Tree Species Classification with Deep Learning

Introduction:

Classification is a machine learning model used to predict mutually exclusive categories when there are no continuous values. It does so by using labels to classify data, thus predicting a discrete number of values.

Deep Learning is one of many of machine learning methods which bases itself on artificial neural networks with multiple layers of processing that are used to extract progressively higher-level features from data; such can be supervised, semi-supervised or unsupervised.

This Contextual Image Classification Project makes use of Deep Learning to train a Classification model to recognize and correctly classify tree species in an urban context. Deep Learning models often require a considerable amount of relevant training data to be able to provide good results, thus the collection of 400 eye-picked images that depict trees in different settings, seasons and lightings. When compiling this dataset, I had extra care to ensure the images I was collecting were either public domain or free to use as long as the owner is acknowledged; to ensure this, I created a csv file with the sources for each image.

Proposed solution:

Trees vary in shape, colour, density and foliage, which can become a struggle for AI to determine a pattern as the model will have to be trained to recognize intra-class variations (different species of trees), scale variations (trees have different sizes), perspective variation (images portray trees from different angles), different illumination levels (influenced by different stages of the day such as morning, dusk, or night), and background clutter, to name a few. Due to all these factors, I highly based myself on a research paper [1] on tree species classification to design an architecture that best suits the task.

To better train my module, I collected data (520 images) from two different urban environments with different biomes and cultural settings, thus having two cities which will provide enough diversity in terms of tree species, scale, perspective, illumination and clutter, strengthening my model to more accurate and precise levels.

I chose to work with rather small datasets (training = 260, testing = 260, validating = combination of 260 random images from both training and testing sets) as this will result in noisy updates to model weights; that is to say that there will be many updates with different estimates of the gradient error.

This can be useful, resulting in faster learning and (sometimes) a more robust model as noisier batch sizes offer better regularizing effects, lower generalization error, and make it easier to fit one batch worth of training data in GPU memory.

My solution was an implementation of VGG16 as it has a simple and straightforward architecture (13 convolutional layers and 3 fully connected layers), has state-of-the-art performance in many image datasets, has been pre-trained on the ImageNet dataset which means it can be used as a starting point for training new image classification models (aka transfer learning which significantly reduces the amount of data and time needed to train a model from scratch), and because this CNN highly benefits from its open-source model that is widely for popular deep learning frameworks such as TensorFlow which I used to implement my code.

Results:

I ran each model once with 10 epochs, achieve an average of 63% in accuracy for City A and an average of 78% for City B. In cross-scenario examples, City A’s model on City B’s dataset performed better than City B’s model on City A’s data: this was expected as City A has a wider variety of flora (more diverse tree species, different colour and shape in foliage) as compared to City B which has more similarly looking trees, thus limiting what City B’s model perceived as a tree. Nonetheless, City B’s model performed better in classifying images in the “not” label (does not have tree in image), which I believe to be due to City B’s tree species being more similar which allows the module to grow more robust when it comes to learning to identify an object as a tree.

Conclusion:

In conclusion, the models worked satisfyingly with an average of 60-80% accuracy (depending on the model and if it was being used on its city or on the other city). The model for City A performed better in the cross scenario example as the tree species from City A are similar to the majority of species from City B, yet the module from City B performed better in both scenarios when it came to label images as containing a tree or not when compared to how City A’s module analysed this.

Sources:

[1] C. Zhang, K. Xia, H. Feng, Y. Yang, and X. Du, “Tree species classification using deep learning and RGB optical images obtained by an unmanned aerial vehicle,” Journal of Forestry Research, vol. 32, no. 5, pp. 1879–1888, 2020.

[2] J. Brownlee, “How to control the stability of training neural networks with the batch size,” MachineLearningMastery.com, 27-Aug-2020. [Online]. Available: https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/. [Accessed: 24-Mar-2023].

[3] K. Team, “Keras documentation: Introduction to keras for engineers,” Keras. [Online]. Available: https://keras.io/getting_started/intro_to_keras_for_engineers/. [Accessed: 25-Mar-2023].

[4] J. Brownlee, “How to train an object detection model with keras,” MachineLearningMastery.com, 01-Sep-2020. [Online]. Available: https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/. [Accessed: 25-Mar-2023].

[5] Matterport, “Mask R-CNN for Object Detection and Segmentation,” GitHub, 19-Mar-2018. [Online]. Available: https://github.com/matterport/Mask_RCNN. [Accessed: 25-Mar-2023].

[6] “Image classifier using CNN,” GeeksforGeeks, 11-Jan-2023. [Online]. Available: https://www.geeksforgeeks.org/image-classifier-using-cnn/. [Accessed: 25-Mar-2023].

[7] “Python: Image classification using keras,” GeeksforGeeks, 03-Feb-2023. [Online]. Available: https://www.geeksforgeeks.org/python-image-classification-using-keras/. [Accessed: 25-Mar-2023].