Project

github link to megadetector:

https://github.com/agentmorris/MegaDetector/tree/main

MegaDetector is an AI model that identifies animals, people, and vehicles in camera trap images (which also makes it useful for eliminating blank images). This model is trained on several million images from a variety of ecosystems.

MegaDetector only finds animals, it doesn't identify them to species level. If you're looking for a species classifier, check out SpeciesNet, a species classifier that plays nicely with MegaDetector.

> https://github.com/google/cameratrapai

Project Scope:

1. Camera Traps are deployed in regions (in this case Margalla Hills Islamabad)
2. Use of AI to identify if animal present (MegaDetector) ()

3. Use of another AI to identify which animal present (SpeciesNet) ()
4. Output to dashboard on an application
5. Alert The authorities for whatever Reason

Goal:

> **Improve the detection and classification of animals in day and nighttime scenarios**

Location in which Snow Leopards Are present:

The snow leopard's habitat range extends across the mountainous regions of 12 countries across Asia: **Afghanistan, Bhutan, China, India, Kazakhstan, Kyrgyz Republic, Mongolia, Nepal, Pakistan, Russia, Tajikistan, and Uzbekistan**.

Given Information:

1. Serengeti LILA dataset, only outlines the dataset of animals in the savanahs of africa and dont have snow leopards in them, so i dont see how that is useful, we need a labeled camera trap images dataset of Snow Leopards

Our Goal is to make the identification of Snow Leopards better By making Megadetector(MD) better in detecting Them snow leopards, but MD only identifies animals, not their specific species, so the game plan here is to Identify:

1. Why isnt it good at detecting snow leopards in the first place (Camouflage issue)
2. How to make it good at detecting snow Leopards (Camouflage training on the model) (more data and images on snow leopards and then train the model)

Next Steps to make the model better:

1. Download LILA dataset

2. Augment it
3. pass it through MD

Questions to ask:

1. how do we fine tune the megadetector now? (state of the art)
2. hitnet model usage and how do we use it to improve megadetector
3. lila dataset has now been downloaded so what do we do about that
4. how to train hitnet using the new architecture

classification on COD, show metrics, species classification

run on COD

check on LILA dataset

Timeline of the project:

Start

1. Research papers (mention the research papers and how they helped us understand)
2. presentation ()

1. camoflague object detection,
2. primary goal (lila dataset, due to which deliverables had issue)
3. meeting with sir
4. work on cod10k dataset and also that we will try obtaining the lila dataset
5. now scope is to train the megadetector on cod10k,
6. initally checked the baseline models evalution on cod10k (metrics and graphs)

7. we have to train megadetector on cod10k,
8. training started (one repository available, agent morris didnt work, and the microsoft camera traps did, the notebook that sir gave us was old and things have changed)
9. training new methods and yolo new method so the notebook was kinda obselete
10. in the microsoft camera traps repo, the way to integrate weights wasnt mentioned, so we researched on it for a while
11. how to get weights paths
12. and it still resulted into nothing because no information was given, then we ran a train instance on the model with the dataset to check if it generates weights, when called, and it did surprisingly
13. files explored
14. util.py has one function, in which you add the model name, get model path is the function name, then after extracting the path you put that in the config,
15. then we had change the format of the dataset cod10k from coco to yolo which is the only acceptable format for the yolov6 model
16. on google collab we couldnt train for more than 20 epochs
17. dataset
18. the first train instance was to generate an instance of the weights so it didnt have the correct labels either
19. results were unsatisfactory,
20. generated proper labels, then training started and we saw a decrease in the bounding box loss
21. then we trained the model for 40 epochs which took 5 hours, and had a significant decrease in the loss
22. the loss kept decreasing, so if we have the resources we can run it for 100 epochs and then

divisions

1. test
2. train
3. val

overall the process and the pipelines is correct, we just had some logistical issues, which led us to spending a lot of time on fixing that,