David Lappin
Capstone Report – BrainStation
4/09/2023

Project Statement/Background Subject Matter:

For this project the aim is to develop a strong understanding of Convolutional Neural Networks (CNN) and their use in image classification problems. Coming from a background in marine and environmental sciences, I thought it would be interesting to form a project around species classification. By using a diverse dataset featuring a variety of species we can showcase the strength of CNNs and outline potential use-cases for classification models in these fields.

Potential use cases that can be explored from species classification include but are not limited to:

- Species identification on trail camera monitors for state and personal population/habitat monitoring
  - The use of trail cameras is extremely common by state and private entities when evaluating population dynamics of different species for hunting or general research purposes
  - Can we automatically classify species based on trail camera footage and add value to our population estimates
- Mobile applications for species identification while outdoors
  - Using collected images can we tell the user what species it is?
  - Could we also provide additional information on that species including facts, potential hazards, habitats, endangered status ect…

Dataset:

The data set features 20 unique species classes that are found in Oregon. The data was web-scraped from google images in 2019. Each species was stored in a specific subfolder within the main directory. The data set contains ~14,000 images.

The data for the project can be found here: https://www.kaggle.com/datasets/virtualdvid/oregon-wildlife

Cleaning and Preprocessing:

The cleaning and preprocessing of the data was relatively straight forward. The main issue faced was identifying and removing incompatible files. Since the data set was web-scrapped, the images contained multiple file extensions and varying dimensionalities. In our compatibility testing we received some errors indicating that the sizing was incompatible with the model. To solve this we step-wise removed file extension types until we sourced the problem. In the end we only had to remove less than 0.05% of the data in order for it to be compatible.

The last step was to split our data into new directories. From our original main directory we split the data into our Training, Validation, and Testing Directories (Figure 1). In this way, we could ensure the data for each of these purposes was stored separately. After cleaning and separating the data we checked to ensure the classes remained balanced and found that the classes were, for the most part, well balanced (Figure 2).

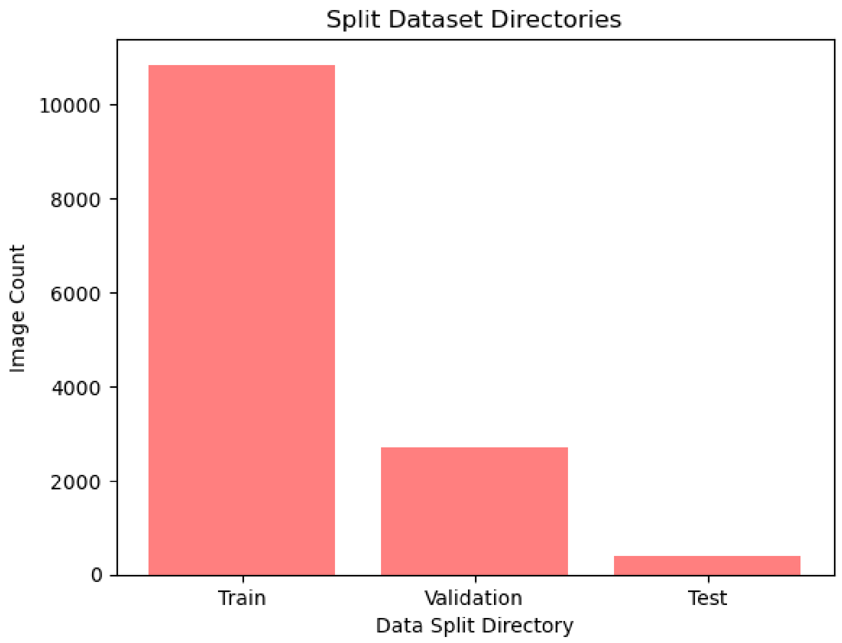


Figure 1 – The data was split into three directories for training (10850 images), validation (2700 images), and testing (400 images)

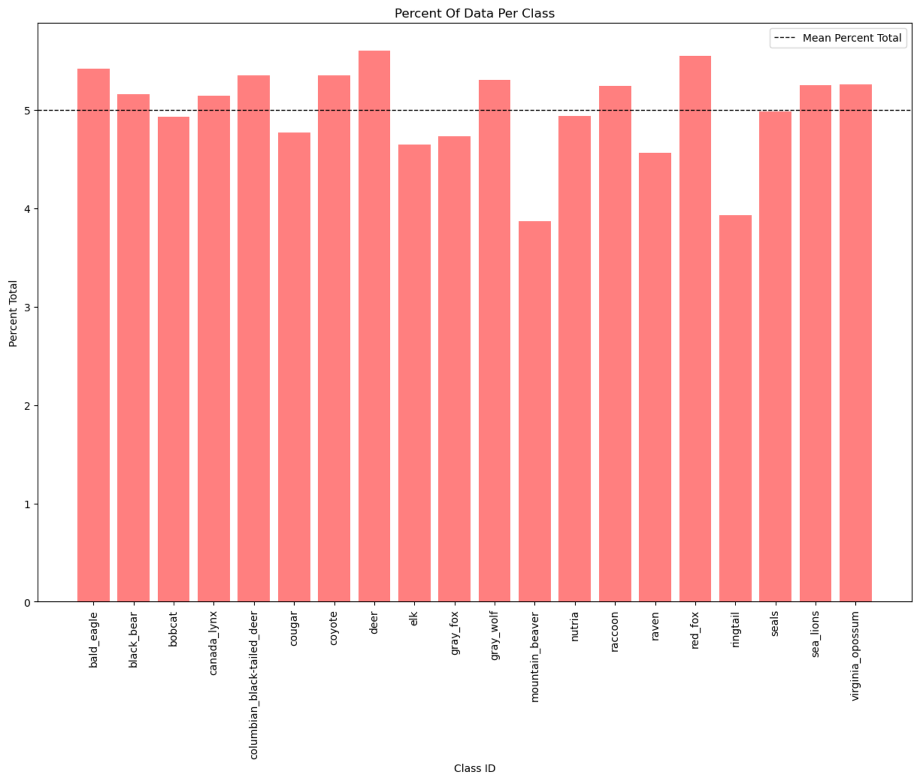


Figure 2 – Looking at the class distribution shows that most classes are well balanced. 'Mountain_beaver' and 'ringtail' are slightly under-represented

Modeling and Results:

For this project, modelling was conducted in two main steps. Firstly, we created a CNN from scratch and adjusted the model over multiple iterations. Secondly, we implemented transfer learning with the MobileNetV2 pretrained model. All models were created in Keras and Tensorflow. Note that the following models discussed can be referenced in the Model Performance Summary (Figure 3).

CNN from Scratch:

Baseline Model:

Our first step to creating a viable CNN model was to create a simple baseline model to improve on. Interestingly, our baseline model performed suspiciously well and it prompted us to take a second look at our data. Based on the generally high performance of the baseline model, we concluded that there was likely to be an unforeseen data leakage problem. Our conclusion was that since the data was web-scrapped from google images, it was entirely possible that duplicate images were present in the data-set. Unfortunately, due to the naming regime and the directory structure, there was no way other than manually sifting through the data to identify and remove duplicates.

Second Model:

In order to continue working with the data set, and to reduce the effect of the data leakage we decided to implement data augmentation in our subsequent model. The data augmentation served to reduce over fitting, but also helped to increase noise in the training, thus reducing the effect of the data leakage across directories. We also included additional normalization and regularization techniques such as drop out and batch normalization in our following model to alleviate some of the overfitting we observed in our baseline.

In our second model we observed a lower overall validation accuracy, indicating that the data augmentation was decreasing the effect of the data leakage. So, while it may seem like a step backward in performance, this was a good sign that we were addressing some of the issues with our data set.

Third Model:

In our third and final model from scratch we chose to increase the complexity of the model architecture to try and improve the model performance. Through these efforts we were able to increase the models accuracy by roughly 5-8% (dependent on the optimal epoch range selected) and the models weighted f1-score (a calculated metric used to compare across models on a scale of 0-1, with 1 being the best) by 0.06. While these changes may appear small, it showed that we were able to improve our model within the limit of our computational power.

Transfer Learning:

Our last model implemented transfer learning to utilize the pretrained model ‘MobileNetV2’. This model was trained on millions of images within the ImageNet data set and is a powerful and computationally

efficient model used for classification problems. By using this model, and fine tuning it to our specific data set we were able to achieve nearly double the accuracy and f1-scores from our scratch model. While it was a challenging experiment to build and improve a scratch model, implementing transfer learning is more closely aligned with the industry standard. Our results show its power.

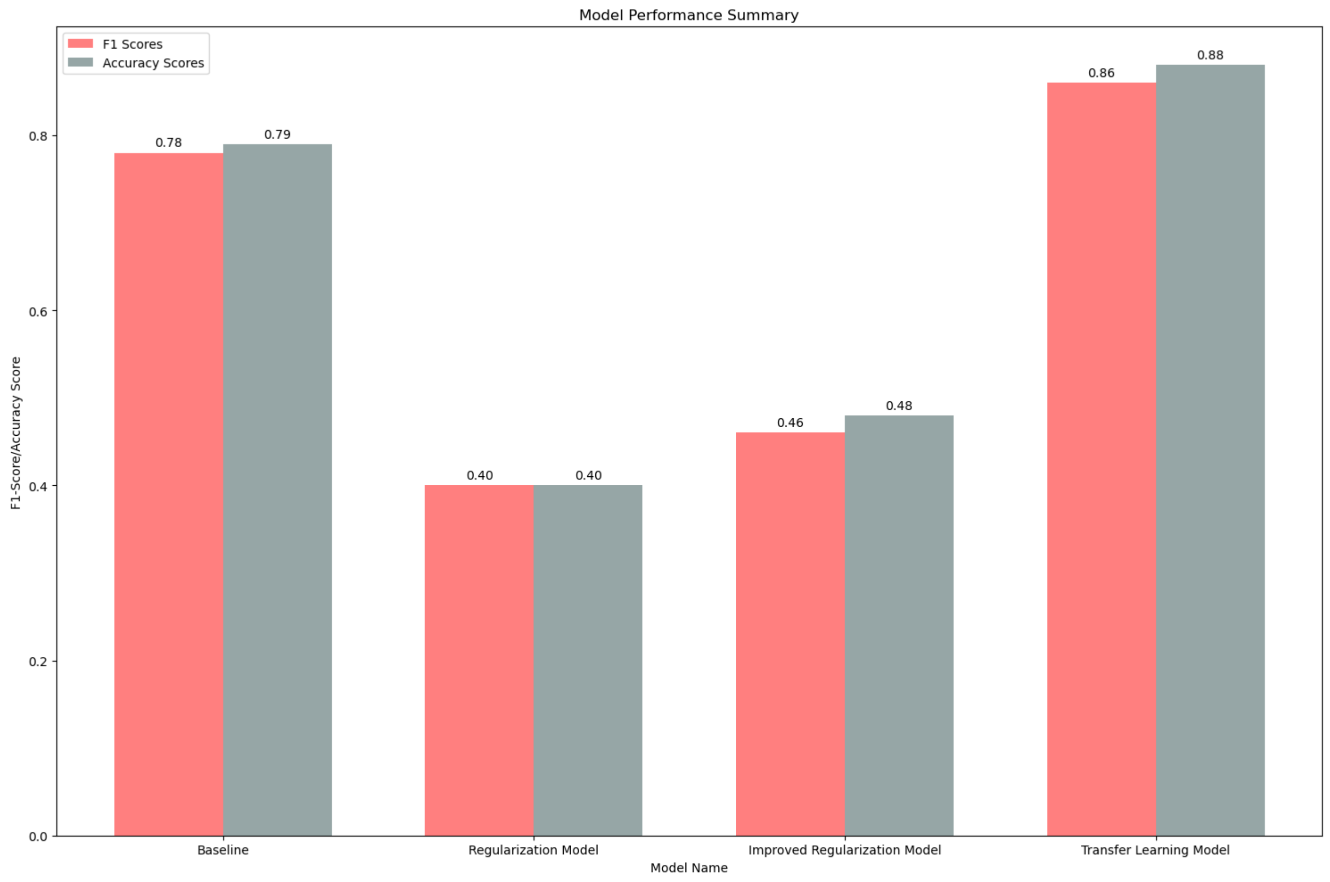
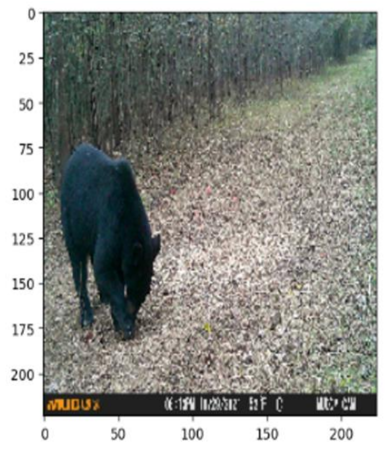


Figure 3 – Shows the various model performances throughout our testing. Note the baseline is disproportionately high performing due to suspected data leakage as explained in the report. The remaining models show improvement to both the accuracies and the weighted average f1-scores through the workflow.

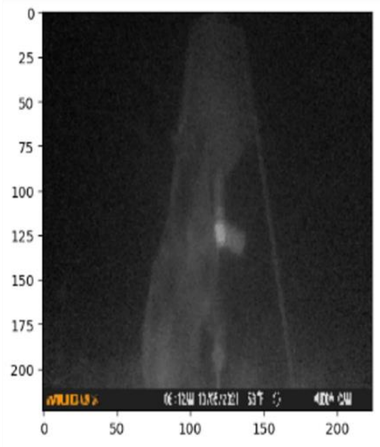
<u>**Testing a Potential Use Case with Trail Camera Footage:**</u>

Thanks to a friend I was able to secure a very small data set of trail camera footage from his farm. The data set included a variety of images of deer and some black bears at a feeder. The majority of these images were taken at night. As such, they provided an interesting challenge to the model which was primarily trained on image data taken in the light. The model was able to correctly extract the features of most of the images and provide us with accurate predictions. This was a promising start to further developing the project going forward. Some examples of the predictions are below:

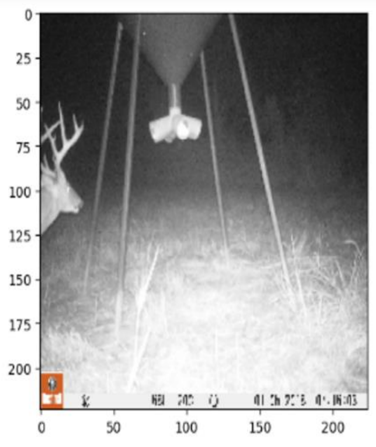
David Lappin
Capstone Report – BrainStation
4/09/2023



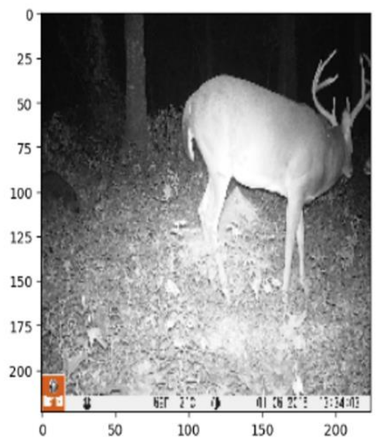
```
1/1 [==============================] - 0s 33ms/step
This image most likely belongs to black_bear with a 12.44% confidence.
```



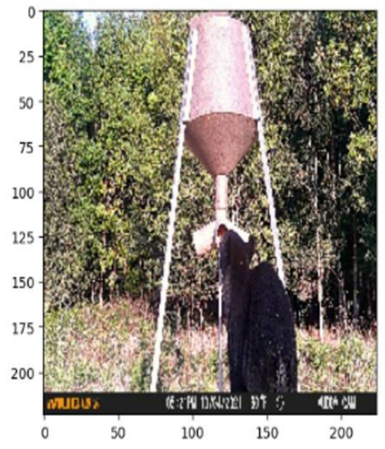
```
1/1 [==============================] - 0s 31ms/step
This image most likely belongs to raven with a 11.46% confidence.
```



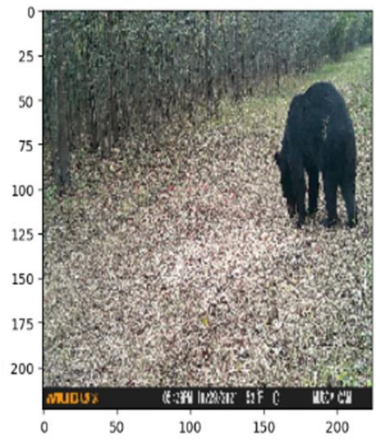
```
1/1 [==============================] - 0s 31ms/step
This image most likely belongs to columbian_black-tailed_deer with a 7.17% confidence.
```



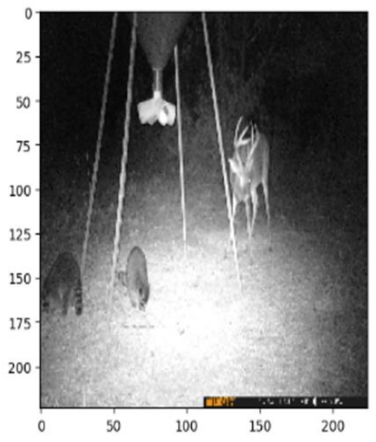
```
1/1 [==============================] - 0s 31ms/step
This image most likely belongs to deer with a 7.24% confidence.
```



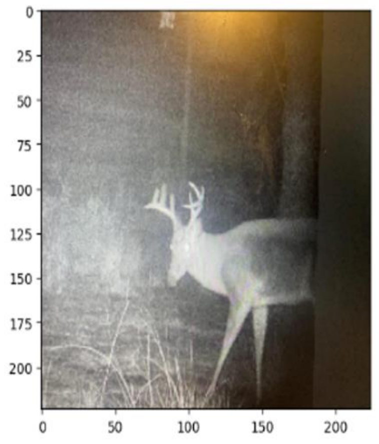
```
1/1 [==============================] - 0s 32ms/step
This image most likely belongs to black_bear with a 11.98% confidence.
```



```
1/1 [==============================] - 0s 32ms/step
This image most likely belongs to black_bear with a 12.32% confidence.
```



```
1/1 [==============================] - 0s 31ms/step
This image most likely belongs to columbian_black-tailed_deer with a 6.51% confidence.
```



```
1/1 [==============================] - 0s 31ms/step
This image most likely belongs to deer with a 12.50% confidence.
```

David Lappin
Capstone Report – BrainStation
4/09/2023

Findings and Conclusions:

Our findings and conclusions can be summarized below:

- From our EDA and Baseline modeling we uncovered that the data-set had some data leakage issue sin the form of duplicate images. We were unable to manually remove these, and so we used data augmentation to alleviate the issue.
- We were able to build and improve on our scratch CNN model through multiple iterations of increased complexity and with the addition of regularization/normalization strategies.
- We used transfer learning with the MobileNetV2 model to create our most powerful classification model. We concluded that this method, as the industry standard, was the most powerful and accurate strategy.
- We tested our new model on some unseen data and showed that it was capable of predicting species in trail cam footage

For our next steps we would:

- Work to fine tune the model with more specific trail camera footage
- Deploy the model to an application to allow for user interaction with personal/new images