Luke Moore

Claudio Delrieux

COMP 4449

October 12, 2022

Quail Image Classification using CUDA and VGG Neural Networks

The purpose of this project is to develop a further understanding of how neural networks are used for image classification tasks, and to examine various VGG networks on a specific quail classification problem. Three pretrained VGG neural networks (VGG11, VGG16 and VGG19) were reconfigured to be trained on a dataset with three classifications of quail. VGG11 was the best performing model with approximately 98% accuracy.

The dataset was gathered from Kaggle (Kaggle, 2022). The dataset contains images of 450 bird species in 224x224x3 colored .jpg format. However, only 3 species of birds were chosen to perform a smaller scale image classification. From the larger dataset, images of California Quail, Gambel's Quail and Harlequin quail were selected. California quail and Gambel's quail are similar characteristically, while the Harlequin quail has much more distinctive characteristics (see appendix A). The final dataset was approximately 450 training images (about 150 for each bird), 15 validation images (5 for each bird), and 15 test images (5 for each bird).

To prepare the data for image classification in the VGG networks, some basic transformations were added to the data loading step to provide some variation for the model to train on. Random rotations, random flips, random crops and padding were used. VGG networks require specific image sizes that were already met by the dataset, so there was very little pre-processing done on the dataset to prepare it for modeling.

Three VGG networks were downloaded from the Pytorch module. Each network was loaded with predefined weights based on previous training with IMAGENET. IMAGENET is a large set of annotated photographs with over 21,000 groups. To accurately predict on a quail-only dataset, the final layer of the network was modified with three output features instead of 1000. To ensure the probability of each class being calculated correctly, this is the most important change. The accuracy of each network is generated using the Softmax Function, a multinomial logistic regression where the combination of each class probabilities equals 1. The criterion for loss estimation was a cross-entropy loss function instantiated through the Torch module. Optimization was done with a predefined Adam module from Torch, with a learning rate of 0.0001. Cross entropy loss and the optimization parameters were chosen as they are typical values for this type of problem.

The network training was conducted using the NVIDIA CUDA application on an EVGA GEFORCE RTX 2080Ti graphics processor. Each network was trained for ten epochs. Each epoch took approximately 10 seconds to train. Accuracy and loss were measured on both the training set and validation set. VGG11 performed the best with 98% accuracy and a training loss of 0.06. VGG16 performed poorly with a training accuracy of 40% and a training loss of 1.1. VGG19 performed the worst with a training accuracy of 28% and training loss of 1.35. The model accuracies and losses were graphed for each model, and then accuracies were compared between models (see appendix A).

The models were tested on a variety of randomly selected images from Google Images. One of each class of bird was provided to the model and each was predicted correctly. To experiment, a fourth quail not trained on was provided to the model. A bobwhite quail, which is most like the harlequin quail, was predicted as a harlequin quail.

Post Analysis

The first question raised is: why did VGG11 perform so much better than VGG16 and VGG19? In competitions, VGG16 and VGG19 typically perform better. The VGG11 architecture contains 11 weight layers in total, with 8 convolutional layers and 3 fully connected layers, for a total of 133 million parameters being trained. For both VGG16 and VGG19, the number of parameters and total layers are increased. With such a small dataset for training, the problem of overfitting is apparent here. With more parameters and more features to look for in each image, the simple task of identifying mostly characteristically independent features between the quail becomes more cumbersome. The VGG16/19 networks are designed to handle much larger datasets and would likely perform better if we included the entire dataset with all 450 birds.

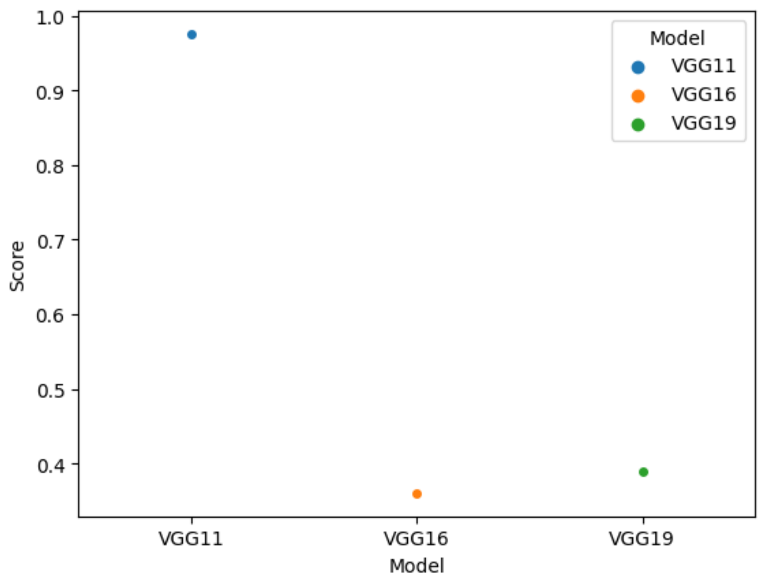
The second question is: should we tune any hyperparameters to increase our accuracies? With our VGG11 accuracy at 98% for a three-class problem, the model is already performing well. The training time could be extended for more epochs, further solidifying the accuracies. It is common to train for 100+ epochs. Furthermore, a loop to find an optimal learning rate could be useful.

Conclusion

For a simple three quail classification task, VGG11 with little network and parameter tuning can achieve 98% accuracy. VGG16 and VGG19 do not perform as well, but further testing on a larger dataset of more bird classes should be done to test the limits of those networks. With the main goal being to develop understanding of neural networks and image classification tasks, this project begins the journey of diving deeper into more complicated modeling.

Appendix A

1. Three classes of quail trained during this project



2. Model accuracies compared

Works Cited

Dataset:

https://www.kaggle.com/datasets/gpiosenka/100-bird-species

Models:

https://pytorch.org/vision/stable/models.html

Imagenet:

https://www.image-net.org/