OPTIMIZING MARKING TECHNIQUES FOR MARK-RECAPTURE STUDIES OF MOUNTAIN PINE BEETLES

J. BENESH[1], D. GOODMAN[2], R. HAN[3], J. HOEPNER[4], H. HUANG[5], and M. RAY[6]

ABSTRACT: Whereas mark-recapture studies are sometimes used to estimate population size, mark-recapture studies initiated by Natural Resources Canada attempt to estimate movement of individuals in the population by marking them at source locations and recapturing them at various surrounding trap sites. A novel variation of traditional mark-recapture techniques developed by Natural Resources Canada involves coating trees with paper that fluoresces under black light such that the beetles are marked with paper dust as they emerge. Recaptured beetles are then photographed under black light. In this work, we classify images of the recaptured beetles as marked or unmarked using deep neural networks. In particular, we use transfer learning where existing top-performing classifiers are applied to our beetle image classification problem. We compare the performance of ResNet50 and EfficentNet base models by varying certain parameters, and finally obtain the most optimal model to classify images.

KEYWORDS: image classification, machine learning

[1] University of Lethbridge, Alberta, Canada; joel.benesh@uleth.ca
[2] Northern Forestry Centre, Natural Resources Canada, Canada; devin.goodsman@canada.ca
[3] University of British Columbia, British Columbia, Canada; 13hanr@gmail.com
[4] University of Victoria, British Columbia, Canada; julesihoepner@gmail.com
[5] University of Calgary, Alberta, Canada; hui.huang1@ucalgary.ca
[6] University of Calgary, Alberta, Canada; mishty.ray@ucalgary.ca

1. INTRODUCTION

Since 1990, an outbreak of the mountain pine beetle (*Dendroctonus ponderosae*) has affected over 20 million hectares of forest in western Canada, making it the largest recorded insect outbreak in North American history. Mountain pine beetle adults disperse to attack and colonize trees to lay eggs beneath the outer bark. The process of attack and colonization disrupts nutrient flow and results in tree death **(Dhar, Parrott and Hawkins 2016)**. Although understanding beetle dispersal in this context is vital in making well-informed environmental decisions, tracking beetle relocation when adults disperse is challenging. Natural Resources Canada (NRCan) recently initiated a study designed to develop new and improved methods to quantify mountain pine beetle dispersal and to understand how far dispersing mountain pine beetles fly.

Mark-recapture studies typically involve the application of a harmless indicator to a small number of individuals, which are then released back into the general population. The likelihood of recapturing a marked individual is thus inversely proportional to the size of the population, assuming nearly all of the marked individuals are still alive, provided no significant immigration in or out of the population has occurred between the release and recapture dates. The goal of mark recapture studies initiated by NRCan is slightly different: Marked beetles are recaptured at various locations from release sites to better understand the dispersal process **(Safranyik, et al. 1992)**.

A recently developed NRCAN marking technique involves covering trees in paper that fluoresces under black light such that the beetles are coated in paper dust as they emerge, thereby allowing the marked beetles to naturally disperse without direct human intervention. Mountain pine beetles emerging from papered trees and control trees were later captured and photographed under black light.

Manually classifying each image as marked or unmarked can be tedious and prone to error, hence it would be beneficial to automate the process using machine learning. The goal of this project is to optimize pre-existing image classification algorithms to identify marked beetles. We chose algorithms based on ResNet50 and EfficientNet as they are some of the top performing image classification techniques available. The optimization is done in two phases. In phase I, we compare ResNet50 and EfficientNet by training those models on our original dataset and the "threshcropped" dataset (we explain how we obtain this new dataset in Section 3.3. The comparisons are made on the basis on F1 score and time taken for the model to train. We do a second phase of finetuning on the best model chosen from phase I by varying certain parameters in our algorithm. After phase II, we obtain our "best model". These techniques are discussed in further detail in Section 2, and their implementation is presented in Section 3. In Section 4, the results are summarized and potential improvements to the algorithm are addressed.

2. DEEP NEURAL NETWORKS

Deep learning has been well justified by its tremendous empirical success and state-of-the-art performance on various real-life applications such as speech recognition **(Hannun, et al. 2014)**, image recognition **(He, et al. 2016)**, language translation **(Vaswani, et al. 2017)**, and as a novel method for scientific computing **(Berner, Grohs and Jentzen 2020)**. It is an approach that enables the realization of complex tasks such as the ones mentioned above, by means of highly parameterized functions, called deep artificial neural networks $\mathcal{N}: \boldsymbol{R}^{d_0} \to \boldsymbol{R}^{d_L}$. A classical architecture is the one of feed-forward artificial neural networks of the type

$$\mathcal{N}(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{W}_L^\top \boldsymbol{\sigma}(\boldsymbol{W}_{L-1}^\top \dots \boldsymbol{\sigma}(\boldsymbol{W}_1^\top \boldsymbol{x} + \boldsymbol{b}_1) \dots) + \boldsymbol{b}_L),$$

where $\boldsymbol{L}$ is depth of the network, the function $\boldsymbol{\sigma}$ is a scalar activation function acting component-wise on vectors, for each layer $\mathbf{l} = \mathbf{1}, \dots, \mathbf{L}$, the matrix $\boldsymbol{W}_l \in \boldsymbol{R}^{d_{l-1} \times d_l}$ represents a collection of weights, and the vector $\boldsymbol{b}_l$ represents shifts/biases. The neural network $\mathcal{N}$ is then trained to minimize a given loss function (e.g, Mean Squared Error, Cross-Entropy, Kullback-Leibler divergence, or Wasserstein distances) over the parameters (weights and biases) of the network, usually measuring the misfit of input-output information over a given finite number of labeled training samples.

In this paper, we use Convolutional Neural Networks (CNNs) to solve our image classification problem. However, to train on a very large dataset, deep CNN models may take a significant amount of time. A way to bypass this process is to re-use the model parameters from pre-trained top performing CNN models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. This is the so-called transfer learning. There are many top-performing models that are available for the basis for image recognition tasks, such as VGG (e.g. VGG19 **(Simonyan and Zisserman 2014)**), GoogLeNet (e.g. InceptionV3 **(Szegedy, et al. 2016)**), Residual Network (e.g. ResNet50 **(He, et al. 2016)**) and EfficientNet (e.g. EfficientNetB0 **(Tan and Le 2019)**). In the following we are going to focus on the implementations on the Residual Network and the EfficientNet models, which are the state-of-the-art methods in imagine classification.

2.1 RESNET50

ResNet50 is one of the most powerful and award winning deep neural networks **(He, et al. 2016)**. It was proposed to solve the issue of vanishing/exploding gradient phenomenon \citehere?. The idea is to use the 'Residual Block' to skip connections and after-addition activations. If we denote by $\boldsymbol{\mathcal{F}}(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b})$ a generic layer of the network, then the residual layer can be described as $\boldsymbol{x}^{\boldsymbol{n}+\mathbf{1}} = \boldsymbol{x}^{\boldsymbol{n}} + \boldsymbol{\mathcal{F}}(\boldsymbol{x}^{\boldsymbol{n}})$.

2.2 EFFICIENTNET

EfficientNet was first introduced in **(Tan and Le 2019)**, since then it has become one of the most efficient models that reaches state-of-the-art accuracy on both ImageNet and common image classification transfer learning tasks. It proposes a compound scaling method to scale up CNNs to obtain better accuracy and efficiency. Unlike conventional approaches that arbitrarily scale network dimensions, such as width, depth, and resolution, EfficientNet uniformly scales each dimension with a fixed set of scaling coefficients. More specifically, it uses a compound coefficient $\boldsymbol{\varphi}$ to uniformly scales network width, depth, and resolution in a principled way

depth: $\boldsymbol{d} = \boldsymbol{\alpha}^{\boldsymbol{\Phi}}$,

width: $\boldsymbol{w} = \boldsymbol{\beta}^{\boldsymbol{\Phi}}$, and

resolution: $\boldsymbol{r} = \boldsymbol{\gamma}^{\boldsymbol{\Phi}}$

such that $\boldsymbol{\alpha} \cdot \boldsymbol{\beta}^{\mathbf{2}} \cdot \boldsymbol{\gamma}^{\mathbf{2}} \approx \mathbf{2}$; with $\boldsymbol{\alpha} \geq \mathbf{1}$, $\boldsymbol{\beta} \geq \mathbf{1}$ and $\boldsymbol{\gamma} \geq \mathbf{1}$ to be determined by a grid search. For our image classification problem, depending on the choice of the resolution of the input image, we can use a series of EfficientNet models from B0 to B7.

2.3 METRICS FOR COMPARISON

We quantify model performance based on $\boldsymbol{F_1}$ scores and accuracy. We also record the time taken to train our models during phase I of comparisons. We explain our metrics in detail below.

We split the original dataset, comprising of beetle images, into training and validation sets. We wish to train our model to perform a binary classification – classify beetle images as marked or unmarked. Once the model is trained on the training set, we evaluate its performance by using it to classify the validation set. From the results, we obtain a confusion matrix.

\insertimagehere

Here TP denotes true positives, TF denotes true negatives, FP denotes false positives, and FN denotes false negatives. Using these values, we may calculate the following:

1. *Precision* is the ratio of true positives with respect to all instances marked positive by the classifier:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

2. *Recall* is the ratio of the true positives with respect to all instances which are actual positives:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

3. $\boldsymbol{F_1}$ *score* is the harmonic mean of precision and recall:

$$\boldsymbol{F_1\,score} = \boldsymbol{2} \cdot \frac{\boldsymbol{precision \cdot recall}}{\boldsymbol{precision + recall}}$$

4. *Accuracy* is the ratio of correctly identified instances with respect to all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3. DATASET EXPLORATION

Our dataset consisted of 1057 images primarily of beetles (some other insects were also captured due to natural interference; they were photographed and recorded nonetheless) in .tif file format. Here is a sample of the filenames:

PaperedTransparent22v.tif
PaperedMixed24d.tif
PaperedControl4d.tif
NoPaperedGreen76v.tif
PinkPapierMache1d.tif
PinkPaintedPaper1v_light.tif
Trap89072019540pmv.tif

3.1. READING THE DATASET

The filenames of each image classified them as follows.

1. The first component of each name provides information on the source from where the beetles were captured.
   a. **"Papered"** and **"NoPapered"** determines whether the tree segment from where the beetle in the image emerged was papered or not. Papered bolts were considered marked whereas unpapered bolts can be considered unmarked. For example, PaperedMixed24d.tif is considered marked and NoPaperedGreen76v.tif is considered unmarked.
   b. **"Trap"** indicates that the beetles originated from a separate outdoor experiment in which standing trees that were infested with mountain pine beetles were papered. A number of Lindgren funnel traps were set up in the vicinity to capture beetles emerging from trees in the area. Most of the trapped beetles likely emerged from unmarked trees. For these beetles (and other

insects), we do not know whether they emerged from papered trees. For example, Trap89072019540pmv.tif is the filename of a beetle in this category.

c. "PinkPapierMache" or "PinkPaintedPaper" also refer to marked beetles from papered trees.

2. The next component of each name is the color of the paint that was applied to the outside of the trees from which beetles emerged: Possible values of the paint color include: **transparent, green, pink, mixed, or control (no paint)**. For example, PaperedControl4d.tif comes from a tree with no paint, whereas NoPaperedGreen76v.tif comes from a tree with green paint. If this component says "mixed", this means that the beetles were mixed in a jar together to test the persistence of the marking paper in a slightly more realistic context. We note that mixing could potentially lead to cross-contamination of unpapered beetles because of physical contact with marked beetles or with paper fragments that may have been shed from them.
3. The third component is a number that is not unique to each beetle.
4. The final component indicates whether beetles were photographed on their **dorsal (d)** or **ventral (v)** sides. For example, PaperedControl4d.tif was photographed on its dorsal side. We note that the tips of the abdomens and the mandibles on the ventral side are a location of higher concentrations of paper particles in some cases when beetles were marked.
5. Another identifier that is added to some of the images is the label **"light"**. These images were not imaged under blacklight and had a very distinct look to them (see Figure \ref{bvl}) in comparison to the rest of the images.

Figure 1: Part (a) shows a sample image from the original dataset that is included in the training set. Part (b) shows an image with 'light' in the filename which is excluded due to its distinct appearance

3.2. REFINING THE DATASET

To refine the input dataset to train the binary classifier, we only consider images with filenames that start with “Papered” or “NoPapered”. Papered filenames are considered **marked** beetles and NoPapered filenames are considered **unmarked**. We explicitly exclude filenames with “light” in them as their appearance was very different and we expect any new data given to our trained model for classification to be images of beetles photographed under a blacklight.

We do not include filenames with “Trap”. Most of these beetles likely emerged from unmarked trees. After obtaining the “best” model, we use it on the dataset with these filenames to classify them as marked or unmarked. We do not include filenames starting with the “Pink” tag, as they are from papered trees and hence marked. Including them would give us a dataset of 735 marked images and 278 unmarked images, which makes it quite unbalanced and may produce a heavy bias towards predicting marked beetles. \citation about unbalanced datasets in image classification? Upon removing these images, we were left with a total of 479 marked images and 278 unmarked images, which gave us a much better balance to begin training our machine learning algorithms on. We call this dataset of 757 images the **original dataset**.

\insertimage of dataset statistics

3.3. THRESHCROPPED DATASET

We create a new of the original dataset to obtained images that are cropped and grayscale. We create this using thresholding to convert each image into one with a binary colour scheme and then cropping around the beetle. We therefore suggestively refer to this data set as the **Threshcropped dataset.** See Figure \addfigref to compare a sample from each data set. This new dataset has a lower file size and excludes much of the coloured background from each image. Since we have streamlined the input and reduced file size, we wish to look at whether this will improve the time taken to train the model and how the performance

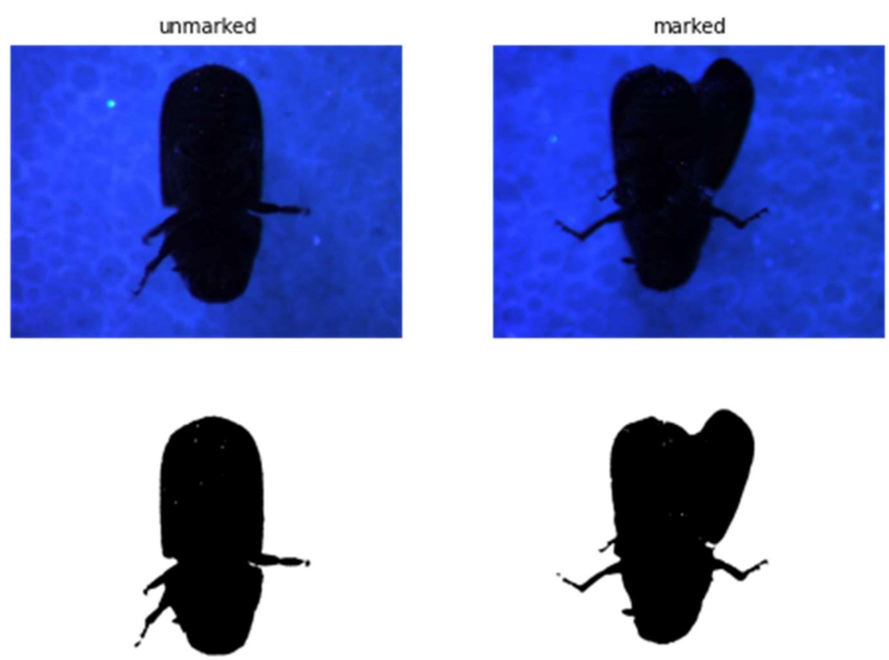

Figure 2: Original vs Threshcropped beetles. The column label indicates whether the beetles are marked or unmarked. The first row consists of original images and the second consists of threshcropped versions of the same images.

metrics will change.

4. IMPLEMENTATION AND RESULTS

We train our models on both the original and threshcropped datasets. We split each dataset into training and validation sets with an 80-20 ratio. We summarize how the data is split up in Figure. Resnet50 and EfficientNet were chosen because they are top performing image classification techniques. Our goal is to optimize them for our setup so that we may choose the best performing model to classify captured beetles into marked and unmarked categories.

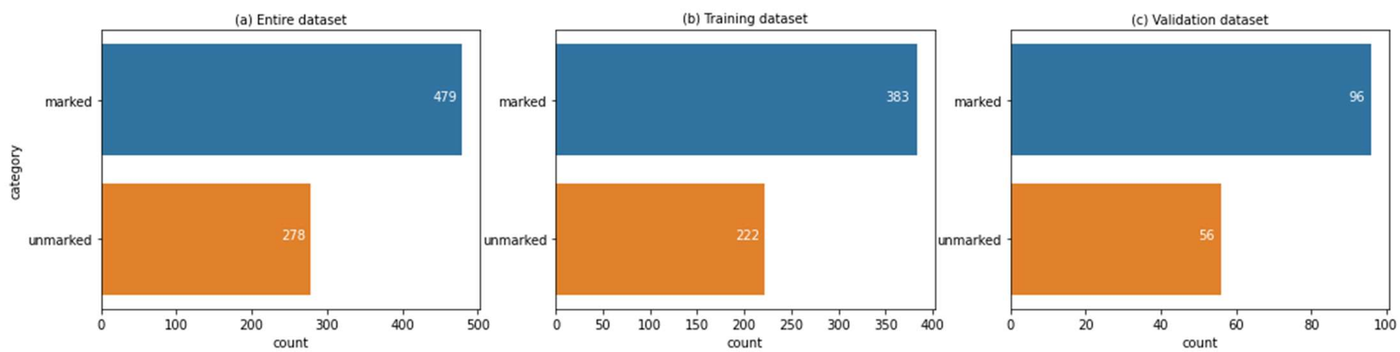

Figure 3: The entire original dataset (757 images) is split into training dataset (605 images) and validation dataset (152 images). The figure shows the comparison of number of marked vs unmarked beetles in each dataset.

4.1 PHASE I

We train Resnet50 and various EfficientNet models on both original and threshcropped datasets and record the resulting time taken, accuracy, and F_1 score from the resulting confusion matrices. During these experiments, we keep all variable parameters and preprocessing functions the same. A member of our group classified these images based on naked eye observations. We obtain a confusion matrix from this classification and calculate the metrics. We call this model the human classifier. The table below summarizes the results of our experiments.

We note that all models perform better with the threshcropped dataset. EfficientNet B0 runs the fastest but is outperformed in other metrics. EfficientNet B3 and B7 are very costly in terms of time taken to run, so we eliminate them from phase II finetuning. ResNet50 takes longer than EfficientNet B0, but otherwise runs on a reasonable timeframe. It produces much better accuracies and F_1 scores.

Table 1: Comparison of metrics of model performance

Model	Original dataset			Threshcropped dataset		
	Time (min)	Accuracy (%)	F_1 score (%)	Time (min)	Accuracy (%)	F_1 score (%)
Human Classifier	NA	76.40	78.28	NA	NA	NA
EfficientNet B0	50	63.15	77.41	41	75.65	82.94
EfficientNet B3	398	63.81	77.73	74	79.60	84.42
EfficientNet B7	628	67.10	77.06	909	76.31	82.17
Resnet50	72	74.34	74.83	103	84.86	87.83

From this table, we conclude that Resnet50 with threshcropped images is the best model for further finetuning.

4.2 PHASE II: FINETUNING RESNET50

We vary two parameters – image size and batch size in the ResNet50 algorithm for both original and threshcropped images. We include original images in the finetuning because we wish to see if we can get better or comparable metrics without needing to generate a threshcropped dataset. We obtain the accuracy and F_1 score calculated on the validation

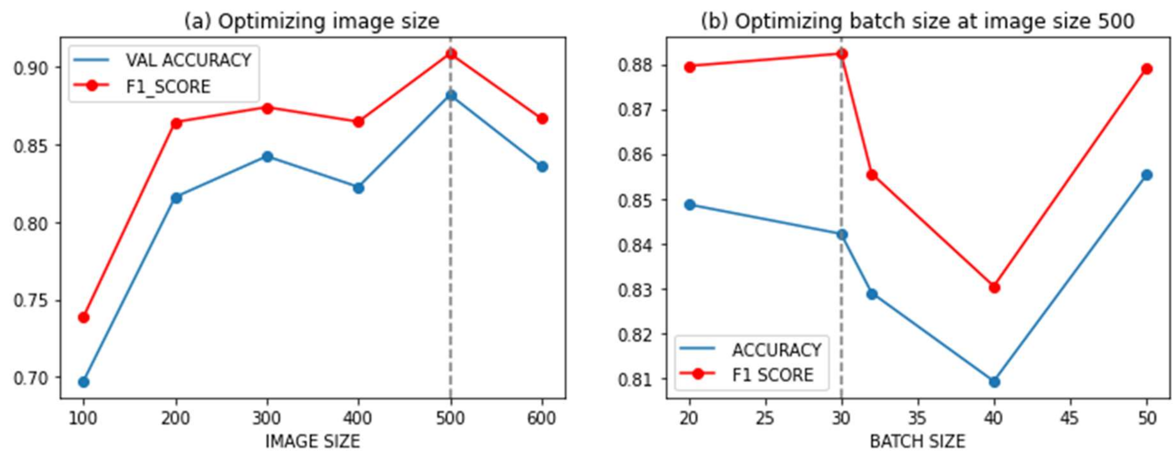

Figure 4: This figure summarizes the optimization workflow for the threshcropped dataset. Part (a) shows the best image size = 500, which is fixed at this value in part (b), which shows best batchsize of 30.

dataset for each iteration. We drop the time comparison because we have fixed ResNet50, and the time taken to run various iterations of ResNet50 is reasonable (and comparable to what was recorded), moreover we now prioritize better prediction over time. We summarize our results in Error! Reference source not found. and **Figure 5.**

We see in **Figure 6** that ResNet50 with original images yields very good metrics and outperforms its threshcropped counterpart, with an accuracy of 97.36% and an F_1 score of 97.93%. This is our best model. We use this best model classify the trap data (see section 3.1). In other words, we use ResNet50 with image size 300 and batch size 32 trained on original images. Out of 77 trap beetles, the classifier tells us that 64 are marked and 13 are unmarked. See **Figure 7** to visualize the results.

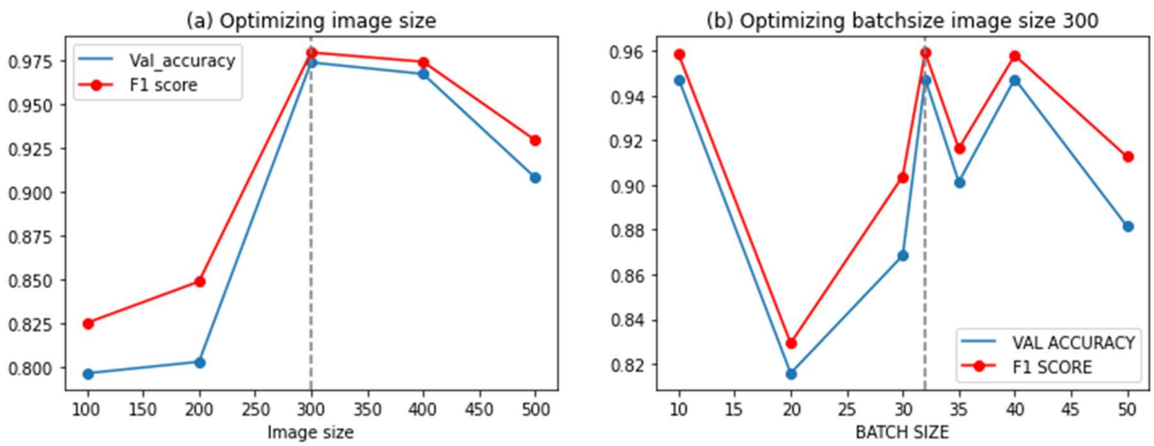

Figure 5: This figure summarizes the optimization workflow for the threshcropped dataset. Part (a) shows the best image size = 300, which is fixed at this value in part (b), which shows best batchsize of 32.

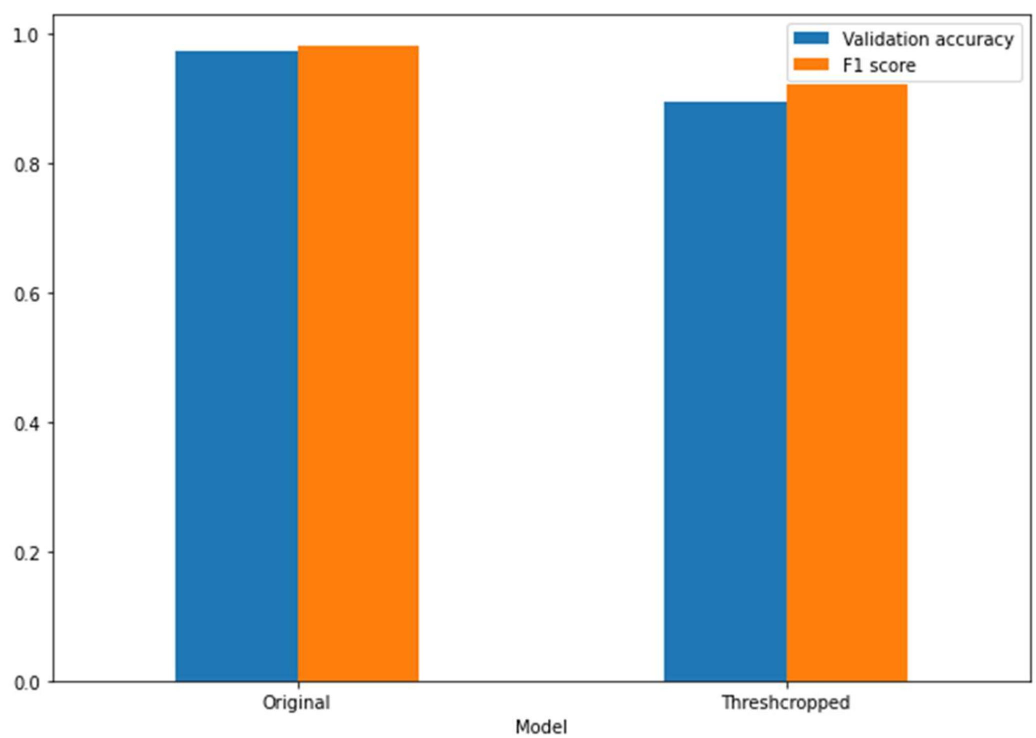

Figure 6: Summary of metrics of the optimized models for original and threshcropped datasets

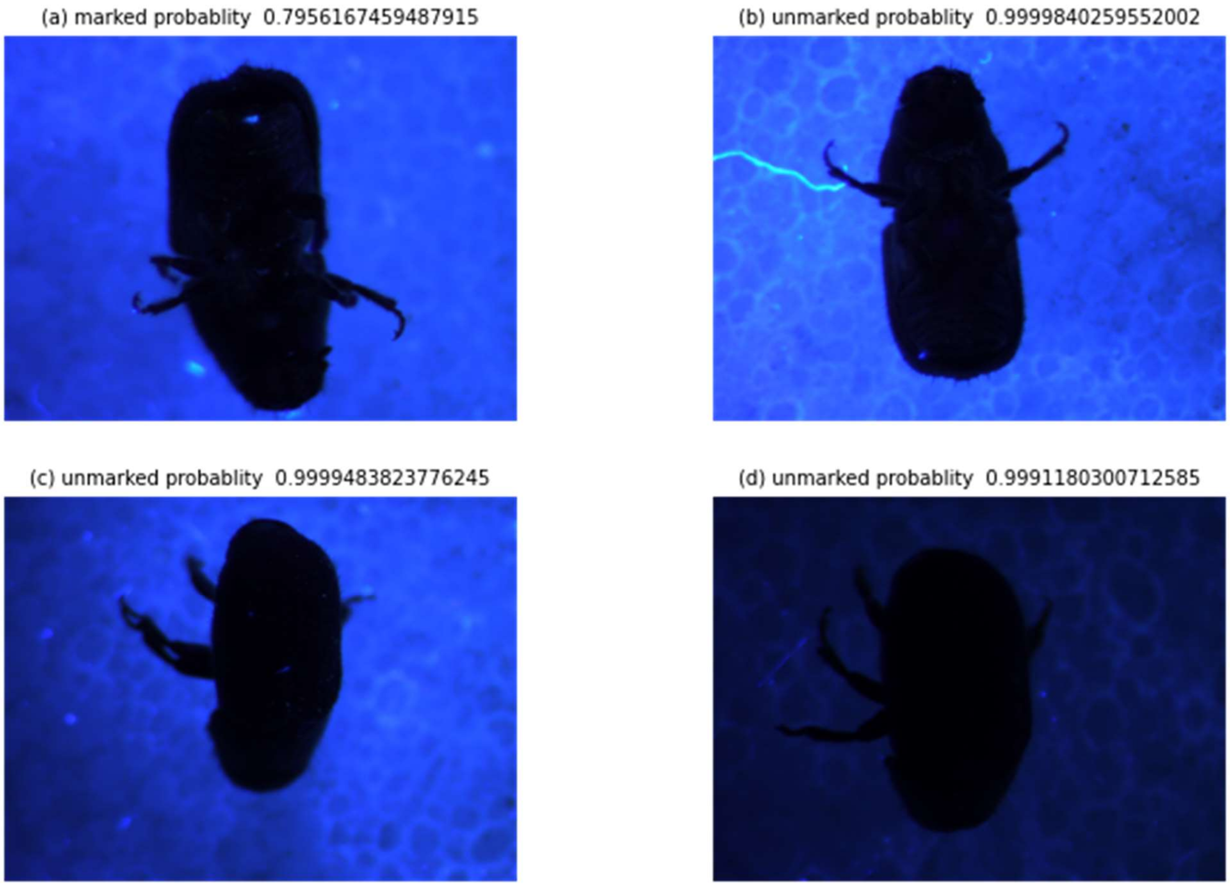

Figure 7: Beetle images from the trap data along with the prediction probabilities

ACKNOWLEDGEMENTS

Mishty Ray would like to acknowledge Natural Resources Canada for contract #3000736333 that aided in the completion of this project. All authors would like to thank the organizers of the Math to Power Industry Workshop during which most of the set up of this project was carried out.

BIBLIOGRAPHY

Berner, Julius, Philipp Grohs, and Arnulf Jentzen. 2020. "Analysis of the Generalization Error: Empirical Risk Minimization over Deep Artificial Neural Networks Overcomes the Curse of Dimensionality in the Numerical Approximation of Black--Scholes Partial Differential Equations." *SIAM Journal on Mathematics of Data Science* (SIAM) 2 (3): 631--657.

Dhar, Amalesh , Lael Parrott, and Chris Hawkins. 2016. "Aftermath of mountain pine beetle outbreak in British Columbia: Stand dynamics, management response and ecosystem resilience." *Forests* (Multidisciplinary Digital Publishing Institute) 7: 171.

Hannun, Awni, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, and Adam Coates. 2014. "Deep speech: Scaling up end-to-end speech recognition."

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770--778.

Safranyik, L, DA Linton, R Silversides, and LH McMullen. 1992. "Dispersal of released mountain pine beetles under the canopy of a mature lodgepole pine stand." *Journal of Applied Entomology* (Wiley Online Library) 113: 441--450.

Safranyik, Lee. 1989. "Mountain pine beetle: biology overview." *Proceedings: Symposium on the Management of Lodgepole Pine to Minimize Losses to the Mountain Pine Beetle. USDA Forest Service, Intermountain Forest and Range Experiment Station, Gen. Tech. Rep. INT-262.* 9--12.

Simonyan, Karen, and Andrew Zisserman. 2014. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556.*

Szegedy, Christian, Vincent Vanhoucke, Sergey Loffe, Jon Shlens, and Zbigniew Wojna. 2016. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2818--2826.

Tan, Mingxing, and Quoc Le. 2019. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International Conference on Machine Learning.* PMLR. 6105--6114.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention is all you need." *Advances in neural information processing systems.* 5998--6008.

SUPPORTING MATERIALS