Alexander Noah King-Bailey

Introduction to Machine Learning (DAT2025)

Adam Forland

Bootstrap Neural Net Reflection

May 9, 2023

In this machine learning project, we utilized convolutional neural networks to analyze a series of trail cam footage and identify different objects, such as deer, humans, rabbits, and coyotes. The main objective was to create a model that could accurately and efficiently identify these objects in the footage.

The most used tool we used for this project was TensorFlow. Used for a variety of functions from word embeddings to symbol classification, we used it to import the sequential model mechanic so the neural network could process the data in the order of which the layers are programmed. TensorFlow provided the activation function and the ability to flatten the processed trail cam footage for final interpretation by the programmers as well as a way for the neural net to recognize two-dimensional imagery in a convolutional architecture called Conv2D. We also deployed a subsidiary of TensorFlow called Keras. Keras is an API that makes other TensorFlow tools compatible with the python coding language. At the same time, it simplifies tool functions and outputs for increased understanding amongst human users.

One of the key contributions I made to the project was commenting on how the neural net worked. However, I did not make comments in the traditional sense due to the sheer amount of content for each code function, coupled with a lack of reasonable time. Instead, I summarized each section of the project in which code was deployed to fulfill a specific task. I covered what the code did, how it interacted with other sections, and how it fulfilled the overall purpose. This was important because it helped the team understand the mechanics of the model and how it was making its predictions. Should any member of the team wish to reference the project for later endeavors or present it to a non-technical audience, the comments will provide an advantageous element of understanding and credibility. Additionally, I provided a proportion of the imagery to feed into the neural net, which was crucial to the training and validation process.

We began the project by collecting a large dataset of footage, a significant portion of which containing the objects we were interested in identifying. The footage came from a series of trail cams deployed along different locations on the Red Rocks campus to capture any elements not previously apart of the frames. We used a python function built by two of our technicians called 'Sort Master' to go through a folder's worth of all the trail cam photos and identified any objects that stood out. We selected an assigned folder containing the photographs from the cameras, and using a grid location tool to point out and label specific entities such as rabbits, deer, squirrels, and people. All the photos we touched were consolidated into CSV images and turned into a single dataset. We then split the dataset into training, validation, and test sets, ensuring that the proportion of each object was balanced across the sets. After that, a neural network with convolutional architecture was used to train the model to identify the objects in the footage.

The neural network we built would operate in a sequential manner. In other words, each layer will process and interpret the data in the order of which they were written. Each set of layers used relu as their activation function and had a set kernel size that was both friendly to the storage of the server but also high enough in resolution that objects in the data images can be recognized by the neural network to create its label conclusions. Layers were also given a specific number of visual filters at which the neural network would view the trail cam footage through to detect the objects we wanted labeled. And with each progression of the neural net we added more filters by a factor of 64.

Once the image data was processed by the neural net, we displayed it's outputs and parameters. The outputs we saw were inclinations of how large and how wide the processed images were following processing. The unit of measurement was in pixels. In addition, the final summary displayed how many filters were applied to the data with each layer. From there, the total amount of parameters in the neural net, alongside how many were and were not trainable were displayed. For the sake of this project, the parameters numbered up to ninety million and all of them were trainable.

After we trained the data, we imported weights from a callback file so the neural network would be capable of enhancing its education from previous sessions, it ran and processed the data. From there, we built a test data function to properly evaluate the performance of the neural net's classifications. We used a set of trail cam images touched by the team but not by the neural net and set its batch size to 32. This set would also be used as the target with which the neural net would set as its objective to reach. Specific object categories the test function would be required to label would be labeled 0-5 and corresponding with the most regularly occurring identifiable objects in the footage. The test function would then add any of it's correctly predicted classifications to an array and keep out the ones that were inaccurate.

In the end, we were able to develop a model that could accurately identify the objects we were interested in, with an accuracy rate of over 90%. This was a significant achievement, as it demonstrated the potential of using machine learning to analyze and interpret wildlife data.