Capstone Proposal

Robert Young

February 27, 2018

1 Domain Background

An invasive species can be defined as a plant, fungus or animal that is not native to a specific location. The United States Department of Agriculture defines an invasive species as "non-native (or alien) to the ecosystem under consideration and whose introduction causes or is likely to cause economic or environmental harm or harm to human health"[1]. The Kaggle Competition, "Invasive Species Monitoring", recognises the widespread impact of invasive species and emphasises the cost and time involved to track both the location and spread of these invasive species on a large scale.

To monitor an ecosystem and plant distribution requires expert knowledge in the form of trained scientists. These scientists must visit effected areas to examine and record the invasive species present. This is time consuming, expensive, and not feasible on a vast scale. The objective of this competition is to identify the invasive species, *Hydrangea*, in images taken in the Brazilian national forest, using machine learning techniques. From the training images and labels provided, the designed model should predict the presence of this invasive species in the testing pictures.

2 Problem Statement

In this capstone, the problem can be solved by creating a machine learning algorithm that can identify, with reasonable accuracy, if an invasive plant is present in images of forests and foliage. This will be treated as an image classification problem, where the

algorithm output will be the probability of an image containing the invasive species *Hydrangea*. The work of Razavian et al [7] found that features obtained from deep learning with a convolutional neural network should be considered as the primary tool for tasks requiring image recognition. To this end, I intend to solve this problem using a pre-trained convolutional neural network.

The model created will be trained, tested and validated against the "Invasive Species Monitoring" Kaggle Competition datasets [4]. The model accuracy will be validated on the area under the ROC curve [10] between predicted probability and the observed target. The model will examine 1531 test images. In this competition, there is only one invasive species of interest, but this model, once developed, could be retrained applied to other plant based classification tasks.

3 Datasets and Inputs

For this Capstone, three datasets will be used, as provided on the Kaggle Competition page. These are the following:

- train.7z - This is the training set containing 2295 images [6].
- train_labels.csv - These are the correct labels for the training set [6].
- test.7z - This is the testing set containing 1531 images [6].

3.1 Training Set

The training set is made up of 2295 different images. These are compiled of photos of forest and foliage, as can be seen in Figure 1. All images are of the JPEG filetype, no larger than 1.6MB in size, with identical dimensions of 1154 x 866. This set of data will be the training examples my model will use for learning. As I intend to develop a model based on a Convolution Neural Network, I will also need to split out a portion of my training set data for use in a validation set. Whereas the training set is used to fit the model weights, the validation set will be used to see how the model is performing. This is to ensure, ideally, that our model is not overfitting to the training set. The model which produces the lowest validation loss will be selected.

Figure 1: Example of Supplied Training Image

3.2 Training Set Labels

The train_labels.csv file contains the correct labels for the training set. This comes in the form of two columns:

- 'name' - This is a number associated with the image name (also a number).
- 'invasive' - This is a binary value, which states whether an invasive species is present in the image or not.

These labels will be used in conjunction with the training set, so the model can learn through training if an invasive species is present in an image.

3.3 Testing Set

The testing set is made up of 1531 different images. Similar to the training set, the testing set is also made up of images taken of forest and foliage. An example of the provided images can be seen below in Figure 2. These are all filetype JPEG, no larger than 1.6MB in size, with dimensions of 1154 x 866. If my model, which is fit to the training dataset, also fits well to the testing set, a minimal amount of overfitting will have taken place. If better fitting of the training dataset versus the test dataset occurs, this would suggest overfitting [8]. This set will be used to assess the performance of my fully specified classifier by checking the accuracy of the trained model.

Figure 2: Example of Supplied Test Image

4 Solution Statement

My proposed solution to this problem is to apply Deep Learning in the form of a Convolutional Neural Network (CNN). For image processing and classification, a CNN is preferable over a Multilayer Perceptron (MLP) in the case of real world, messy data. This is as an MLP converts images into vectors, meaning it has no knowledge that these numbers were originally spatially arranged as a grid. A CNN however, understands that image pixels in close proximity are more related than those that are further apart [9]. Artificial systems have difficulty recognising objects in the same way as the human visual system. This is due to artificial systems struggling as a result of viewpoint-dependant object variability and the high in-class variability of object types, in contrast to the human visual system. Certain CNN types are the best adaptive image recognisers, when provided with a labelled dataset of adequate size [2].

My intention is to adapt a CNN through transfer learning for image detection of the invasive plant species. This will allow me to take the learned understanding of CNN architectures and use it in my deep learning model. How I apply transfer learning depend on:

- The size of the data set
- The similarity to training data

This will be discussed in more detail in Section 7.

The resulting CNN will be able to analyse an image, and predict the probability of an invasive species being present in this image. The performance will be evaluated based on the area under the ROC curve between the predicted probability and the observed target. The purpose of this competition is to detect a specific invasive species, *Hydrangea*, in the Brazilian national forest. This model, if successful, could be retrained to detect other invasive species.

5 Benchmark Model

The data set and task is based on the Kaggle Competition, Invasive Species Monitoring - Identify Images of Invasive Hydrangea. The public Leaderboard, ranked based on the evaluation criteria, will be used as the foundation for benchmarking my model. Submissions are evaluated on the area under the ROC curve between the predicted probability and the observed target. This will be discussed in more detail in Section 6. The Kaggle Public Leaderboard which has 1794 entries, was downloaded, and this data was used to calculate benchmark values evaluated on the area under the receiver operator curve:

- Average performance Kaggle Leaderboard = 0.902378093
- 80th percentile performance Kaggle Leaderboard = 0.98887

Based on these values, my model should aim to achieve greater than the average performance present on the Kaggle Leaderboard of 0.9023. I will set a stretch target of trying to achieve performance that meets or exceeds the 80th percentile of the Kaggle Leaderboard.

6 Evaluation Metrics

As per the Kaggle Competition evaluation rules, submissions are evaluated on the area under the ROC curve (AUC) between the predicted probability and the observed target. A receiver operating characteristic curve is described as "a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied" [10]. It is created by plotting the true positive rate against the false positive rate. For my model, this is a two-class prediction problem, also known as binary classification. The ROC AUC varies between a value of 0 and 1. with an uninformative class yielding 0.5.

The evaluation metric will take the ROC curve and evaluate against the area under the predicted probability and observed target. Using normalised units, the area under the curve is equal to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Mathematically, this can be expressed as follows [3]:

$$A = \int_{\infty}^{-\infty} TPR(T)(-FPR'(T))\,dT = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(T' > T) f_1(T') f_0(T)\,dT'\,dT = P(X_1 > X_0)$$

Where:

- TPR is the true positive rate.
- FPR is the false positive rate.
- X is a continuous random variable.
- T is a given threshold parameter.
- f_0 probability density if instance X does not belong to a positive class.
- f_1 probability density if instance X belongs to a positive class.
- X_1 score for a positive instance.
- X_0 score for a negative instance.

7 Project Design

The workflow for this project can be broken down into three key steps. These will be discussed in detail in the following subsections:

- Data Processing
- CNN Creation
- Model Evaluation

7.1 Data Processing

The project design will begin with the acquisition and import of the relevant datasets. These will be loaded and split between the training set, the testing set and the validation set. Once completed, dataset statistics will be obtained such as the total number of images, the number of training images, the number of testing images, the number of validation images and the number of test images. As Keras CNNs require a 4D array as an input, the input images will be converted to an array and resized to a 4D tensor. The images are in colour, which means the number of channels will be equal to 3. The required tensor will have the following shape:

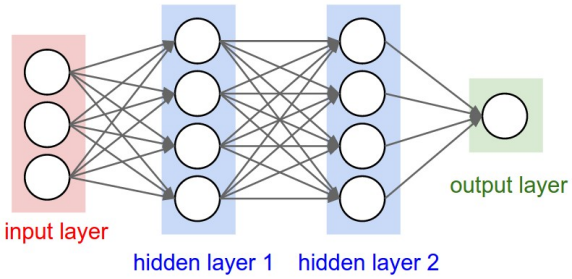
$$(1, 224, 224, 3)$$

Furthermore, as the intention is to use a pre-trained Keras model for my CNN, an additional processing step is required. This involves converting RGB images to BGR by re-ordering the channels. Pre-trained models will also undergo a normalisation step where the mean pixel must be subtracted from every pixel in each image.

7.2 CNN Creation

With the data analysed and processed, the next step is model architecture definition. The CNN will take the structure of a regular 3-layer neural network, as seen in Figure 3. To take advantage of the years of research, model training, and parameter tuning, transfer learning will be applied to the CNN.

Figure 3: Regular 3-Layer Neural Network [5]



Adaption of the CNN architecture will depend on which of the following four cases applies:

- If the new data set is small and similar to the original training data - change the end of the convolutional network.
- If the new data set is small and different to the original training data - change the start of the convolutional network.
- If the new data set is large and similar to the original training data - fine-tune the neural network.
- If the new data set is large and different to the original training data - fine-tune or retrain the neural network.

Based on the predicted size and complexity of the dataset, my intention is to proceed with a model architecture where the last convolutional output of the bottleneck feature is fed as input to my model. A global average pooling layer and a fully connected layer will also be added, with the fully connected layer deploying softmax.

The intention is to test my model using the following five bottleneck features - VGG-16, VGG-19, ResNet-50, Xception and Inception. The model will be compiled with each of the bottleneck features, and then trained. Upon selection of the best performing bottleneck feature, I will then tune and adjust the loss function, optimiser, number of epochs and batch size parameters.

7.3 Model Evaluation

When each model has been trained and compiled, I will load the model with the best validation loss. I will then test the model performance based on test accuracy, and the Kaggle Competion evaluation metrics, as discussed in Section 6.

References

[1] National Invasive Species Information Center. What is an invasive species? *United States Department of Agriculture: National Agriculture Library*, 2006.

[2] Dan C et al. Flexible, high performance convolutional neural networks for image classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 3, 2011.

[3] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[4] Kaggle. Invasive species monitoring. *Kaggle*, 2017.

[5] Andrej Karpathy. Convolutional neural networks for visual recognition. online, 2018.

[6] Christian Requena Mesa, Thore Engel, Amrita Menon, and Emma Bradley., 2017. Data from Brazilian rainforest. Hydrangea invasive species.

[7] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, pages 512–519. IEEE Computer Society, 2014.

[8] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press, 1996.

[9] Udacity. Machine learning nanodegree course. online, 2018.

[10] Wikipedia. Receiver operating characteristic. *Wikipedia*, 2018.