CSCU9M6 Classification Project

Hunter Lowery
Stirling, Scotland
hul00047@students.stir.ac.uk

I. Introduction

In this assignment, the task to be undertaken is to classify certain objects within aerial images from two different datasets; a set of images from Branson, Missouri, which may be referred to as 'City A', and a set of images from Stirling, Scotland, which may be referred to as 'City B'. All images will be gathered from Google Earth.

The objects to be classified these images are trees and cars. The goal of this task is to take an image, and determine whether or not it contains a tree or a car; it will also determine whether it contains both, or neither. Whether the image is from Branson or Stirling will not be contained in the classifier. The images come from two different cities simply so that the model does not become overfitted for a specific city. For example, if the images were only from Stirling, to identify a car it might look for the yellow of a license plate, or for specific characteristics of the trees native to Stirling.

II. Proposed Solution with Justifications

To accomplish the task of classification, a Deep Learning approach will be used, more specifically, a Convolutional Neural Network (CNN) will be used. The specific model chosen will be AlexNet.

The model will not come pre-trained, and will only train on the datasets of city A and city B. This will reduce accuracy of the model specifically in this assignment, and will also mean it is less applicable more generally to image classification. But by not pre-training the model, this means that the results of the model can be analyzed to more accurately see how choices made with the dataset, data preparation, and parameters impact the training, and how suited the model itself is toward this task.

A CNN was chosen because this type of network is particularly well suited towards classifying images. Through a method of convolution, images are able to be transformed or deciphered easily in order to find common features through which to classify images.

AlexNet was chosen specifically because while it is not state-of-the-art in terms of image classification, it once was, and it is small enough in terms of layers and simple enough to understand that the model can be understood and computations computed in a reasonable amount of time to complete this project. To construct the model and run tests, the tutorial on how to accomplish this in PyTorch from the PyTorch official documentation will be followed, found here. [1]

III. Results

The results from training and testing the models was that, in the best performance, models would obtain an accuracy in the range of 80 to 90 percent in training, but would only show an accuracy of around 60 to 70 percent in testing. This accuracy would fall even further, to the 40 to 50 percent range, when the test data was from the city the model was not trained on. And only one set of parameters achieved results like this; out of the 10 models, 8 of them performed considerably worse than this in at least one metric.

Model Accuracy			
Model	***Training***[a]	***Testing***	***Testing (Opposite City)***
Branson1	72%	68%	48%
Branson2	86%	62%	45%
Branson3	61%	59%	46%
Branson4	59%	59%	46%
Branson5	58%	61%	46%
Stirling1	54%	47%	53%
Stirling2	92%	60%	42%
Stirling3	37%	46%	59%
Stirling4	34%	09%	17%
Stirling5	35%	33%	15%

[a]The accuracy presented here is the accuracy of the final epoch in training.

To see the parameters used in each model, see Figures 1 through 10, which can be found at the end of this document, or the list in the next section.

IV. Discussion

The training and testing of the models revolved around the changing of two parameters; the number of epochs, and the learning rate. The models were constructed as follows:

- Model 1 (Standard): 20 Epochs, 0.001 Learning Rate.
- Model 2: 50 Epochs, 0.001 Learning Rate.
- Model 3: 4 Epochs, 0.001 Learning Rate.
- Model 4: 20 Epochs, 0.9 Learning Rate.
- Model 5: 20 Epochs, 0.02 Learning Rate.

Model 2 performed the best in training, and still performed well in both rounds of testing. The Branson model 2 was beaten with the Branson standard model in both rounds of testing, while the Stirling model 2 was only beaten when by the Stirling standard model the test dataset was the city of Branson. The standard model and model 2 are the only two models that showed consistent growth in accuracy over the course of the training. Model 3 was not trained for enough epochs to achieve growth, and models 4 and 5 had inconsistent accuracy across all epochs.

By looking at the confusion matrices and the results of training, a problem in the data can be found; this is particularly striking in Figure 3. In model 3, the model would predict an image of trees 100% of the time. By doing so, it achieved a 61% accuracy.

It can also be seen in Figure 2, that although the training model achieved 86% accuracy, it only correctly identified images where neither feature, neither trees nor cars, were present 11% of the time.

Seen in Figure 7, Stirling Model 2 has a confusion matrix that reflects one of a desirable, accurate model, where each classification has a high accuracy, and also one close to the overall accuracy of the model.

All of this information would suggest that there's a problem present in the data, which is more pronounced in the Branson Dataset than the Stirling Dataset, which causes the models to overly bias toward predicting that the image is a tree. But the success of Stirling Model 2 suggests that the underlying model of the AlexNet is appropriate and can be properly trained for this task.

V. Conclusion

In conclusion, the models trained were shown to have successes in some areas, but ultimately fall short of what is known to be possible in others. The primary culprit for the lower accuracy is suspected to be problems in the data set. The images gathered for each classifier are not balanced in terms of number, and given the drop in accuracy in testing, the images gathered are not diverse enough to prevent overfitting to the training dataset. Some of this could probably be mitigated by choosing a more complex model, but a high accuracy, in the 85-95% range, is possible with the AlexNet if more samples were gathered, and the number of images for each classifiers were properly balanced in number.

References

[1] Datasets & dataloaders¶ (no date) Datasets & DataLoaders - PyTorch Tutorials 2.0.0+cu117 documentation. Available at: https://pytorch.org/tutorials/beginner/basics/data_tutorial.html (Accessed: March 27, 2023).

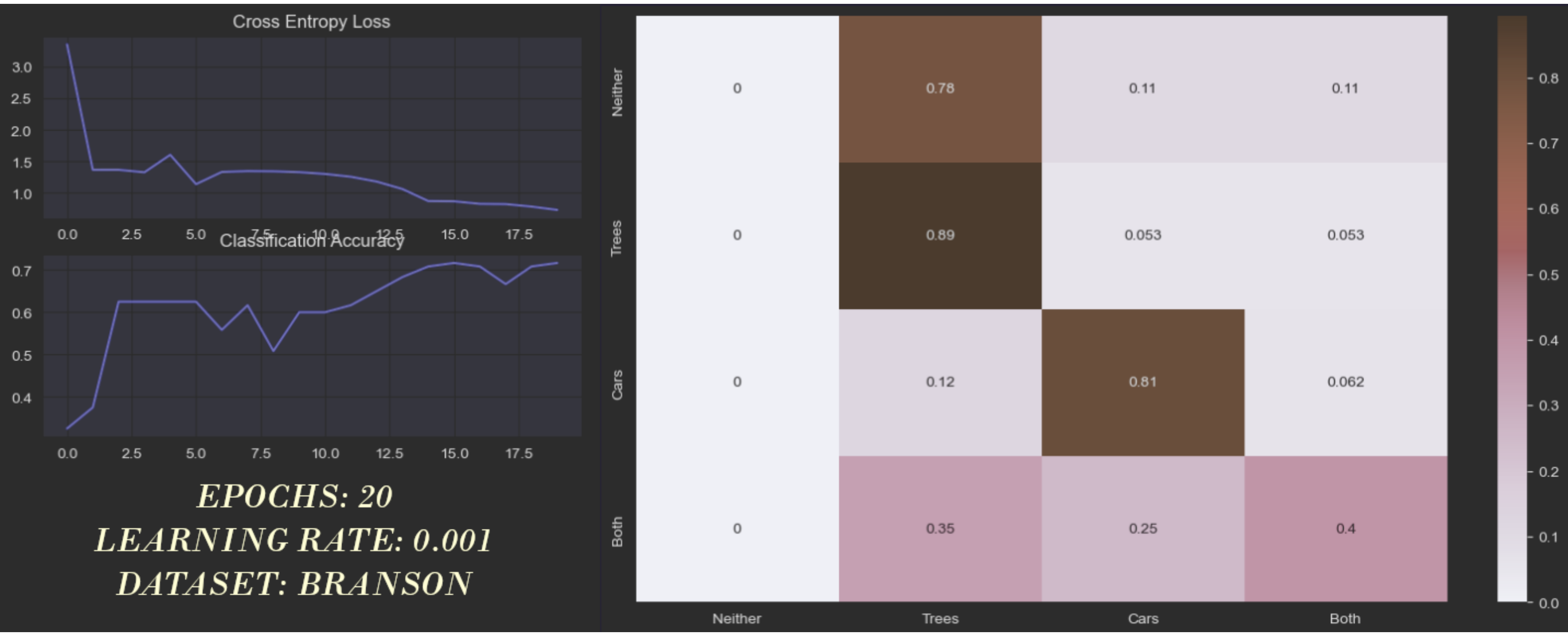

Fig. 1. Branson Model 1's cross entropy and classification accuracy over training, and its confusion matrix.

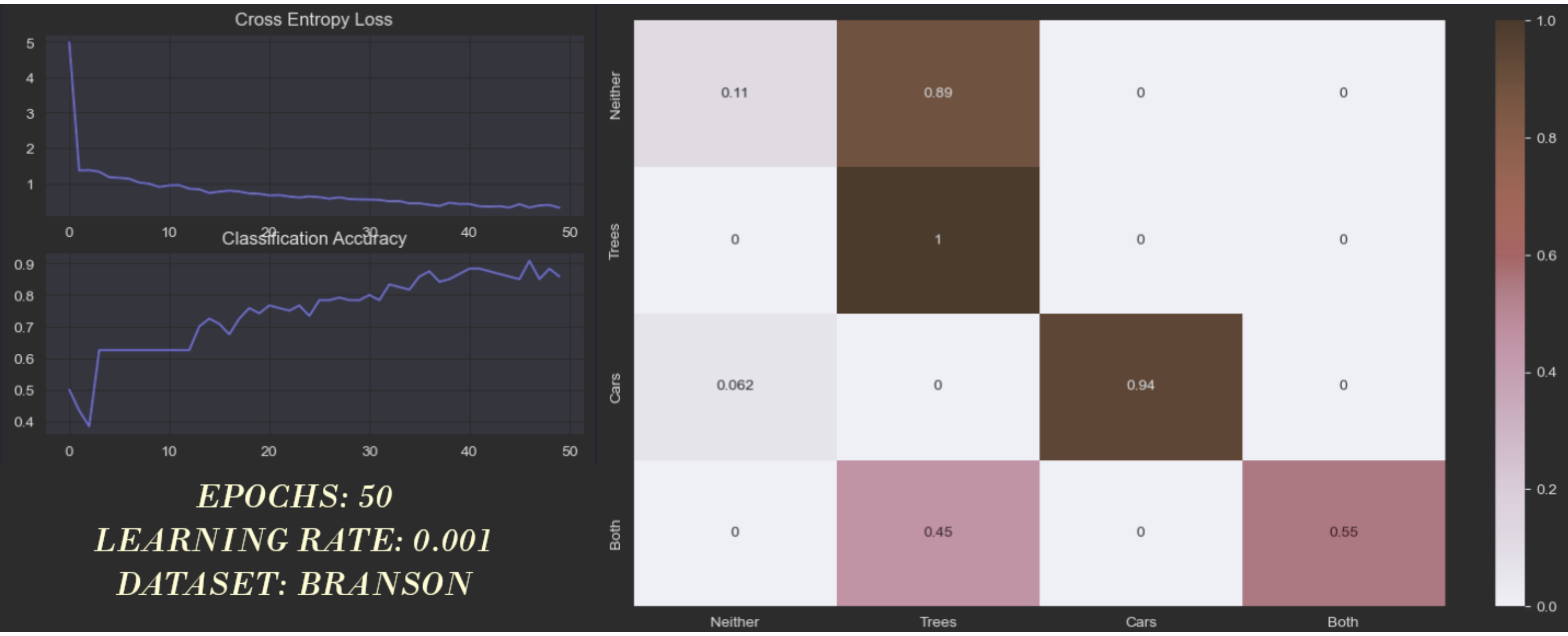

Fig. 2. Branson Model 2's cross entropy and classification accuracy over training, and its confusion matrix.

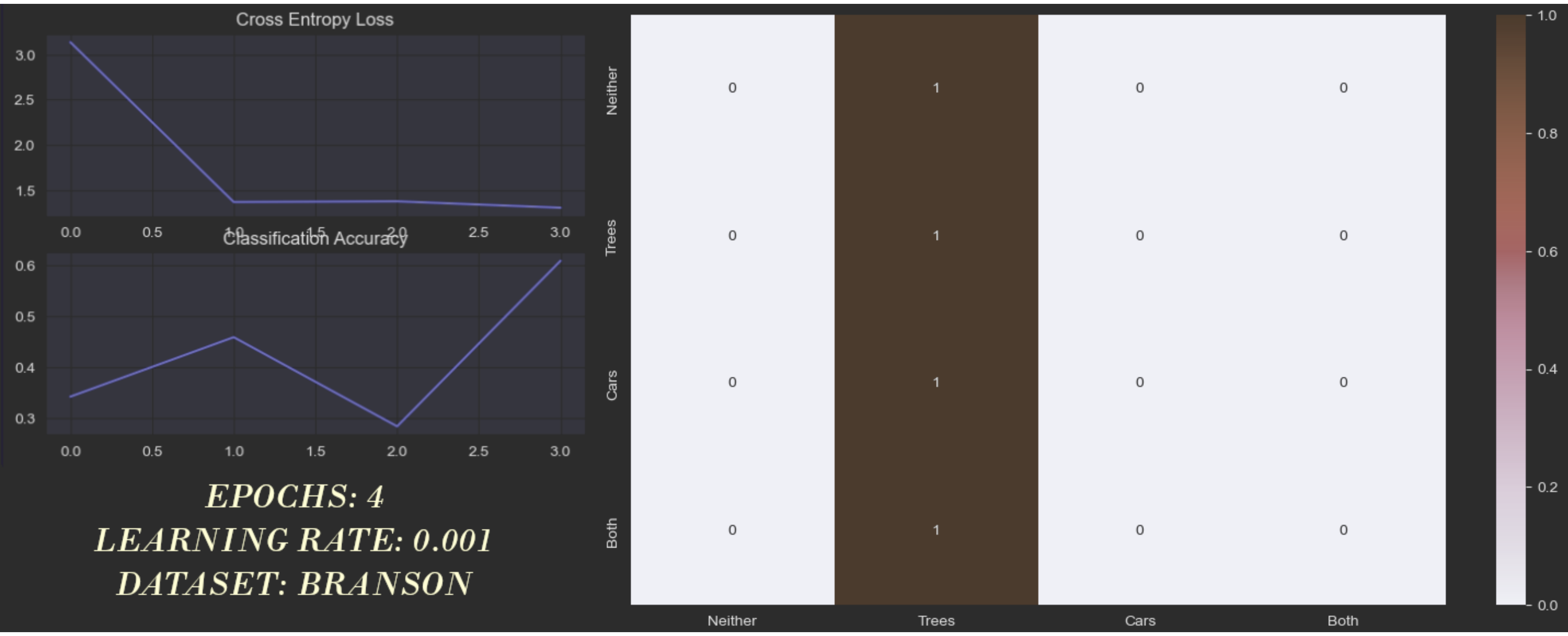

Fig. 3. Branson Model 3's cross entropy and classification accuracy over training, and its confusion matrix.

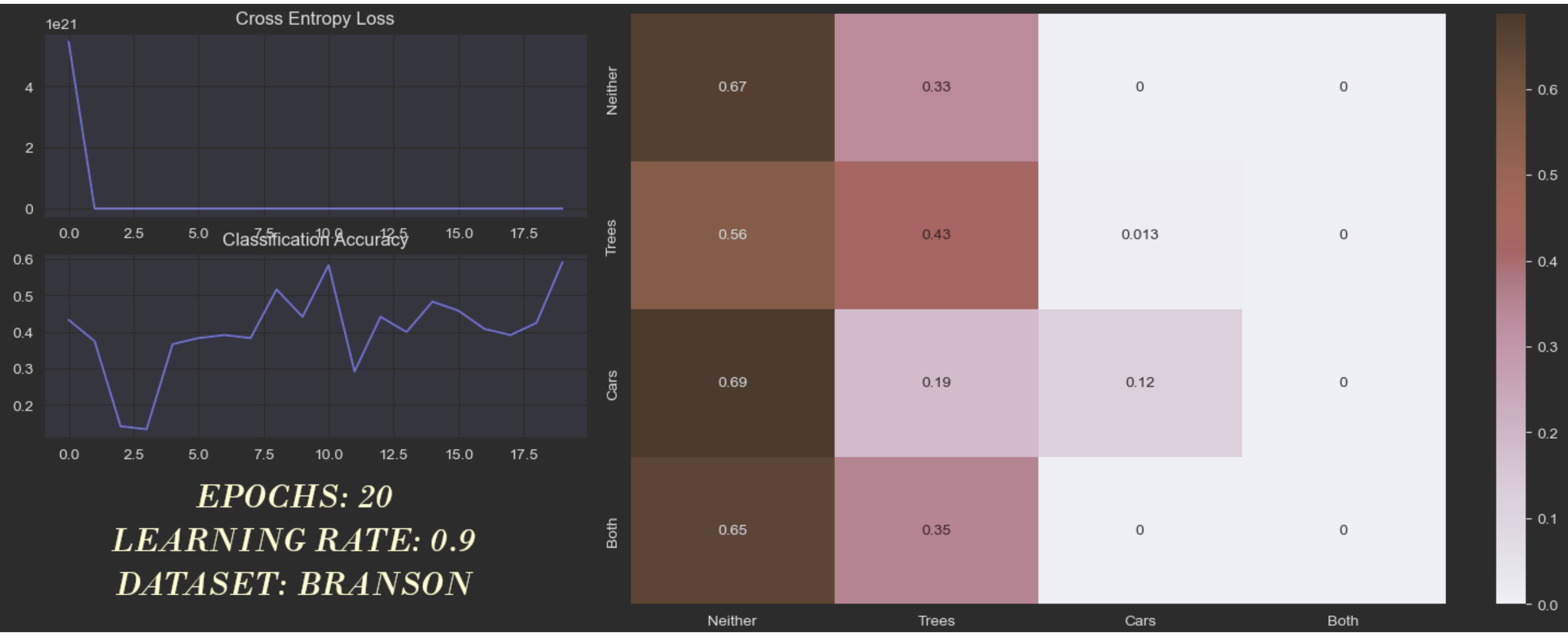

Fig. 4. Branson Model 4's cross entropy and classification accuracy over training, and its confusion matrix.

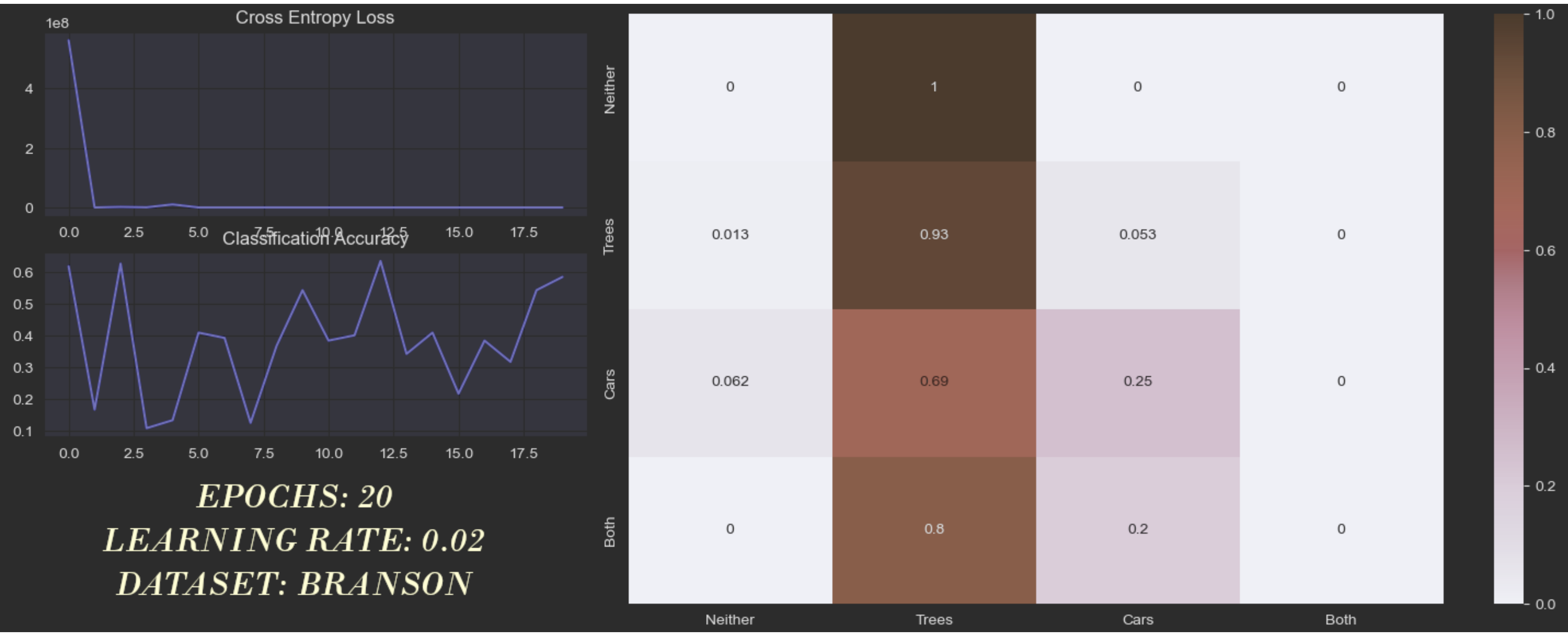

Fig. 5. Branson Model 5's cross entropy and classification accuracy over training, and its confusion matrix.

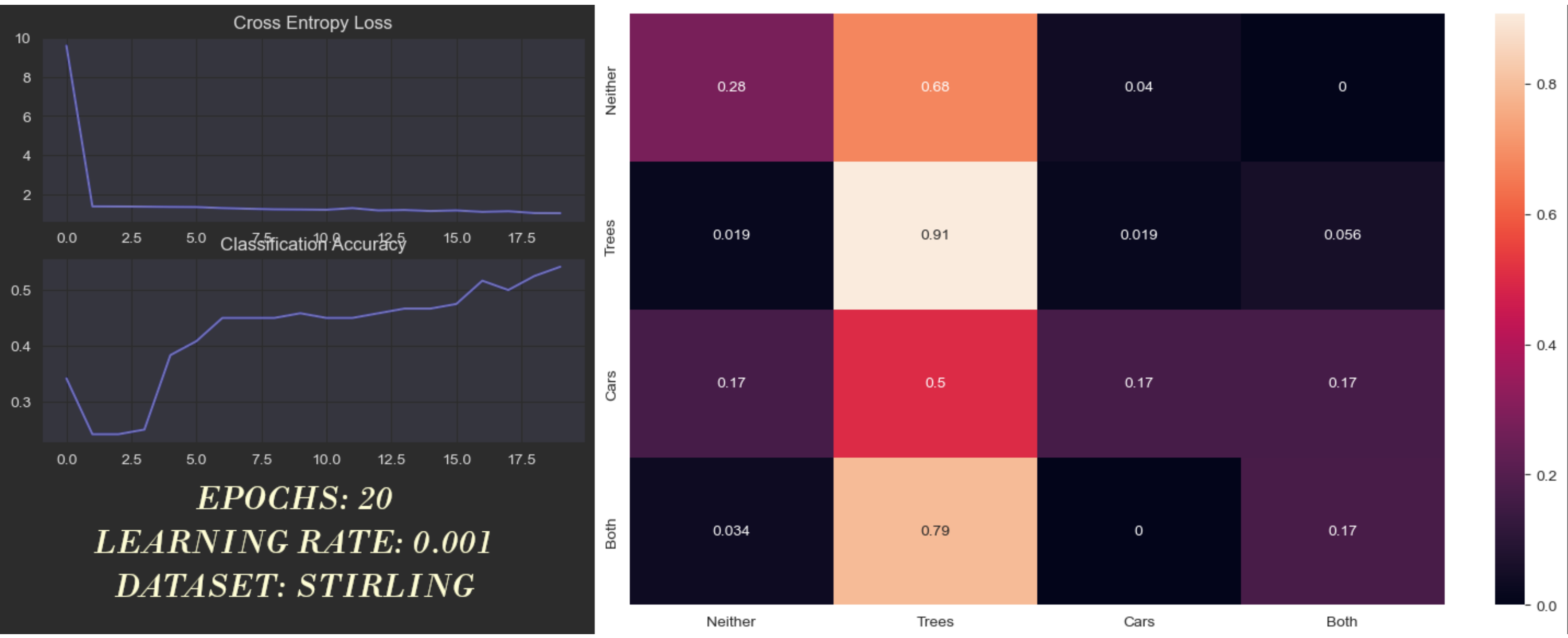

Fig. 6. Stirling Model 1's cross entropy and classification accuracy over training, and its confusion matrix.

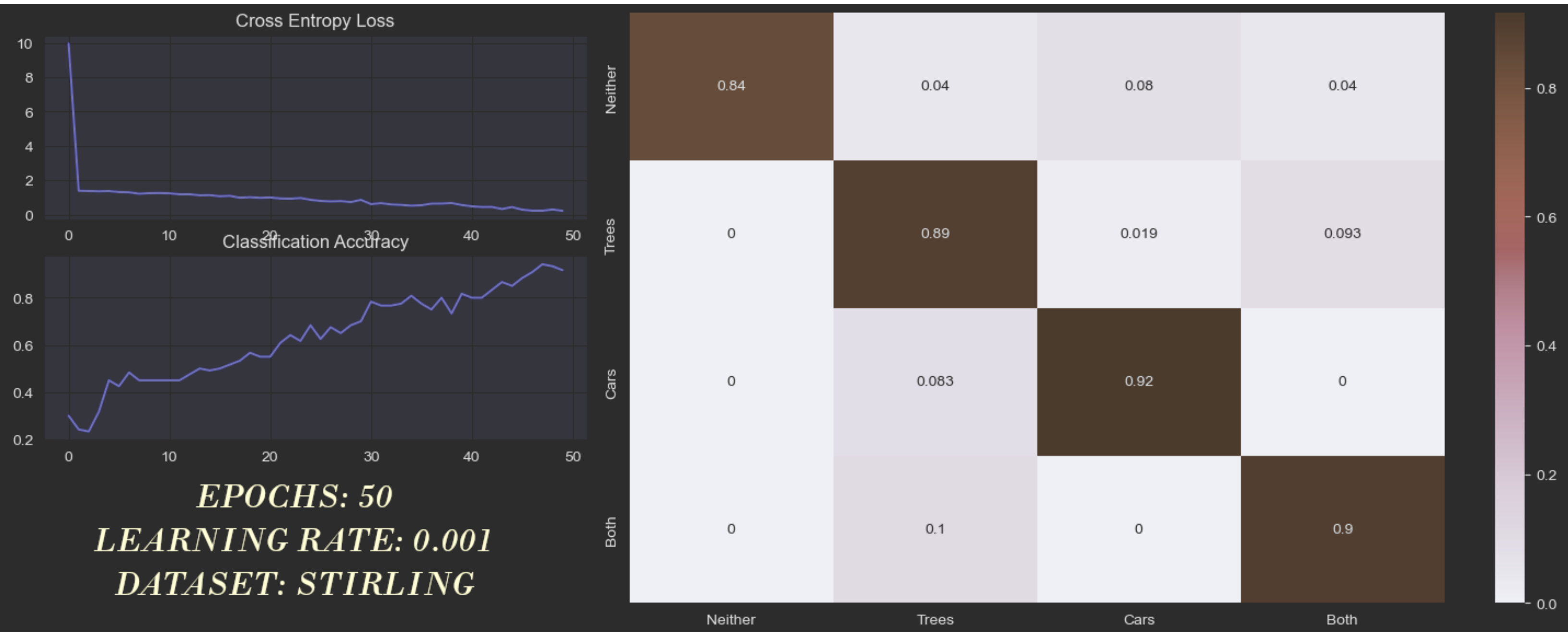

Fig. 7. Stirling Model 1's cross entropy and classification accuracy over training, and its confusion matrix.

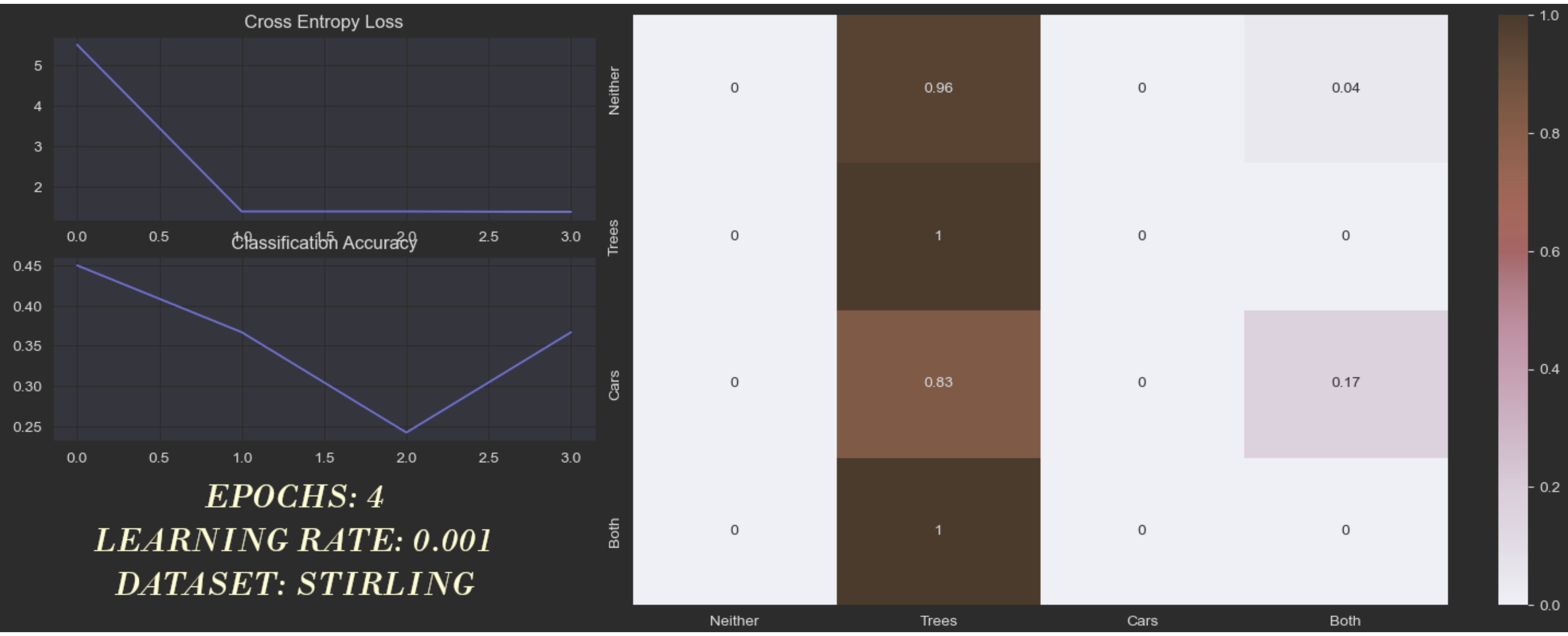

Fig. 8. Stirling Model 3's cross entropy and classification accuracy over training, and its confusion matrix.

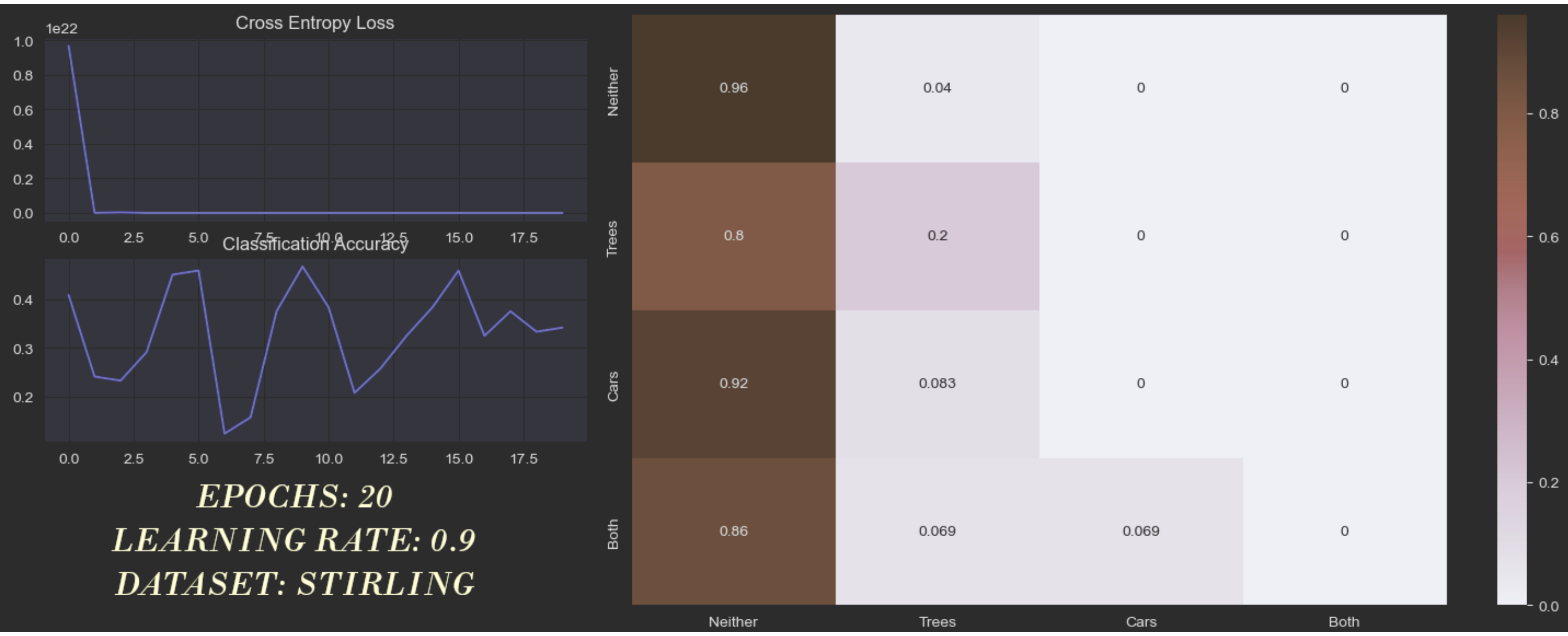

Fig. 9. Stirling Model 4's cross entropy and classification accuracy over training, and its confusion matrix.

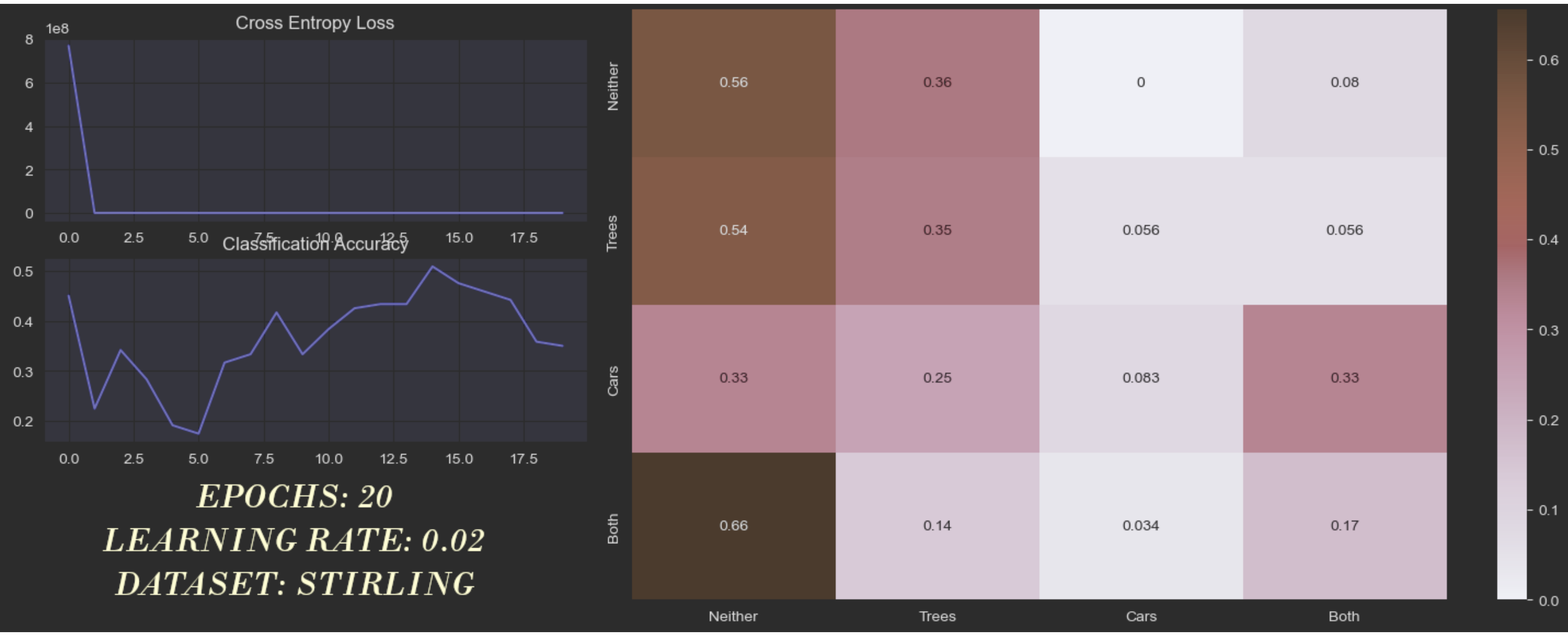

Fig. 10. Stirling Model 5's cross entropy and classification accuracy over training, and its confusion matrix.