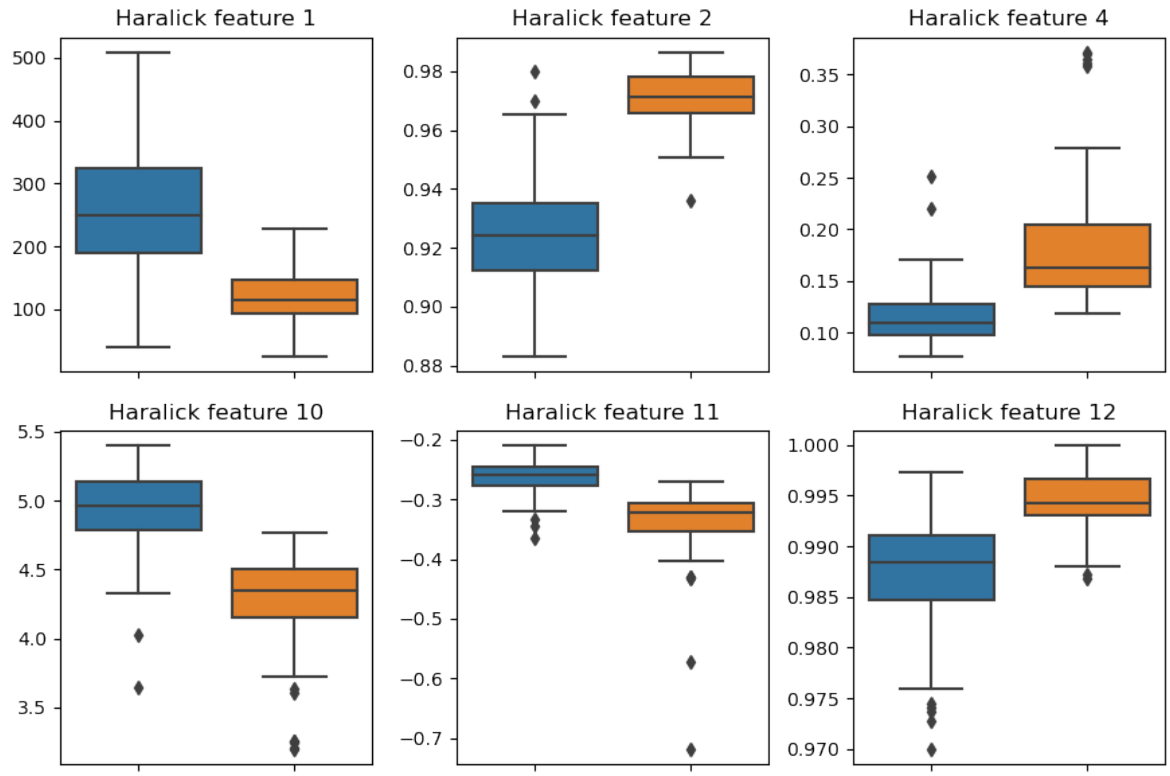

Figure 5. Comparison of Haralick features computed from images from the training set (left) compared to features computed from the streaked/blurred images in the inference set (right).

characteristics in future aerial data collection. The streaking and blurring observed on steep slopes is just one example of surprising changes to imagery collected; additional potential alteration could be generated from changes in the altitude of the aircraft, or changes in season or time of day of image collection that affect the coloring and saturation of the forest canopy images. To prevent performance decreases stemming from these issues, we are considering three complementary strategies.

The first and most readily available pipeline alteration is simply to examine the values of the Haralick textural feature for each new inference image and flag as outliers the images sufficiently distinct from the training set. Subsequently, the end user can examine the geographical distribution of flagged images and visually examine the imagery collected from the affected areas. The second strategy is to utilize image augmentation and to artifically add alterations to the training set, such as changes in luminance, changes in viewing angle and the introduction of some blurring. Next, the model could be retrained using the augmented dataset and performance re-evaluated. Third and most laborious process would involve additional labeling and subsequent model retraining.

One potential strategy to reduce the labeling load is to focus the effort in specific regions rich in 'novel' images as detected by examining the Haralick features through an outlier detection algorithm. As illustrated in Figure 5, some textural features are highly distinct in the novel versus training dataset images. In this scenario, the end user would pay particular attention to the emergence of spatially clustered patterns of 'novel' images. If the potential tree fern detections flagged as questionable because they originate from novel images are relatively uniformly distributed over the entire area surveyed, then this is likely a typical false-positive issue which might require a simple revision upward of the detection threshold. If, on the other hand, the spatial patterns of questionable predictions are clustered as specific locations, then a more careful visual inspection of the affected images is warranted. In these areas the false positives will be corrected and the false negatives missed by the algorithm correctly labeled. We expect that this procedure would greatly reduce the load of labeling new imagery data, as it would not be necessary to manually inspect *de novo* the entire area newly surveyed. Instead, the operator can just focus on the much smaller subset of images that are both atypical of previous surveys and have candidate tree ferns detected.

4.3. Conclusions

The partnership between AWS and TNC has resulted in a working pipeline that automates the process of tree fern detection and adds efficiency to the process of eliminating invasive tree ferns from the Hawaiian Islands. Our pipeline has features that could be of interest to other conservation efforts or more generally to agricultural or landscape survey data. Importantly, we demonstrate that ML technologies not only automate and scale the process of inference, but also allow the creation of high-quality training datasets with minimal human labeling effort that is coupled with an iterative ML based data annotation pipeline.

References

[1] John A. Cigliano and Heidi L. Ballard. *Citizen Science for Coastal and Marine Conservation*. Routledge, Oct. 2017. Google-Books-ID: xjQ8DwAAQBAJ.

[2] Luis Pedro Coelho. Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software*, 1(1):e3, July 2013. Number: 1 Publisher: Ubiquity Press.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. ISSN: 1063-6919.

[4] Michael C. Dietze, Andrew Fox, Lindsay M. Beck-Johnson, Julio L. Betancourt, Mevin B. Hooten, Catherine S. Jarnevich, Timothy H. Keitt, Melissa A. Kenney, Christine M. Laney, Laurel G. Larsen, Henry W. Loescher, Claire K. Lunch, Bryan C. Pijanowski, James T. Randerson, Emily K. Read, Andrew T. Tredennick, Rodrigo Vargas, Kathleen C. Weathers, and Ethan P. White. Iterative near-term ecological forecasting: Needs, opportunities, and challenges. *Proceedings of the National Academy of Sciences*, 115(7):1424–1432, Feb. 2018. Publisher: Proceedings of the National Academy of Sciences.

[5] Scott S Farley, Andria Dawson, Simon J Goring, and John W Williams. Situating Ecology as a Big-Data Science: Current Advances, Challenges, and Solutions. *BioScience*, 68(8):563–576, Aug. 2018.

[6] Lisa Feldkamp. Citizen Science Tuesday: Hawaii Challenge, June 2014.

[7] H. L. Fourie. *AWS Lambda and Step Functions*. CreateSpace Independent Publishing Platform, May 2017. Google-Books-ID: 7pNntAEACAAJ.

[8] Chris Fregly and Antje Barth. *Data Science on AWS*. "O'Reilly Media, Inc.", Apr. 2021. Google-Books-ID: SdgnEAAAQBAJ.

[9] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. "O'Reilly Media, Inc.", Sept. 2019. Google-Books-ID: HnetDwAAQBAJ.

[10] Robert M. Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973. Publisher: Ieee.

[11] Natalie Kurashima, Jason Jeremiah, and and Tamara Ticktin. I Ka Wā Ma Mua: The Value of a Historical Ecology Approach to Ecological Restoration in Hawai'i 1. *Pacific Science*, 71(4):437–456, Oct. 2017. Publisher: University of Hawai'i Press.

[12] Jessica L. Morgan, Sarah E. Gergel, and Nicholas C. Coops. Aerial Photography: A Rapidly Evolving Tool for Ecological Management. *BioScience*, 60(1):47–59, Jan. 2010. _eprint: https://academic.oup.com/bioscience/article-pdf/60/1/47/19404239/60-1-47.pdf.

[13] Sireesha Muppala, Randy DeFauw, and Shelbee Eigenbrode. *Amazon SageMaker Best Practices: Proven tips and tricks to build successful machine learning solutions on Amazon SageMaker*. Packt Publishing Ltd, Sept. 2021.

[14] Werner Rammer and Rupert Seidl. Harnessing Deep Learning in Ecology: An Example Predicting Bark Beetle Outbreaks. *Frontiers in Plant Science*, 10, 2019.

[15] Kim Sheehan and Matthew Pittman. *Amazon's Mechanical Turk for Academics: The HIT Handbook for Social Science Research*. Melvin & Leigh, Publishers, 2016. Google-Books-ID: N9DCjwEACAAJ.

[16] Julien Simon. *Learn Amazon SageMaker: A guide to building, training, and deploying machine learning models for developers and data scientists*. Packt Publishing Ltd, Nov. 2021. Google-Books-ID: D2pHEAAAQBAJ.

[17] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, Apr. 2015. arXiv:1409.1556 [cs].

[18] Silas Toms, Paul Crickard, and Eric van Rees. *Mastering Geospatial Analysis with Python: Explore GIS processing and learn to work with GeoDjango, CARTOframes and MapboxGL-Jupyter*. Packt Publishing Ltd, Apr. 2018. Google-Books-ID: ivBZDwAAQBAJ.

[19] Marie Valencia. Rate of Environmental Damage Increasing Across the Planet but There Is Still Time to Reverse Worst Impacts if Governments Act Now, UNEP Assessment Says, May 2016.

[20] Gemine Vivone, Paolo Addesso, and Amanda Ziemann. *Remote Sensing for Target Object Detection and Identification*. MDPI, Mar. 2020. Google-Books-ID: YpfUDwAAQBAJ.

[21] Adam E. Vorsino, Lucas B. Fortini, Fred A. Amidon, Stephen E. Miller, James D. Jacobi, Jonathan P. Price, Sam 'Ohukani'ohi'a Gon Iii, and Gregory A. Koob. Modeling Hawaiian Ecosystem Degradation due to Invasive Plants under Current and Future Climates. *PLOS ONE*, 9(5):e95427, May 2014. Publisher: Public Library of Science.