Applications of FastAI Pretrained Models in Image Classification Problem

Author: Antonio Tapia Maldonado
Advisor:Dr. Jeffrey Duffany
Computer Science Department

Abstract

Since the inception of Transformers and GPTs, Artificial Intelligence has proliferated. Fastai is among the cutting-edge libraries that are leading new advancements in the field. To harness the power of FastAI and other advancements in the field, we set out to try and evaluate the practicality of the FastAI library.

We choose a given use case for Artificial Intelligence and then set out to fulfill said use case by leveraging Fastai. We created three image classification models through Fastai and then made an application that used those models. The use case we chose was a local wildlife fauna and flora classifier. The results from training the models were models with meager error rates, and these models had little to no data engineering.

Introduction

One of the most cutting-edge libraries in Machine Learning is FastAI. This library includes very sophisticated pre-trained models and tools for fine-tuning them. It also contains the tools to leverage Graphical Processing Units(GPUs) to accelerate the fine-tuning and inference processes. Thus, our purpose for this project was to develop a system that included Deep Learning models to classify flora and fauna in Puerto Rico. We leveraged FastAi and Jupyter notebooks to train various deep-learning models. We also developed a React Native based app to test the models with custom data.

Background

Fastai is a deep learning library which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches.[2].

One of the most cutting-edge libraries in Machine Learning is FastAI. This library includes very sophisticated pre-trained models and tools for fine-tuning them. It also contains the tools to leverage Graphical Processing Units(GPUs) to accelerate the fine-tuning and inference processes.

To our interest, the Fastai library also has a set of tools to facilitate the implementation of deep-learning neural networks. These deep learning networks come bundled with all the knowledge needed to be fine-tuned into the problem because these networks have already been trained with state-of-the-art hardware. So, they already have a lot of knowledge embedded in them. Note that deep neural networks are defined as those with three or more hidden layers.

Problem

This project was to develop a system that included Deep Learning models to classify flora and fauna in Puerto Rico. We leveraged FastAi and Jupyter notebooks to train various deep-learning models. We also developed a React Native application to facilitate testing the model's inferences process.

Methodology

After a preliminary evaluation of all the pre-trained models, we chose the resnet50 pre-trained model, which provided the lowest error percentage for a limited test of three epochs and a preliminary image size of 150x150 on a tree classification dataset.

Intending to improve the accuracy output of the AI models, we decided to create three separate models, instead of training one model to classify between all categories. We also limited the species' categories in each model's output to nine, which should have sufficient scope.

To obtain a set of labeled images, we utilized the Bing Image Search API. These images are what the image model will use to learn how to classify the different images. To feed the images into the model, all images were resized to fit a square of 250px by 250px

To obtain a training and testing set we utilized the Bing Image Search API to download hundreds of labeled images for each species we pick to include as a category. Our objective was for the model to learn to discern the most common features of each species type. We removed some of the labeled images we received from Bing Image Search. Specifically, we removed images with the following characteristics:

1. Two or more species in the same image
2. Collages
3. Sprouts or undeveloped specimens: we are only interested in fully-grown specimens
4. All drawings and illustrations
5. Images with people or other things not related to the species
6. Non jpg format images
7. Non full color images

Our application of image classifications helps tourists identify trees, fungi, and animals in the rainforest trees in PR. To do so, a set of labeled images must be gathered with the most common species for each category on the island. We then utilized the vision_learner class from FastAI to which utilizes pretrained models, to learn how to classify between the species of each category.

Torchserve is a highly scalable and efficient web server specialized in serving inferences from exported machine-learning models. It functions as an easy way to serve inferences from the web. The model was deployed to a virtual machine on the Microsoft Azure Cloud.

Torchserve receives packaged files as input in the Mar file format. To generate the Mar files, we utilized the program "torch-model-archiver," which received the weights file from training the model, a handler for torchserve, and finally, a model class that contained the architecture for reconstructing the model. We utilized this program to generate the mar files that torchserve uses.

The torchserve server automatically resizes all images it receives to have the dimensions of 250 px by 250 px and then makes an inference on the image. This is because we configured the mar file handler to resize the photos and how to send back the inferences.

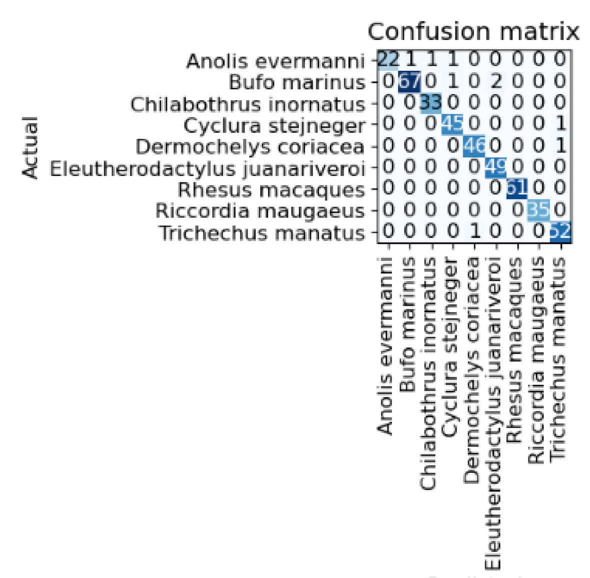
To facilitate the process of submitting images for classification, a React Native client app was developed. It consisted of a simple client holding the inferences downloaded from the torchserve server.

Results and Discussion

One applications we developed as part of our project of image classifications is helping tourists identify animals native to PR. We picked nine of the most iconic animals on the island.[3]

Namely, we picked the following animals by their scientific names: 'Anolis evermanni', 'Bufo marinus', 'Chilabothrus inornatus', 'Cyclura stejneger', 'Dermochelys coriacea', 'Eleutherodactylus juanariveroi', 'Rhesus macaques', 'Riccordia maugaeus,' 'Trichechus manatus.'

After cleaning the dataset, we utilized a randomly generated image split to train and then tested the training results. From the training, we obtained a model with the following characteristics:



epoch	train_loss	valid_loss	error_rate	time
0	0.888312	0.115969	0.028640	03:41

epoch	train_loss	valid_loss	error_rate	time
0	0.180658	0.091292	0.023866	04:55
1	0.144018	0.096881	0.026253	04:27
2	0.135816	0.101307	0.026253	04:38
3	0.135237	0.184823	0.042959	04:38
4	0.116133	0.140800	0.038186	04:48
5	0.101545	0.145114	0.040573	04:25
6	0.074477	0.108284	0.021480	04:21
7	0.057583	0.098347	0.026253	04:17
8	0.042225	0.096191	0.023866	04:16
9	0.036369	0.096404	0.021480	09:26

Figure 1: Confusion matrix and results of training an AI on the selected dataset

From the previous model, we submitted the image below to see if it would be classified correctly:

Figure 2: Image submitted to the models and the resulting inference [1]

The models, as was shown by the confusion matrixes, did not have a one hundred percent accuracy in their predictions. So, we cannot say that the models learned how to classify all characteristics of a given species, as is shown in Figure 3, which is an erroneous prediction.

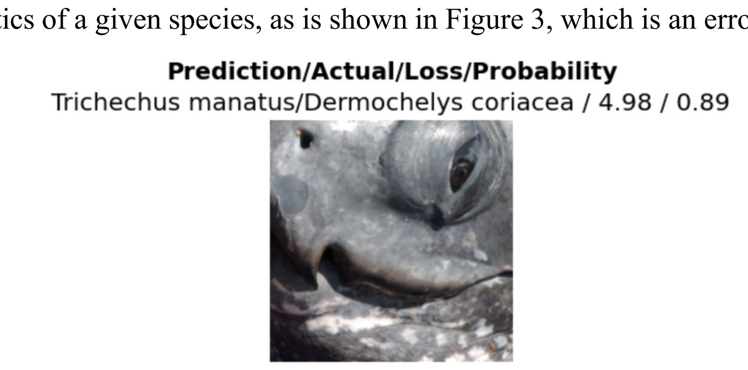


Figure 3: Example of incorrect inference and probability the inference model thinks it is part of the incorrect species

As shown in Figure 15, we can also say that the models struggle to classify species from a distance. It also needs help when they are not the image's central focus, as shown in Figure 4.

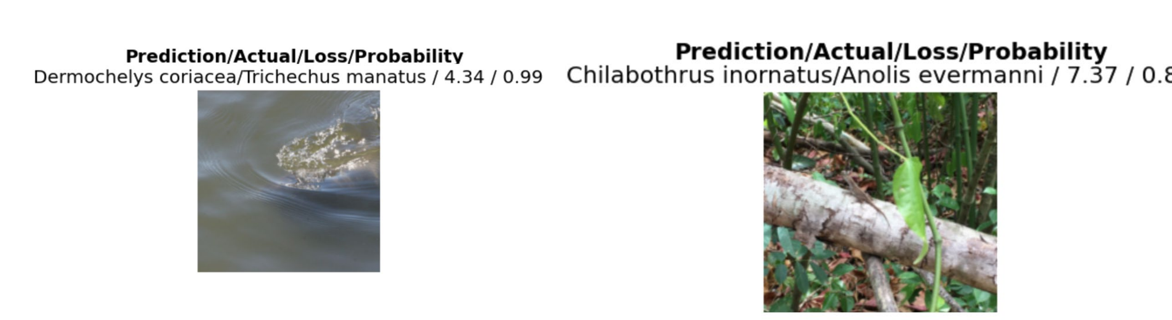


Figure 4: other examples of incorrect inferences

The models also struggle with images of the animals hidden in their natural habitat, as shown in Figure 4, where the manati is hidden underwater. We note that these are severe problems with the application, and a larger dataset is needed to evaluate the shortcomings of the models trained with FastAI. But that's beyond the scope of this paper with our current dataset. From the results, we can say the AI learned enough to classify clear images of each species similar to the photos in our dataset.

Conclusions

From the model standpoint, we got a correct prediction from each of the three models on both images we obtained to infer. This tells us that fastai and pre-trained models are comparable to models tailors just for the problem. Which would require much more time and data engineering to make.

With Torchserve, we succeeded in deploying our three models into a highly efficient inference-serving web server. We also successfully managed to generate a MAR file for each of our models.

The React native app allows us to successfully upload and classify images and store the results. The app is ideal for submitting a massive amount of pictures into an inference model.

The impact of fastai and other pretrained libraries like huggingface should be significant. Pretrained models require little to no expertise in machine learning, making it almost trivial to train a model. With tools like torchserve to serve our models, integrating machine learning into our applications is now easier. Image classification can be done with little to no data engineering.

Future Work

Future work might include the expansion of the dataset used to train the model. This could be done via the accessing of a labeled image database or through the use of a specialist to generate such a database.

To further increase the validity of the conclusion a model may be made from scratch via data engineering. This would include the use of pytorch, numpy and pandas. Plus implementing of the given algorithm to that is part of the model.

Another possible road would be the field test of the react native application to see if the whole system is practical: accurate, fast enough or neither. This could be done in a heavily forested area like el Yunque.

Acknowledgements

I would like to thank my advisor Jeffrey Duffany for all the guidance he gave during the research period.

References

1. National Invasive Species Information Center. (n.d.). Cane Toad | National Invasive Species Information Center. Www.invasivespeciesinfo.gov. https://www.invasivespeciesinfo.gov/aquatic/fish-and-other-vertebrates/cane-toad
2. fastai. (2019). Fast.ai. https://docs.fast.ai/
3. 9 Animals You Can See in El Yunque National Forest. (n.d.). PR Activities. https://puertoricoactivities.com/blog/el-yunque-national-forest-animals/