DEEP CONVOLUTIONAL NEURAL NETWORK FOR AUTOMATIC DETECTION OF TREES

CS-440-02
Drones
Jaylon Kiper, Cameron Herbert, Bobby Gabriel, Lance Oliphant
3/26/22

1. Introduction:

Technologies like the internet of things(IOT) and artificial intelligence(AI) are becoming more universal to our daily lives. This has led to giant volumes of data being generated from the new types of tech being created. Data could be represented as text, images, speech, or another kind of material. With photos and videos, images make up the majority of all global data creation. This is why there are advanced techniques such as machine learning algorithms. Image classification is considered the best solution towards using all this image data. What happens involves categorizing and assigning labels to groups of pixels or vectors inside an image dependent on certain criteria. The categorization law can be applied through one or multiple spectral or textural characterizations. Our group project involved using images of trees and non-trees, then training a CNN model to recognize the distinction between the two subjects.

Background:

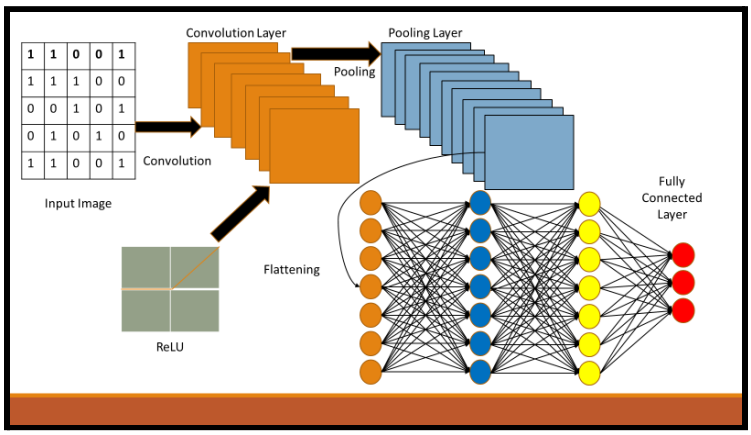
When it comes to performing image classification, the techniques are divided into two categories. This being unsupervised and supervised classification. Unsupervised classification is a fully automated method that does not leverage training data. It uses machine learning algorithms to analyze and cluster unlabeled datasets by discovering hidden patterns or data groups without the need for human intervention. Supervised classification uses previously classified reference samples in order to train the classifier and classify new unknown data. Overall, this is the process of visually choosing samples of training data within the image and allocating them to pre-chosen categories including vegetation, roads, water resources, and buildings. This is done to create statistical measures to be applied to the overall image.

There are several steps when it comes to image classification operating. First, a computer analyzes an image in the form of pixels. This happens by considering the image as an array of matrices with the size of the matrix reliant on the image resolution To put it simply, image classification in a computer's view is the analysis of this statistical data using algorithms. In digital image processing, image classification is done by automatically grouping pixels into specified categories, or "classes". Image recognition using machine learning utilizes the potential of algorithms to learn hidden knowledge from a dataset of organized and unorganized samples. This is also known as supervised learning. Deep learning is the most popular machine learning technique where a lot of hidden layers are used in a model.

Although the task of categorizing an image is instinctive and habitual to humans, it is much more difficult for an automated system to recognize and classify images. The convolutional neural network(CNN) has demonstrated excellent results in computer vision tasks, especially in image classification. CNN is a special form of multi-layer neural network inspired by the mechanism of the optical and neural systems of humans. This framework was developed using machine learning concepts. CNN's can learn and train from data on their own without the need for human intervention. With the pre-processing, they develop and adapt their own image filters, which must be carefully coded for most algorithms and models. CNN frameworks have a set of layers that perform these functions.

One of the beginning steps involves input imaging. For this step, an RGB image is separated by its three-color planes - Red, Green, and Blue. Several color spaces exist within images as well, including Grayscale, HSV, CMYK, etc. The role of the CNN

is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.



2. Project Methodology:

In this section, we introduce the framework and dataset used to evaluate our model. For our project we used primarily the TensorFlow framework with the Keras libraries. Tensorflow is an end-to-end open source platform for machine learning that streamlines the process of building the neural network model. The Keras library within TensorFlow allows for fast prototyping, state-of-the-art research, and production.

2.1 Gathering Images

To build a reliable ML model, we needed to have an extensive set of Training and Testing images. To start, we first gathered all of our drone images we captured during test flights, and data collecting flights. Our criteria for our training model was broken into 2 categories - Tree and no Tree. So, if an image contained a tree in any form or size, we would categorize that in the Tree category. From our Drone imaging we categorized 280 images containing a Tree and put these in our training model.

2.11 Google Image Script

To properly train a model we knew 280 images would not be enough, so we used a

simple python script to scrape images off the internet that additionally contained trees and no trees. For our tree training category, we used keywords such as “Cedar Tree”, “Weeping Willow”, “Walnut Tree”, “Oak Tree”, “Juniper Tree”, “Maple Tree”, and “Teak Tree”. Out of this script, we were able to add an additional 260 images to our training tree model. For the no tree category, we used keywords such as “Cat”, “Dog”, “People”, “Pavement”, “House”, “Car”, “Boat”, “Motorcycle”, “Desert”, and “Bus”. Out of these keywords we were able to add an additional 630 images to our training no tree model.

2.12 Image Augmentation

In addition to our drone imaging and python google script, we still needed to gather more images to add to our model. So, using the Keras library, we were able to augment our existing images to give the neural network more data to work with. This technique of data augmentation is used in many ML models to increase the diversity of the training set by applying random (but realistic) transformations such as image rotation, blurring, mirroring, and zooming. An example of image augmentation on a tree image and no tree image can be found at figure 1.1 and 1.2. The ability to augment images added an additional 750 images to our training tree category and an additional 460 images to our training no tree category.

Figure 1.1 - Image Augmentation on a Tree - blur, zoom, rotation

Figure 1.2 - Image Augmentation on a Motorcycle - blur, rotation, resizing

2.13 Total Images

All the steps we used to gather our training data set, we used to gather our testing data set, except for image augmentation. We did not want image augmentation on our testing set because we want the model to get a clear view of an image when testing. Due to this, our testing set was significantly smaller. Here are the total images:

Training Data Set Total - 2,380 images

Tree - 1,290 images

No Tree - 1,090 images

Testing Data Set Total - 870 images

Tree - 440 images

No Tree - 430 images

2.2 Image Preprocessing

Before we begin training our model, we must first process all the images so the neural network can correctly take them in. Like image augmentation, the image preprocessing

will resize, correctly orient, or correctly recolor an image before taking them in. In our neural network, it is required that all images are the same sized arrays. So preprocessing is a very necessary step.

2.3 Building the CNN model

After collecting our images and properly processing them for the model, we can now put together the necessary layers of the CNN to train our model.

2.31 1st Convolution Layer

Convolution is a procedure where two sources of information are combined. It's an operation that changes a function into something else. In a CNN, kernels are used to detect what features are present throughout an image. In this case it might be an outline of a tree or various leaves. In this process, the convolution converts all pixels into a single value.

2.32 Max Pooling Layer

The pooling layer involves sliding a 2 dimensional filter over each channel of a feature map and summarizing each feature within the region covered by that filter. The pooling layer helps reduce the dimensions of the images and remove unnecessary components and parameters for the model to learn. This, in turn, speeds up the process of training the model and reduces the amount of computations needed to be performed in the network.

2.32 2nd Convolution Layer

After the pooling occurs, we can then add a 2nd convolution layer to additionally help train the model and detect features on the image.

2.33 Flattening Layer

The flattening layer takes the pooled layer's feature map and transforms it into a 1-dimensional array. This part is done so the final step can take on an acceptable input.

2.34 Fully Connected Layer and Output Layer

This is the final classification of the model and it takes all of the pixel data and puts it into one line. It fully connects all the layers to each other so each process can be completed seamlessly. The output layer is the last step in the process and produce the given outputs for the program.

2.4 Training the Model

After building the CNN, we then started training the model using the built in Tensorflow compiler and optimizer. We tested our model with 25 epochs and 60 epochs. The epoch validation accuracy score was found by training the model with the training images then comparing them to the test images. We found that anything higher than 60 epochs did not generate better results. Full results of the model can be found under Section 3.

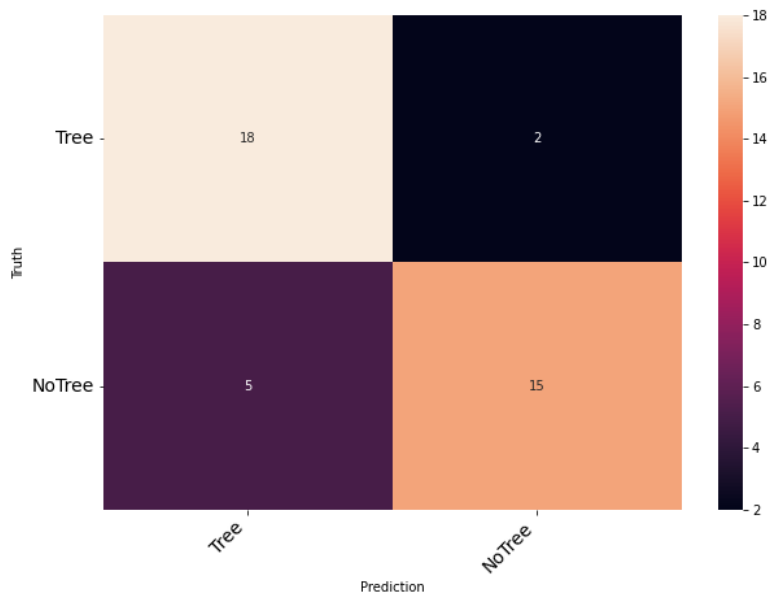
2.5 Model Prediction and Accuracy

As well as using the validation accuracy from the epochs, we also used a 2x2 confusion matrix to test on precision, recall, and give an F1 score. A confusion matrix is used to make direct comparisons of values like True Positives, False Positives, True Negatives, and False Negatives. It also gives important insight to the errors being made by our classifier and the type of errors being made. The results of the confusion matrix and model accuracy can be found in Section 3.

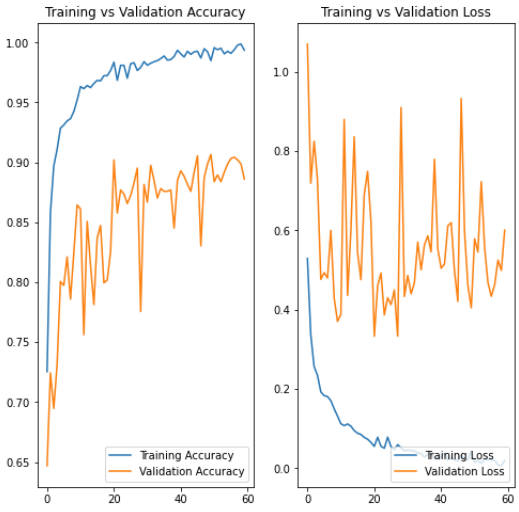
3. Results:

After finishing the CNN training and testing, Confusion matrices and heatmaps were created based on the completion of 25 epochs and 60 epochs. Higher than 60 epochs didn't result in different results, therefore those runs aren't illustrated.

As you can see below, 60 epochs produced a much higher integer on correctly predicting tree and no tree pictures. The model usually didn't detect a tree when there was no tree and didn't detect a tree when an image contained a tree.



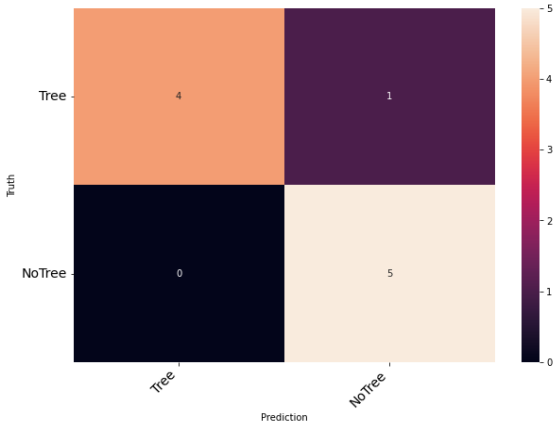
Illustrated on the right is training versus validation accuracy and training versus validation loss. Overall, the training accuracy was much better thanthe validation accuracy which is reflected in the right graph regarding training and validation loss.



	precision	recall	f1-score	support
NoTree	0.78	0.90	0.84	20
Tree	0.88	0.75	0.81	20
accuracy			0.82	40
macro avg	0.83	0.82	0.82	40
weighted avg	0.83	0.82	0.82	40

A confusion matrix was created to show a simpler and more accurate representation of how well the model did. A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This is done using a specific table layout that allows visualization of the performance of an algorithm, where each column of the matrix represents the instances in a predicted class, and each row represents the instances in an actual class. Overall the model had an 82% success rate according to the weighted f-1 score average.

The same model was also run in 25 epochs with the results shown below. The model wasn't able to produce substantial results for the heatmap to be helpful. However, the accuracy and loss graphs showed similar results to 60 epochs.



4. Discussion:

When it comes to the applications for a deep convolutional neural network that can automatically detect the existence of trees, the possibilities are endless. NASA was able to use a similar AI technology to provide vital data to researchers, policymakers, and conservationists for preservation, reforestation, climate change, and other purposes much more efficiently. This technology could also be used by construction project managers looking to build in a forested area. It would greatly simplify the process of finding an area that would require the least amount of trees to be cut down. This technology would also be useful for determining the cost of clearing out the chosen area. The process of counting trees by hand would take a very long time without the help of this technology. There would also be a much greater chance of errors being made.