Final Report

<u>Study Project: The Camera Trap Challenge</u>

Computer Vision & Machine Learning Algorithms to Analyse Remote Sensing Camera Trap Data

<u>Group 07</u>
Mohamed Shamsudeen (536944)
shamsudeen.m@uni-muenster.de

Introduction:

The study of animal populations and behavior has greatly benefited by the use of remote sensing camera traps. These cameras record pictures or videos of animals in their natural environments, giving researchers a non-intrusive tool to track and examine wildlife populations. However, interpreting the enormous volumes of data produced by camera traps can be a daunting task. Computer vision and machine learning methods are useful in this situation. The combination of remote sensing camera traps and computer vision and machine learning algorithms has the potential to revolutionize the way we study and conserve wildlife.

In this project, we have used several machine learning techniques to locate the animals in the images and to classify the animals in the images.

Aim:

The aim of our project is to develop an automated animal identification system using computer vision techniques. We aim to achieve the following objectives:

- Identify coherent sequences of images: To analyze a set of images captured over a certain period of time and identify the coherent sequences of images that are related to each other in terms of time and location.
- Locate the animals in the images: Use object detection algorithms to accurately locate the animals in the images. This will involve identifying the position and size of the animals in the image and drawing bounding boxes around them.
- Classifying the animals in the images: Use machine learning algorithms to classify the animals in the images into different categories such as badger, deer, fox, sheep etc. This will

involve training the system on a large dataset of labeled images to recognize different animal species accurately.

Methods:

Iteration 1

Identify coherent sequences of images (over time)

The process of identifying coherent image sequences involves grouping images taken within a certain time interval. In this code, a function called extract_sequences was created to group images of badgers taken within 10 minutes of each other. The function sorts the images by their timestamps and groups them into sequences based on their time intervals. The resulting sequences of badger images are then printed.

Localization:

For the 1st Iteration, we used the MegaDetector tool to locate animals in images. It is a well-known, prominent Microsoft object detection tool that is employed by several animal-analytics companies. Even pre-trained using data from the Animal CameraTrap, its most recent version, "MDv5". MegaDetector YOLOv5 object detection model trained on wildlife images. When applied to an image, the model analyzes it and outputs a set of bounding boxes that represent the locations of any animals detected in the image.

The bounding boxes are often provided in JSON format, making them simple for analysis and manipulation. Four coordinates are used to identify each box's corners in the image, and each box also has a confidence score that shows how likely it is that an animal is within.

By analyzing the set of bounding boxes returned by the MegaDetector model, it is possible to determine the locations of animals in the image and extract information about their size, position, and other characteristics.

Classification:

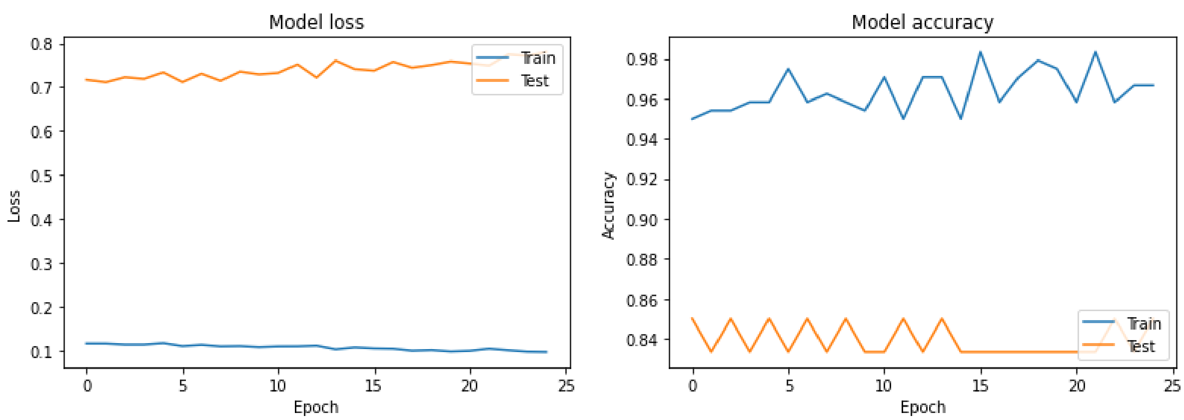
We preprocessed the image dataset by resizing and converting the images to a consistent size and format. We split the dataset into training and test sets and performed data scaling. We used a pre-trained Inception v3 model as a feature extractor and added a dense layer to create a Keras sequential model. We compiled the model using the Adam optimizer and Sparse Categorical Cross Entropy loss function. Furthermore, we trained the model for 25 epochs and evaluated its performance on the test dataset using the evaluate() method in Keras.

	Loss	**Accuracy**
Test	0.780	0.850
Train	0.093	0.962

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 2048)              21802784

 dense (Dense)               (None, 2)                 4098

=================================================================
Total params: 21,806,882
Trainable params: 4,098
Non-trainable params: 21,802,784
_________________________________________________________________
```



Subsequently, we printed the training loss and accuracy values, which indicates that the model is performing well on both the training data and test data. Following we plotted the loss and accuracy using matplotlib library. The training is plotted in blue and the test is plotted in orange. The plot can help us to visualize whether the model is overfitting or underfitting. Finally, to verify the accuracy of our classification model. It precisely gave the correct output.

Iteration 2

Extension

For the second iteration of our next phase, we have expanded our dataset by adding a total of 9 animal classes, which includes day and night images. The animal classes, along with their respective dataset sizes, are presented in the table below.

	Badger	Boar	Deer	Fox	Hare	Rabbit	Red Deer	Roh	Sheep
count	71	166	158	147	116	108	30	279	189

The training dataset comprises 996 images, which represents 80% of the total dataset. Additionally, there is a validation set consisting of 261 images, which represents 20% of the total dataset.

Localization

In this project, the task was to use object detection to identify and locate objects in images. We created both bounding boxes from manual annotation using label img and as well as used a mega detector to find the animals in the image.

Classification

We performed annotation and created four classes of animals, and trained on Megadetector's pretrained weights to classify animals into Badger, Deer, Fox, and Hare using YOLOv5. We also converted the bounding box from the Megadetector's JSON output to YOLO annotation format and verified the annotation using the Draw-YOLO-Box tool. Then, we trained YOLOv8 and used YOLOv8n as the pretrained weight. Finally, we performed detection on both daytime and nighttime images for all four animal classes using both YOLOv5 and YOLOv8.

Conclusion:

Based on the two iterations of our project, we made significant progress in expanding the scope of the project from two classes to nine classes. In both iterations, we used object detection models to locate animals in images, and then used classification models to classify them.

In the first iteration, we used a pre-trained Inception v3 model to classify two classes of animals, deer and badger. In the second iteration, we expanded the dataset to include nine classes of animals, and used both YOLO v5 and v8 for object detection. We also experimented with different hyperparameters and found that YOLO v5 performed better than YOLO v8.

Overall, our project demonstrates the effectiveness of combining object detection and classification models for identifying and classifying wildlife in images. Refining and improving the models used in this project can potentially lead to their application in other domains. For instance, the combination of object detection and classification models can be useful in identifying and tracking objects in surveillance footage or analyzing satellite imagery.

Future Work:

Future work for this project includes

- Fine-tuning the object detection and classification models on more extensive and diverse datasets could result in even more accurate results.
- This project currently identifies nine classes of animals, expanding the animal classes to make the models more comprehensive,
- Adapting the models for real-time video analysis to address the growing demand for real-time analysis, and exploring the application of the models in other domains beyond wildlife conservation.

IMAGES:


2014-02-25 7:37:07 AM M 6/10
10°C
fox 0.98
H2R1-03
RECONYX


2013-10-30 4:31:24 PM M 4/10
14°C
sheep 0.67
sheep 0.63
sheep 0.91
sheep 0.98
sheep 0.97
H1R1-01
RECONYX

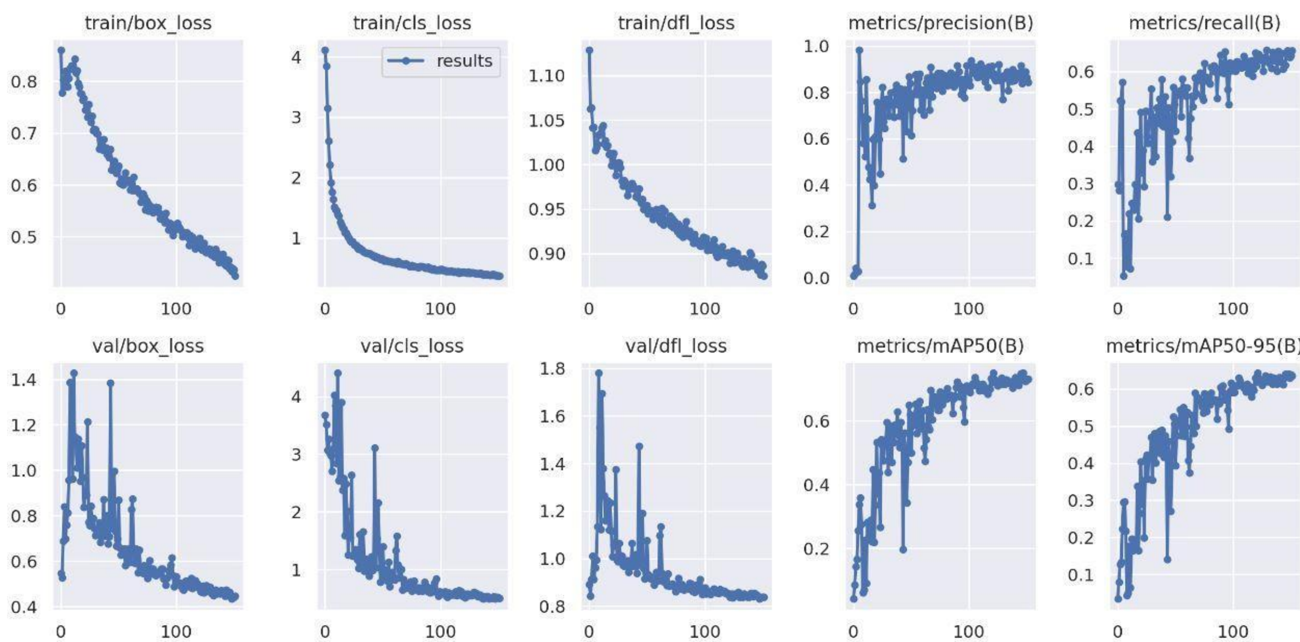

2013-10-10 7:35:15 PM M 9/10
10°C
deer 0.97
H6R1-04
RECONYX


2015-04-08 4:42:19 PM M 2/10
19°C
badger 0.96
V10
RECONYX


2013-09-28 8:38:50 PM M 9/10
12°C
roh 0.74
H4R1-02
RECONYX


2013-10-15 8:37:51 AM M 2/10
9°C
boar 0.57
H4R1-02
RECONYX

Epochs:



200 Epochs

Confusion Matrix for Yolo v8