Teah Markstone

Madeline Grace McLaughlin

Professor Bálint Gyires-Tóth

Deep Learning Spring 2022

Mushroom Genus Classification

Introduction

For our Deep Learning project, we decided to perform image classification on different mushrooms. We thought this kind of model would be useful if it was embedded in a mobile application so that users would be able to identify information about mushrooms they find in the wild. Instead of classifying by species, as a diverse enough dataset would require too many classes, we decided to classify by genus.

Previous solutions and related work

Ideally, this model could be used in conjunction with a mushroom identifier mobile application so that users could take a picture of a mushroom in the wild and identify to which genus it belongs. Our work is based on the fact that mushrooms can be incredibly hard to identify and that when trying to figure out whether a given mushroom is poisonous or not, simple rules may not be sufficient. One similar paper that we looked at to focus our project more and identify how the methods used in related ML/DL work was Classification of multicategory edible fungi based on the infrared spectra of caps and stalks, which focused on categorizing mushrooms in order to identify edible vs. poisonous mushrooms.

Dataset

We used a dataset of the most common North European mushroom genuses (via @CatoDogo on Kaggle), with between 300 and 1500 individual examples of each mushroom genus. The genuses

included are Agaricus, Amanita, Boletus, Cortinarius, Entoloma, Hygrocybe, Lactariusm, Russula, and Suillus. Before training our model, we split our data into training, validation, and test sets (75%/15%/10% respectively) and practiced displaying arbitrary sample images from each class to ensure that the images were copied over correctly. From there, we created image datasets (training, validation, and test subsets of the original shuffled dataset) using the TensorFlow ImageDataGenerator. Target size was 299, class mode was categorical, and batch size was 64. This found 9 classes of images that included 5033 training images, 1005 validation images, and 676 testing images for our model to use.

Proposed Method

We used the pre-trained EfficientNetB7 and a sequential model for our classifier model. First we instantiated the EfficientNetB7 model without final layers, and then constructed a sequential model with the pretrained model and a final dense layer with 9 nodes for the 9 classes and softmax activation. We also used EarlyStopping and ReduceLROnPlateau for callbacks during training. We trained the model over 100 epochs, with the steps per epoch = (number of training samples / batch size) and validation steps = (number of validation samples / batch size).

Evaluation Method

To evaluate our model's performance, we used a number of metrics and graphs. First we wanted to see how loss was affected during training. Using the history object of the model, we were able to graph the training loss and the validation loss over the epochs. We evaluated accuracy in a similar way and graphed the accuracy on the training data and validation data over the epochs. Next we used the model to generate predictions on the test data in order to generate a confusion matrix. This confusion matrix is 9x9 where the rows represent the true class and the columns represent the predicted class. Thus the correct predictions lie on the diagonal of the matrix.

Lastly we used the classification report function from sklearn to display the precision, recall, and f1-score for each class.

Results

Our model had 64,120,736 total parameters, 23,049 of which were trainable. After running our model for 100 epochs, we see that our accuracy improved from 63.98% to 77.91%. Our loss ended at 0.4832 and we saw an inverse relationship between loss and accuracy in training: as accuracy exponentially increased and plateaued off, we found loss to drop significantly and plateau off as well. We found that the precision varied based on the genus, with Agaricus having the lowest precision (0.71) and Hygrocybe the highest (0.96); this could be due to the variance in mushroom characteristics by genus. Overall, we found a weighted average of 80%+ accuracy.

Discussion

Our model performed relatively well. I think we would have gotten better results had we used an additional dataset, but even without that we managed to have 85% accuracy for classifying mushrooms. Additionally, when we took up this project we knew having high accuracy would be difficult as mushrooms are notoriously difficult to identify as they often try to copy other types of mushrooms for survival. According to the classification report, the model had the hardest time classifying the Agaricus genus of mushrooms, with 71% precision. The genus that the model classified the most accurately was the Hygrocybe genus, with 96% precision. By looking at images of mushrooms in this genus we can see that they are very different from the other mushroom genuses as they all have a fairly long stem and a waxy top, usually bright green or red. Most of the other genuses have a mix of brown or white mushrooms that look like they could belong to multiple other genuses. By just a visual analysis we can see why the model was

able to perform better on some genuses than others, which is confirmation that our model performed as expected.