Sea Turtle Conservation AI Project

Part #3

EEL3872

Jay Rosen

Abstract

This study explores the development and application of an AI-driven tool, "Cammy," designed to identify various species of sea turtles from a curated dataset. The project involved labeling a 60-image dataset of sea turtles, training the Cammy AI chatbot to recognize species distinctions based on visual characteristics, and evaluating the model's performance. Key challenges such as copyright infringement concerns and dataset biases—stemming from the professional quality and settings of the images—were addressed. The research highlights the intricacies of AI training with copyrighted material and the implications of dataset selection on model accuracy. The findings underscore the potential and limitations of AI in biodiversity conservation efforts.

Introduction

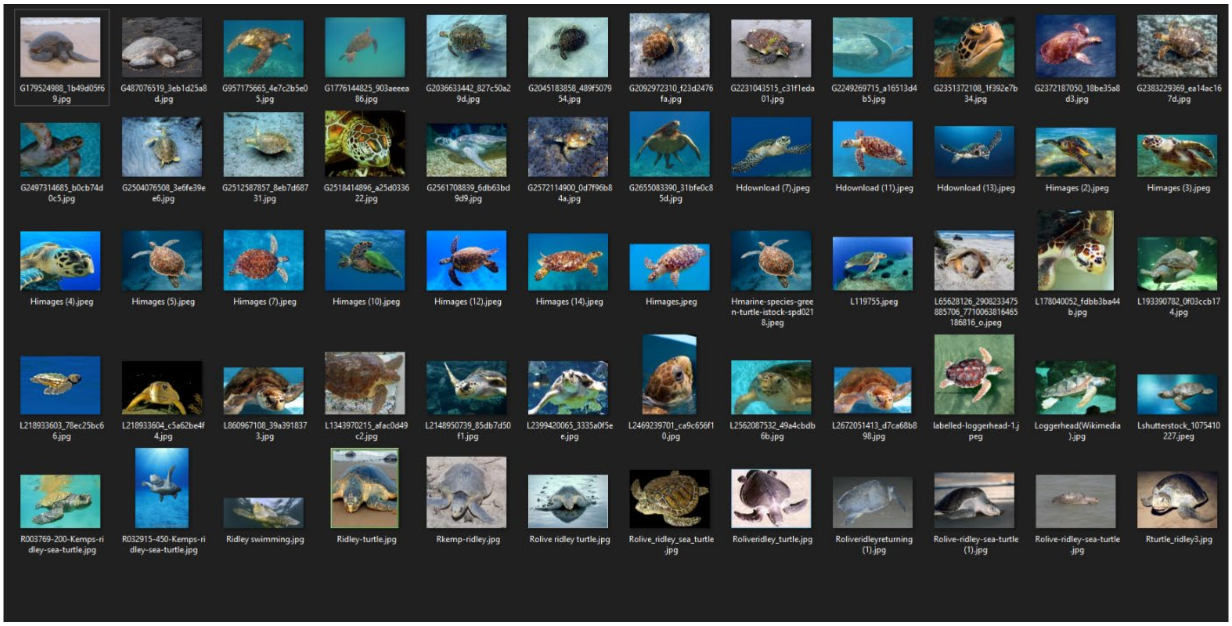
In the realm of biodiversity conservation, the accurate identification of species is paramount. Sea turtles, with their varying species and widespread habitats, present a unique challenge for conservationists. Leveraging the advancements in AI, this study introduces "Cammy," an AI chatbot tailored to discern between different species of sea turtles. The initiative stems from the urgent need to enhance species recognition capabilities to aid conservation efforts effectively. The dataset, comprising 60 images sourced from online platforms, was meticulously labeled to facilitate the AI training process. This research not only ventures into the practical application of AI in conservation but also delves into the ethical and legal ramifications of utilizing copyrighted images for AI learning purposes.

Problem Statement

The core challenge addressed by this study revolves around the accurate and ethical training of AI models for conservation tasks, specifically the identification of sea turtle species from images. The dual issues of copyright infringement and dataset biases present significant hurdles. Copyrighted images, often used without the original creators' consent, pose legal and ethical dilemmas, while the high-quality, professionally taken photographs in the dataset risk skewing the AI's learning, limiting its applicability to real-world scenarios. These concerns highlight the broader challenges faced by the AI industry in responsibly harnessing the power of machine learning for environmental conservation. The study seeks to navigate these complexities, aiming to develop a reliable, ethically sourced AI model capable of contributing to sea turtle conservation efforts.

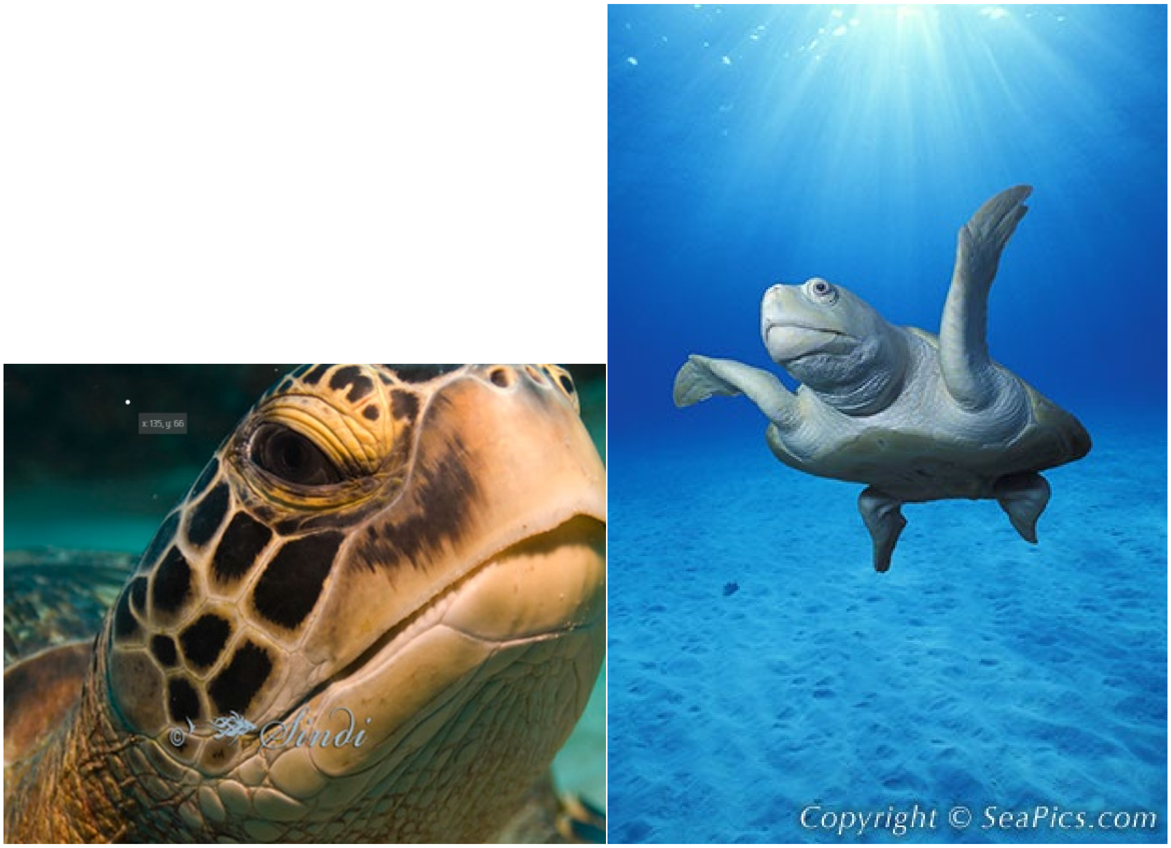
Input Data

For this experiment, we gathered a collection of 60 images of sea turtles that have been pulled from online, and trained Cammy AI chatbot to recognize the species of sea turtles in the images. The images were provided by Cog University as a zip archive, and on inspection of the images, seem to be the same images that appear in Google image search results for Sea turtles. The image file names were rather random, but the first letter of the image file name began with the species name to help us identify the turtle for labeling. The identifier letters were ‘G’ for Green Sea Turtle, ‘H’ for Hawkbill Sea Turtle, ‘L’ for Loggerhead Sea Turtle, and ‘R’ for Ridley Sea Turtle.



The 60-image dataset

It is to be noted that the images provided for this dataset show signs of copyright infringement, with watermarks like "Copyright SeaPics.com" and photographer's names clearly visible in these photos, sometimes directly over the sea turtle itself. Copyright infringement is a legal and ethical concern, but this specific AI experiment was done for scientific educational purposes only, and not for commercial applications. The watermark text will interfere with the AI Training, as the model will learn to identify the watermark as a visual feature of certain sea turtles, which is inaccurate and adds undesirable noise to the refinement process. For LLMs like ChatGPT and AI Image Generators like MidJourney or DALL-E, this is a major problem for AI industry as a whole, as these AI models are being trained on copyright images and copyright works, without the original artist or author's expressed consent. It can be difficult to identify a copyright image if it doesn't contain a watermark or metadata embedded within the image, but it is also even more difficult to gather these professional underwater photographs of Sea Turtles to begin with. This is a dilemma that AI companies, regulators, and the community will have to figure out a way to purge out copyright works from these established AI models, or way to credit or compensate original creators.



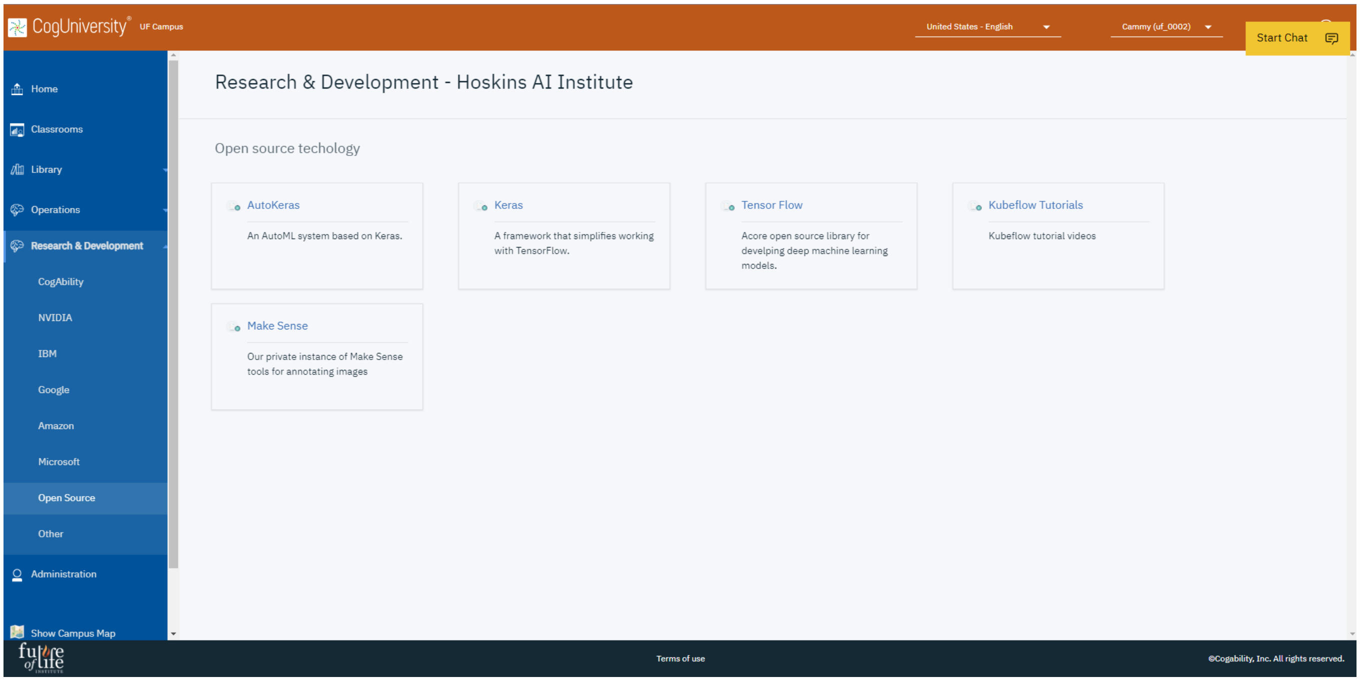
Images with copyright watermarks

Another note about these specific images is they were all taken with professional underwater cameras and gear. These photos are beautiful and taken by professional photographers who have mastered underwater wildlife photography, which means our AI trained model will only know what professionally photographed sea turtles underwater look like, and not trained on wider range of photos such as smartphone photos taken by common person at the beach or at an aquarium. Only the Ridley Sea Turtle photos appear to be shot with a less professional camera, at lower resolution, and with flash enabled, which over-exposes the sea turtle subject. Besides 2 photos of Green Sea Turtles and 1 photo of Loggerhead Sea Turtles, all the Ridley Sea Turtle photos are captured on land, while all other photos in dataset were taken underwater. This bias in the dataset does train the AI model to recognize Riddley Sea Turtles with sandy colored backgrounds instead of aqua colored ocean backgrounds. The photos in general are from variety of angles and distances which is good, but also include close ups of the turtle’s face which can skew the results as the close ups are cropped in and big. These closeups may train the AI model to think that these turtles are larger than they should be, and don't contain other features like shell or arms and legs.

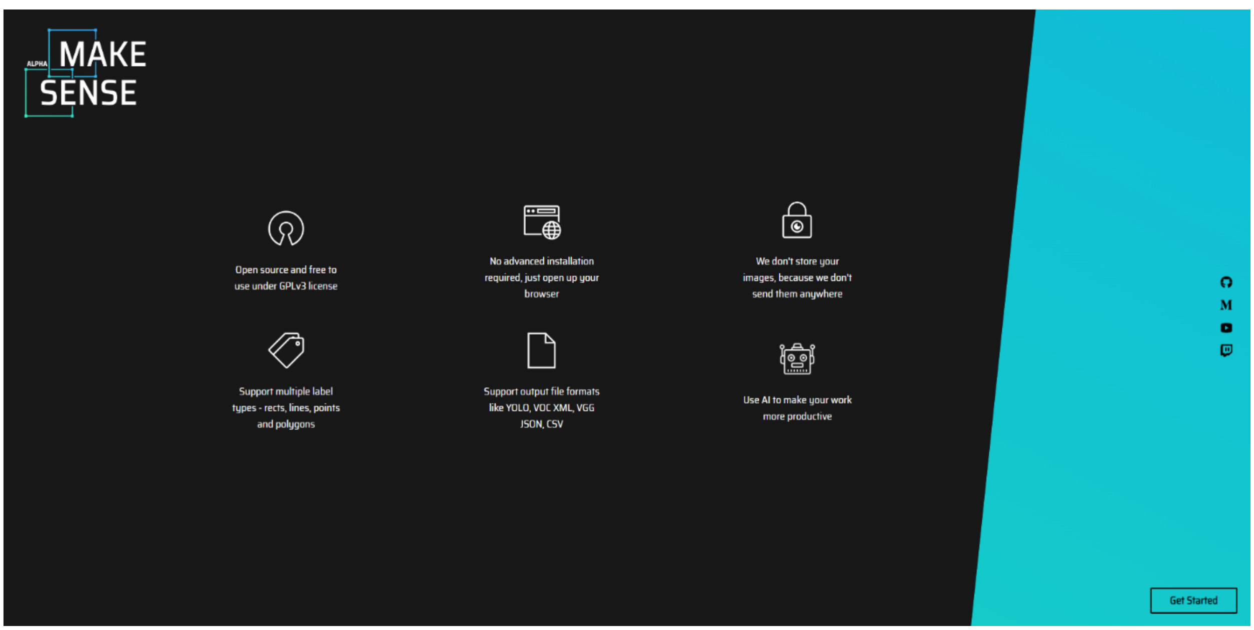
Ridley Sea Turtle photos are lower resolution and taken on land.

Image Labeling with Make Sense

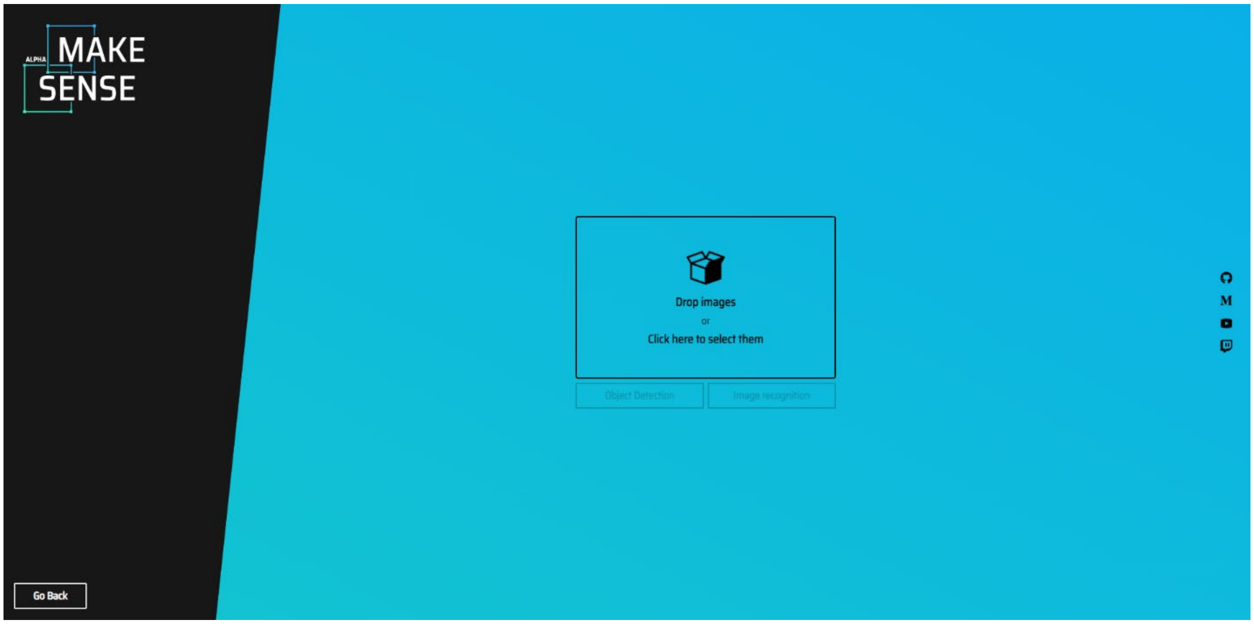
For labeling the images, we used Make Sense AI program which helped us associate words for the images. On the Make Sense AI website image upload screen, there were 2 options to choose from, Object Detection or Image Recognition. For this experiment, the image Recognition option was selected and the 60 images were uploaded. To start the labeling process, the Create Labels dialog screen prompted labels for their mages, and in this case the 4 species of Sea Turtles were each entered as a separate label. Once completed, each image was displayed in the lightbox gallery, and the right sidebar showed the 4 species as labels to associate for the image. Since the images had their filenames starting with the first letter of the species, it was relatively easy to identify the proper label of the Sea Turtle. If the filenames had not been previously named in this scheme, it would be more challenging to use our human brain to identify the Sea Turtle with knowledge of the Sea Turtle species visual features and logical judgment. Once all 60 images had been labeled, the Actions menu generated a 2 columns CSV file containing the images and labels. This CSV was uploaded to Google Looker Studio for simple table and chart analysis.



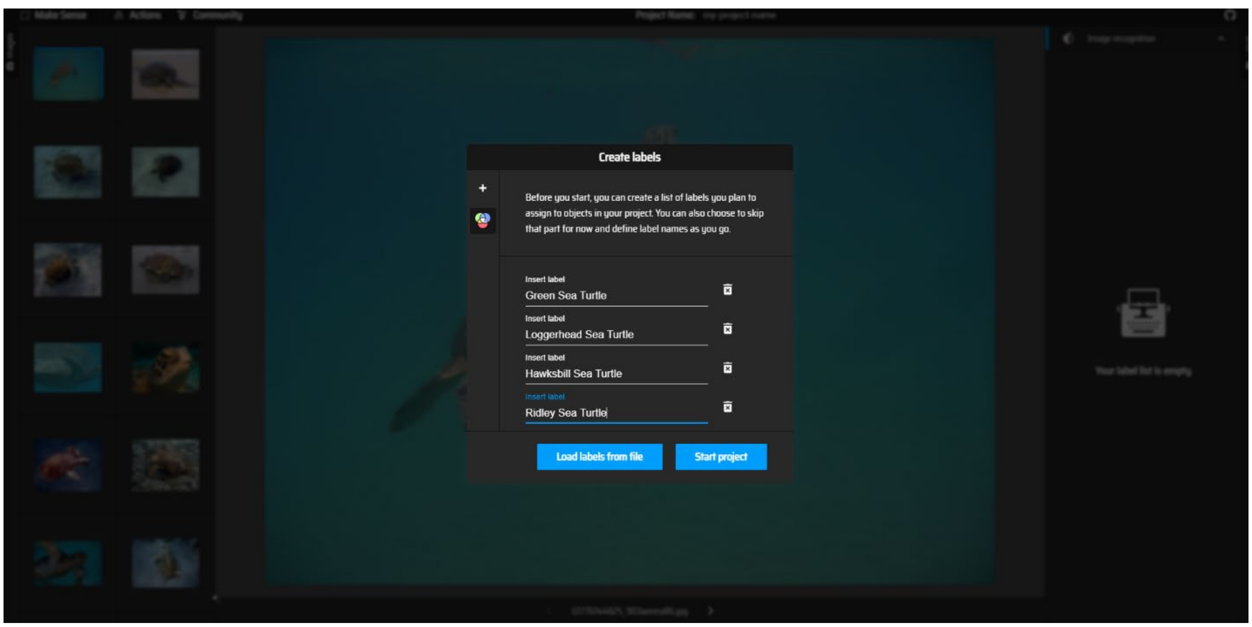
Make Sense AI was located in CogUniveristy > Research & Development > Open Source



Make Sense AI landing page.

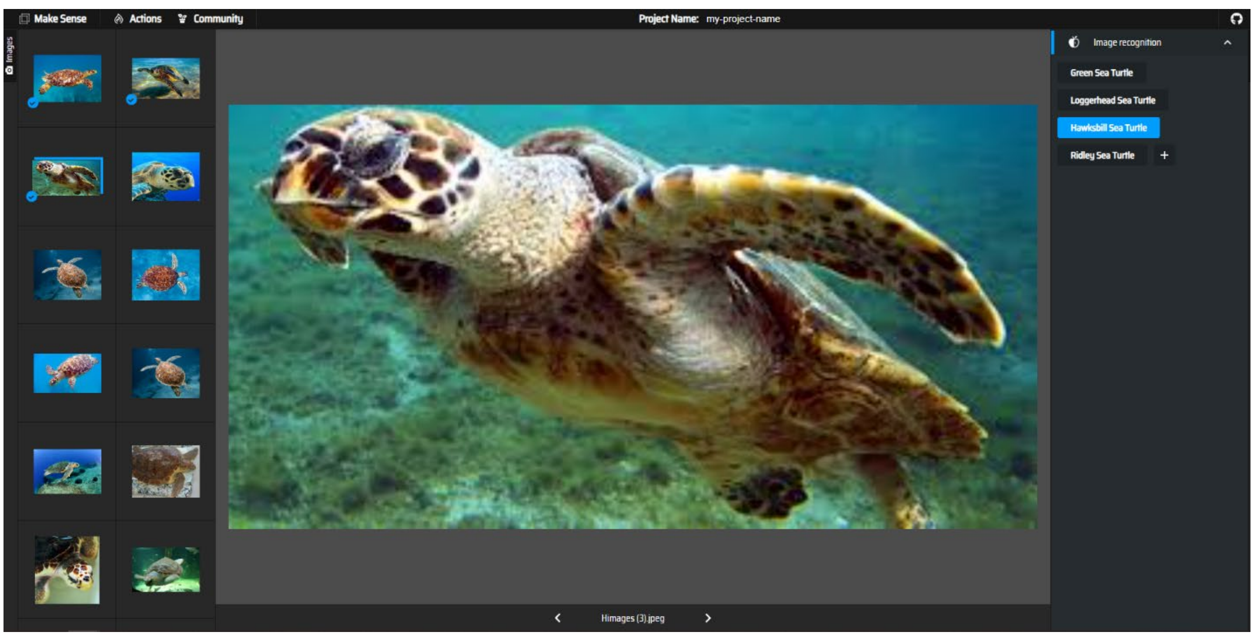


Make Sense Upload screen.



Creating the labels

Labeling a Green Sea Turtle



Labeling Hawksbill Sea Turtle

Labeling Loggerhead Sea Turtle

	A	B
1	G1776144825_903aeeea86.jpg	[Green Sea Turtle]
2	G179524988_1b49d05f69.jpg	[Green Sea Turtle]
3	G2036633442_827c50a29d.jpg	[Green Sea Turtle]
4	G2045183858_489f507954.jpg	[Green Sea Turtle]
5	G2092972310_f23d2476fa.jpg	[Green Sea Turtle]
6	G2231043515_c31f1eda01.jpg	[Green Sea Turtle]
7	G2249269715_a16513d4b5.jpg	[Green Sea Turtle]
8	G2351372108_1f392e7b34.jpg	[Green Sea Turtle]
9	G2372187050_18be35a8d3.jpg	[Green Sea Turtle]
10	G2383229369_ea14ac167d.jpg	[Green Sea Turtle]
11	G2497314685_b0cb74d0c5.jpg	[Green Sea Turtle]
12	G2504076508_3e6fe39ee6.jpg	[Green Sea Turtle]
13	G2512587857_8eb7d68731.jpg	[Green Sea Turtle]
14	G2518414896_a25d033622.jpg	[Green Sea Turtle]
15	G2561708839_6db63bd9d9.jpg	[Green Sea Turtle]
16	G2572114900_0d7f96b84a.jpg	[Green Sea Turtle]
17	G2655083390_31bfe0c85d.jpg	[Green Sea Turtle]
18	G487076519_3eb1d25a8d.jpg	[Green Sea Turtle]
19	G957175665_4e7c2b5e05.jpg	[Green Sea Turtle]
20	Hdownload (11).jpeg	[Hawksbill Sea Turtle]
21	Hdownload (13).jpeg	[Hawksbill Sea Turtle]
22	Hdownload (7).jpeg	[Hawksbill Sea Turtle]
23	Himages (10).jpeg	[Hawksbill Sea Turtle]
24	Himages (12).jpeg	[Hawksbill Sea Turtle]
25	Himages (14).jpeg	[Hawksbill Sea Turtle]
26	Himages (2).jpeg	[Hawksbill Sea Turtle]
27	Himages (3).jpeg	[Hawksbill Sea Turtle]
28	Himages (4).jpeg	[Hawksbill Sea Turtle]
29	Himages (5).jpeg	[Hawksbill Sea Turtle]
30	Himages (7).jpeg	[Hawksbill Sea Turtle]
31	Himages.jpeg	[Hawksbill Sea Turtle]
32	Hmarine-species-green-turtle-istock-spd0218.jpeg	[Hawksbill Sea Turtle]
33	L119755.jpeg	[Loggerhead Sea Turtle]
34	L1343970215_afac0d49c2.jpg	[Loggerhead Sea Turtle]
35	L178040052_fdbb3ba44b.jpg	[Loggerhead Sea Turtle]
36	L193390782_0f03ccb174.jpg	[Loggerhead Sea Turtle]
37	L2148950739_85db7d50f1.jpg	[Loggerhead Sea Turtle]
38	L218933603_78ec25bc66.jpg	[Loggerhead Sea Turtle]

labels_my-project-name_2024-03-

Ready Accessibility: Unavailable

The generated CSV of image and labels

	[Green Sea Turtle]	Record Count ▾
1.	[Green Sea Turtle]	18
2.	[Loggerhead Sea Turtle]	16
3.	[Hawksbill Sea Turtle]	13
4.	[Ridley Sea Turtle]	12

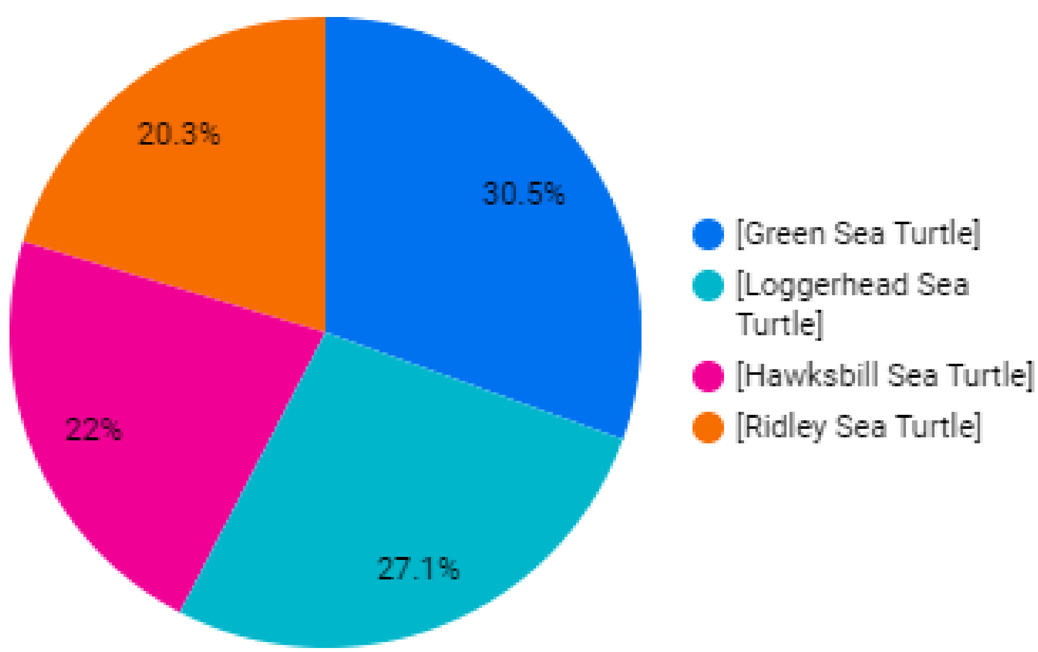
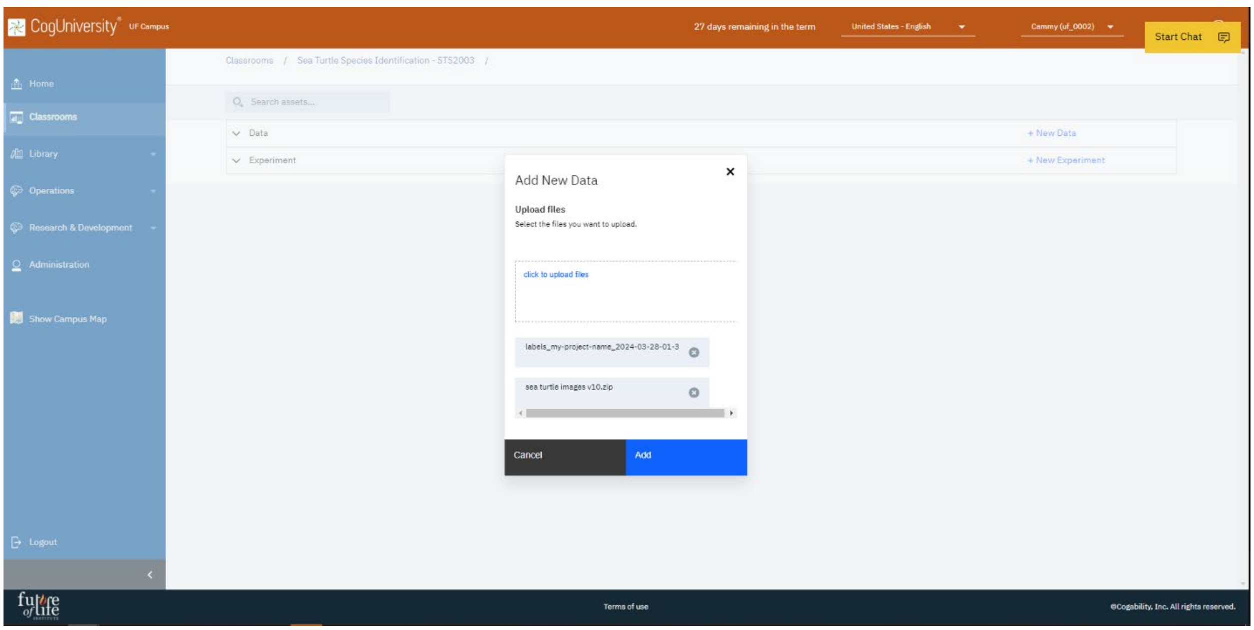
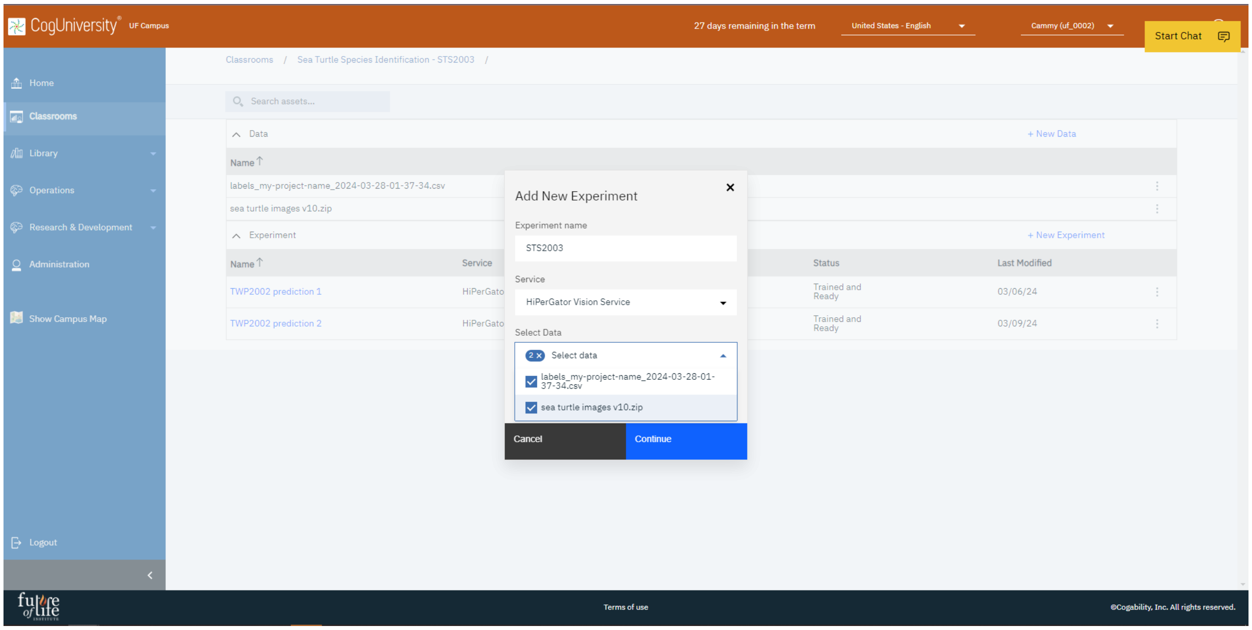


Image Recognition AI Training

Back at CogUniveirsty, the zip archive of images and CSV of labeled images were uploaded as Data for the Experiment. For creating the experiment, the service selected was HiperGator Vision Service, and both uploaded files were used as the data source. The time to train the AI model took a few minutes. There were reports from other colleagues that they had run into issue with this step, as the HiperGator Vision service was not running properly. Fortunately the AI training completed for my experiment after the bug had been fixed by CogUniveristy. After Training completed, the output were 2 text files of the HiperGatorVision service log and a summary.



Uploading the data for training



Creating the AI image recognition experiment

Submitting Experiment for training to HiperGator

Training

Trained and Ready screen with output log and summary

```
/
c0704a-s1.ufhpc
Thu Mar 28 13:50:55 EDT 2024
No GPU detected, loading a CPU version.
Running job request /blue/coguniversity/cml-prod/experiments/82f8adf4-0a4b-409a-bb61-7e0fd37141cc/data null null /blue/coguniversity/cml-prod/experiments/82f8adf4-0a4b-409a-bb61-7e0fd37141cc/out
7e0fd37141cc/model 82f8adf4-0a4b-409a-bb61-7e0fd37141cc uf-hipergator c54548dc8c34457bb417224d87756748/uf_0002/experiments/82f8adf4-0a4b-409a-bb61-7e0fd37141cc HiPerGator-Vision
2024-03-28 13:50:56.851360: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
2024-03-28 13:51:03.895015: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices not set
2024-03-28 13:51:03.896169: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file:
2024-03-28 13:51:03.896195: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2024-03-28 13:51:03.896220: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (c0704a-s1.ufhpc): /proc/driver/nvidia/version doe
2024-03-28 13:51:03.896522: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU ins
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-03-28 13:51:03.896610: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
2024-03-28 13:51:04.045667: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
2024-03-28 13:51:04.057357: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 1996350000 Hz
AKModel2 starting
Creating Image Classification Model
loading training dataset
Found 60 files belonging to 4 classes.
Using 48 files for training.
loading validation dataset
Found 60 files belonging to 4 classes.
Using 12 files for validation.
fitting training data

Search: Running Trial #1

Hyperparameter    |Value             |Best Value So Far
image_block_1/b...|vanilla           |?
image_block_1/n...|True              |?
image_block_1/a...|False             |?
image_block_1/c...|3                 |?
image_block_1/c...|1                 |?
image_block_1/c...|2                 |?
image_block_1/c...|True              |?
image_block_1/c...|False             |?
image_block_1/c...|0.25              |?
image_block_1/c...|32                |?
image_block_1/c...|64                |?
classification_...|flatten           |?
classification_...|0.5               |?
optimizer         |adam              |?
learning_rate     |0.001             |?
```

The training process used CPU instead of GPU processing, indicated by TensorFlow warnings of failing to load CUDA libraries, which was caused by no GPU support for this training experiment. The training was conducted with 20 epochs, with parameters being tested such as image block type, normalization, augmentation, convolution block depth, and optimizer settings.

```
Epoch 1/20

1/1 [==============================] - ETA: 0s - loss: 1.4780 - accuracy: 0.2812
1/1 [==============================] - 6s 6s/step - loss: 1.4780 - accuracy: 0.2812 - val_loss: 24.8129 - val_accuracy: 0.3750
Epoch 2/20

1/1 [==============================] - ETA: 0s - loss: 21.0812 - accuracy: 0.3438
1/1 [==============================] - 5s 5s/step - loss: 21.0812 - accuracy: 0.3438 - val_loss: 9.7577 - val_accuracy: 0.4375
Epoch 3/20

1/1 [==============================] - ETA: 0s - loss: 4.2606 - accuracy: 0.6250
1/1 [==============================] - 6s 6s/step - loss: 4.2606 - accuracy: 0.6250 - val_loss: 13.7769 - val_accuracy: 0.3125
Epoch 4/20

1/1 [==============================] - ETA: 0s - loss: 6.7240 - accuracy: 0.5625
1/1 [==============================] - 5s 5s/step - loss: 6.7240 - accuracy: 0.5625 - val_loss: 10.5579 - val_accuracy: 0.3750
Epoch 5/20

1/1 [==============================] - ETA: 0s - loss: 3.5321 - accuracy: 0.6250
1/1 [==============================] - 5s 5s/step - loss: 3.5321 - accuracy: 0.6250 - val_loss: 6.3897 - val_accuracy: 0.3750
Epoch 6/20

1/1 [==============================] - ETA: 0s - loss: 0.5420 - accuracy: 0.8438
1/1 [==============================] - 7s 7s/step - loss: 0.5420 - accuracy: 0.8438 - val_loss: 5.5841 - val_accuracy: 0.3750
Epoch 7/20

1/1 [==============================] - ETA: 0s - loss: 0.0382 - accuracy: 1.0000
1/1 [==============================] - 6s 6s/step - loss: 0.0382 - accuracy: 1.0000 - val_loss: 7.1130 - val_accuracy: 0.3750
Epoch 8/20

1/1 [==============================] - ETA: 0s - loss: 0.4108 - accuracy: 0.8438
1/1 [==============================] - 5s 5s/step - loss: 0.4108 - accuracy: 0.8438 - val_loss: 7.4062 - val_accuracy: 0.3750
Epoch 9/20

1/1 [==============================] - ETA: 0s - loss: 0.3791 - accuracy: 0.8438
1/1 [==============================] - 9s 9s/step - loss: 0.3791 - accuracy: 0.8438 - val_loss: 6.9181 - val_accuracy: 0.3750
Epoch 10/20

1/1 [==============================] - ETA: 0s - loss: 0.1157 - accuracy: 0.9375
1/1 [==============================] - 8s 8s/step - loss: 0.1157 - accuracy: 0.9375 - val_loss: 6.2464 - val_accuracy: 0.3750
Epoch 11/20

1/1 [==============================] - ETA: 0s - loss: 0.0251 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0251 - accuracy: 1.0000 - val_loss: 5.5852 - val_accuracy: 0.3750
Epoch 12/20

1/1 [==============================] - ETA: 0s - loss: 0.0102 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0102 - accuracy: 1.0000 - val_loss: 4.9490 - val_accuracy: 0.3750
Epoch 13/20

1/1 [==============================] - ETA: 0s - loss: 0.0059 - accuracy: 1.0000
1/1 [==============================] - 6s 6s/step - loss: 0.0059 - accuracy: 1.0000 - val_loss: 4.3583 - val_accuracy: 0.3750
Epoch 14/20

1/1 [==============================] - ETA: 0s - loss: 0.0047 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0047 - accuracy: 1.0000 - val_loss: 3.8472 - val_accuracy: 0.3750
Epoch 15/20

1/1 [==============================] - ETA: 0s - loss: 0.0074 - accuracy: 1.0000
1/1 [==============================] - 6s 6s/step - loss: 0.0074 - accuracy: 1.0000 - val_loss: 3.4286 - val_accuracy: 0.3750
Epoch 16/20

1/1 [==============================] - ETA: 0s - loss: 0.0077 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0077 - accuracy: 1.0000 - val_loss: 3.0993 - val_accuracy: 0.5000
Epoch 17/20

1/1 [==============================] - ETA: 0s - loss: 0.0108 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0108 - accuracy: 1.0000 - val_loss: 2.8638 - val_accuracy: 0.5000
Epoch 18/20

1/1 [==============================] - ETA: 0s - loss: 0.0133 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0133 - accuracy: 1.0000 - val_loss: 2.7114 - val_accuracy: 0.5000
Epoch 19/20

1/1 [==============================] - ETA: 0s - loss: 0.0138 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0138 - accuracy: 1.0000 - val_loss: 2.6300 - val_accuracy: 0.5000
Epoch 20/20

1/1 [==============================] - ETA: 0s - loss: 0.0125 - accuracy: 1.0000
1/1 [==============================] - 5s 5s/step - loss: 0.0125 - accuracy: 1.0000 - val_loss: 2.6136 - val_accuracy: 0.4375
[2K
[2K
Trial 1 Complete [00h 01m 59s]
val_loss: 2.613588809967041

Best val_loss So Far: 2.613588809967041
```

Training the Image Recognition model with 20 epochs

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 180, 180, 3)]     0
_________________________________________________________________
cast_to_float32 (CastToFloat (None, 180, 180, 3)       0
_________________________________________________________________
normalization (Normalization (None, 180, 180, 3)       7
_________________________________________________________________
conv2d (Conv2D)              (None, 178, 178, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 176, 176, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 88, 88, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 88, 88, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 495616)            0
_________________________________________________________________
dropout_1 (Dropout)          (None, 495616)            0
_________________________________________________________________
dense (Dense)                (None, 4)                 1982468
_________________________________________________________________
classification_head_1 (Softm (None, 4)                 0
=================================================================
Total params: 2,001,867
Trainable params: 2,001,860
Non-trainable params: 7
_________________________________________________________________
```

Output Summary

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.478	28.12%	24.8129	37.50%
2	21.0812	34.38%	9.7577	43.75%
3	4.2606	62.50%	13.7769	31.25%
4	6.724	56.25%	10.5579	37.50%
5	3.5321	62.50%	6.3897	37.50%
6	0.542	84.38%	5.5841	37.50%
7	0.0382	100.00%	7.113	37.50%
8	0.4108	84.38%	7.4062	37.50%
9	0.3791	84.38%	6.9181	37.50%
10	0.1157	93.75%	6.2464	37.50%
11	0.0251	100.00%	5.5852	37.50%
12	0.0102	100.00%	4.949	37.50%
13	0.0059	100.00%	4.3583	37.50%
14	0.0047	100.00%	3.8472	37.50%
15	0.0074	100.00%	3.4286	37.50%
16	0.0077	100.00%	3.0993	50.00%
17	0.0108	100.00%	2.8638	50.00%
18	0.0133	100.00%	2.7114	50.00%
19	0.0138	100.00%	2.63	50.00%
20	0.0125	100.00%	2.6136	43.75%

Table summary of training accuracy

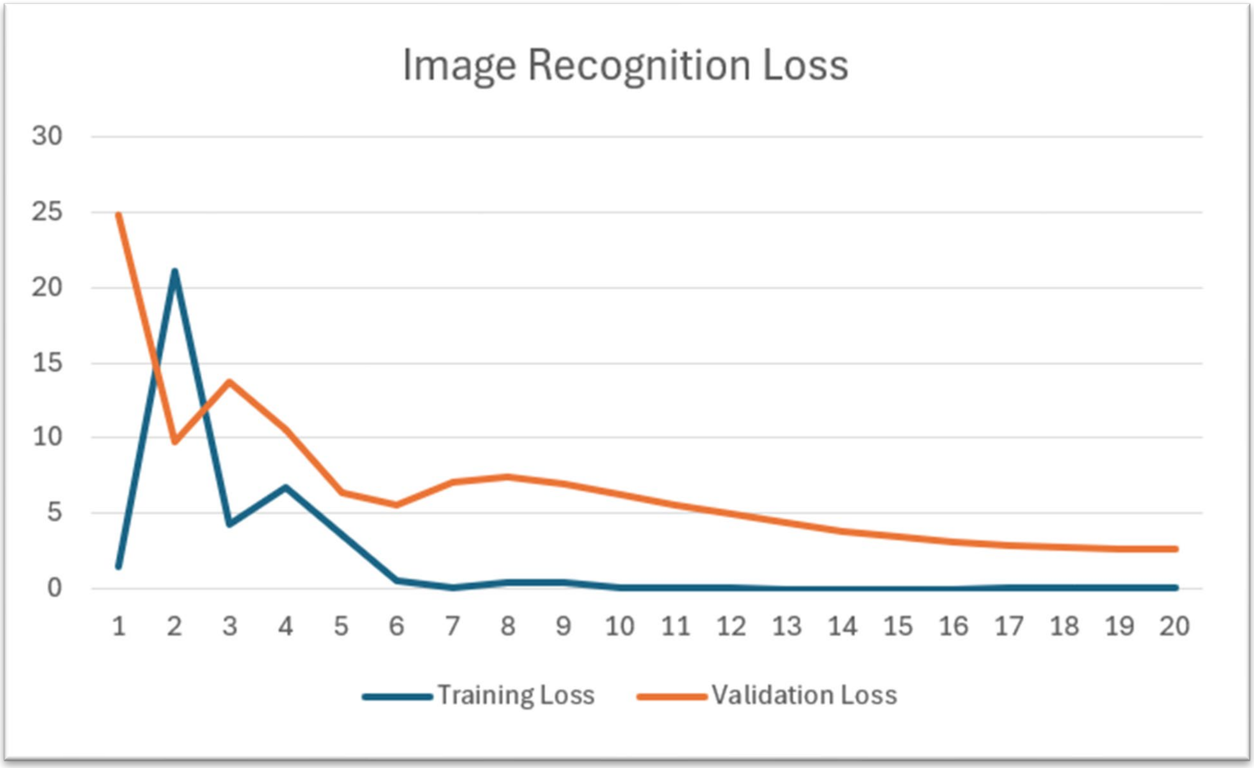


Image Recognition Loss
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Training Loss
Validation Loss

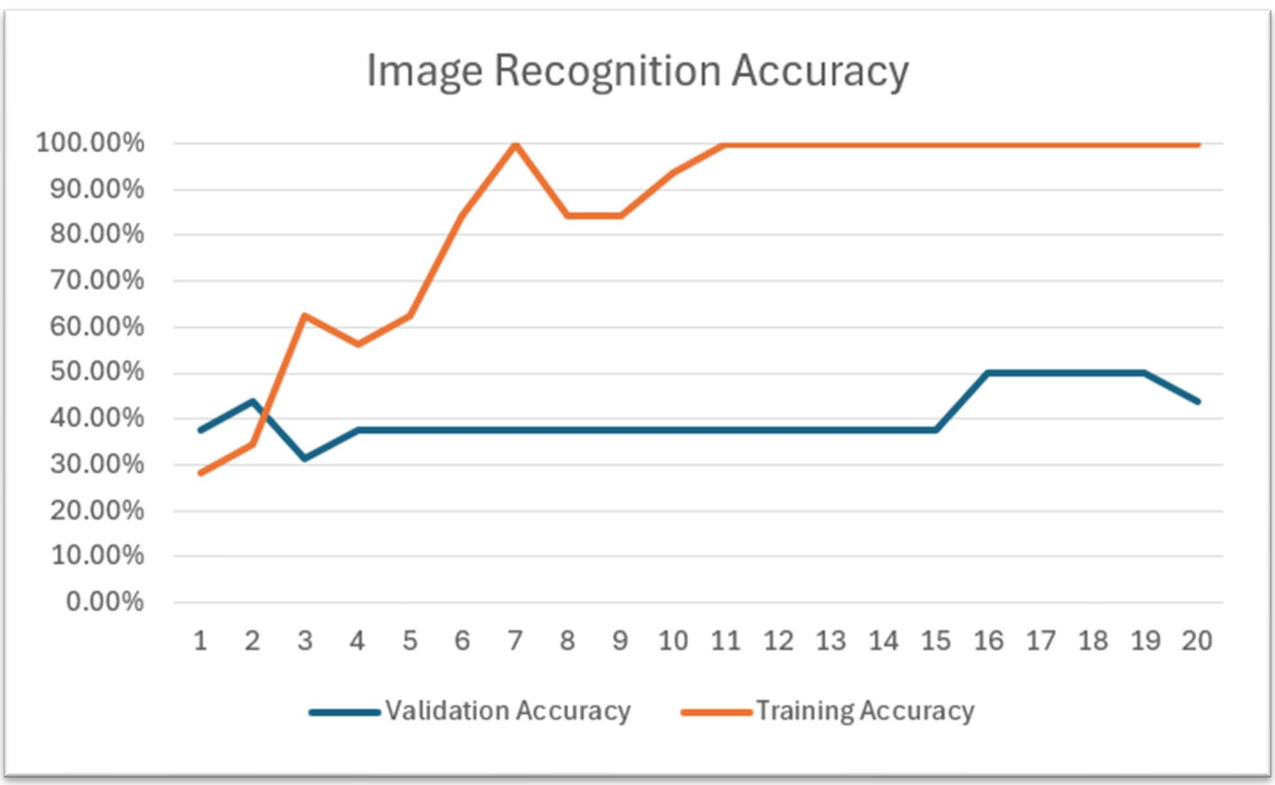
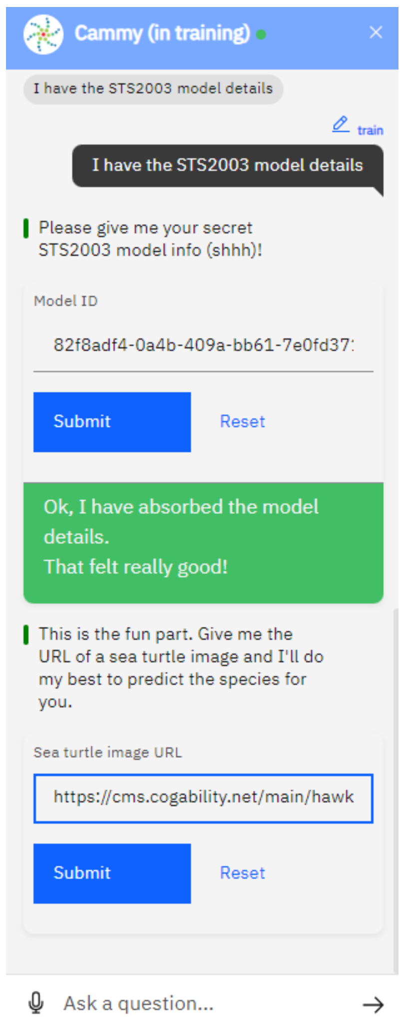


Image Recognition Accuracy
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Validation Accuracy
Training Accuracy

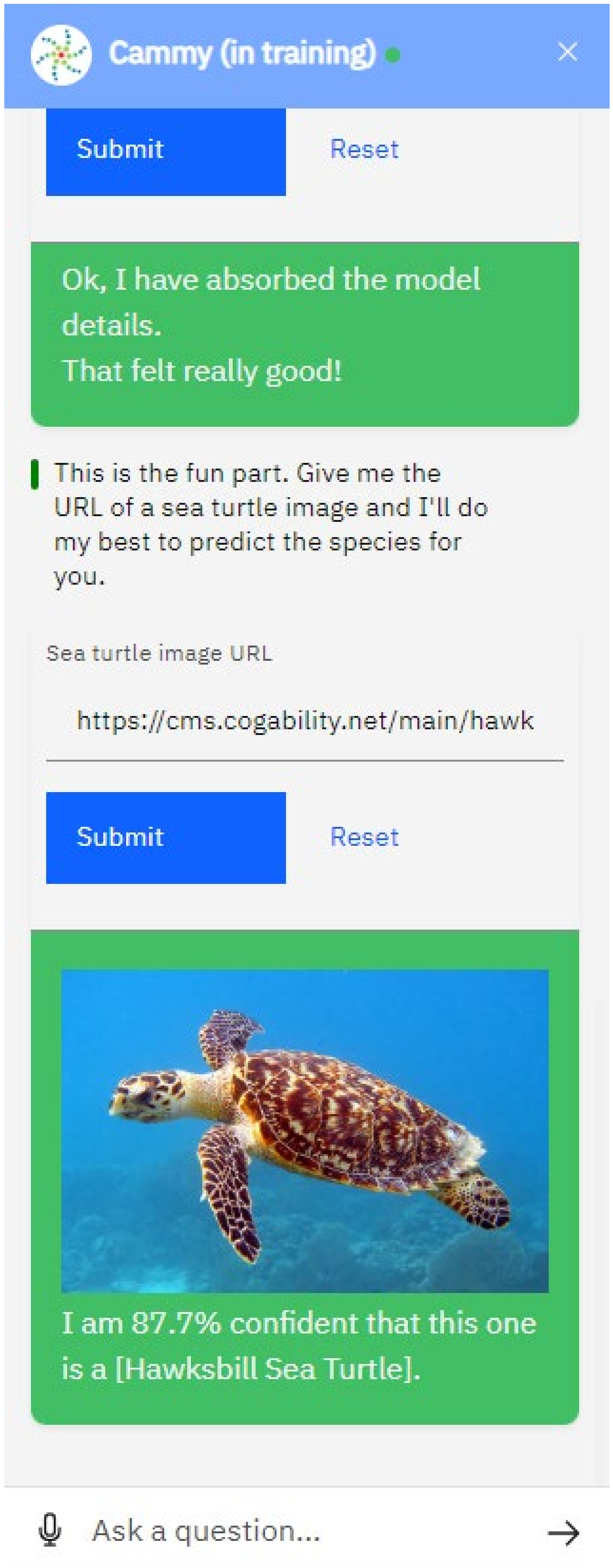
Based on the training results, the training accuracy reached 100% by Epoch 7, indicating the model is perfectly fitting the training data. However, the validation accuracy does not improve at similar rate, only reaching 43.7% by end of training. This indicates that the training data is overfitted and has a lack of different data to analyze and compare.

Image Recognition with Cammy

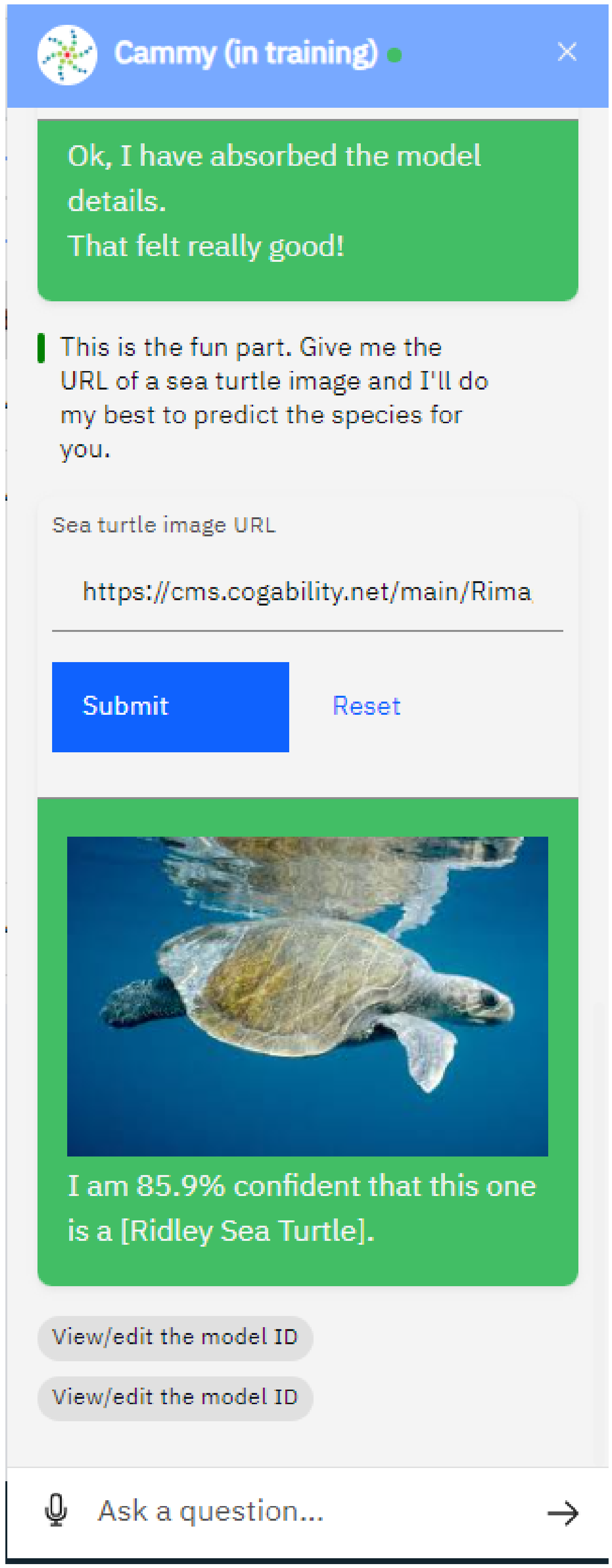
With the model trained on the images and labels, the model was applied to Cammy chatbot to allow it to predict images uploaded. For testing the image recognition, we used 9 test images provided by CogUniversity in a csv file. There were 2 test images per species that Cammy had not seen before, and each of the image urls were pasted into the input and Cammy made the species prediction.



Attaching the trained AI model to Cammy



Correct prediction for Hawksbill Sea Turtle

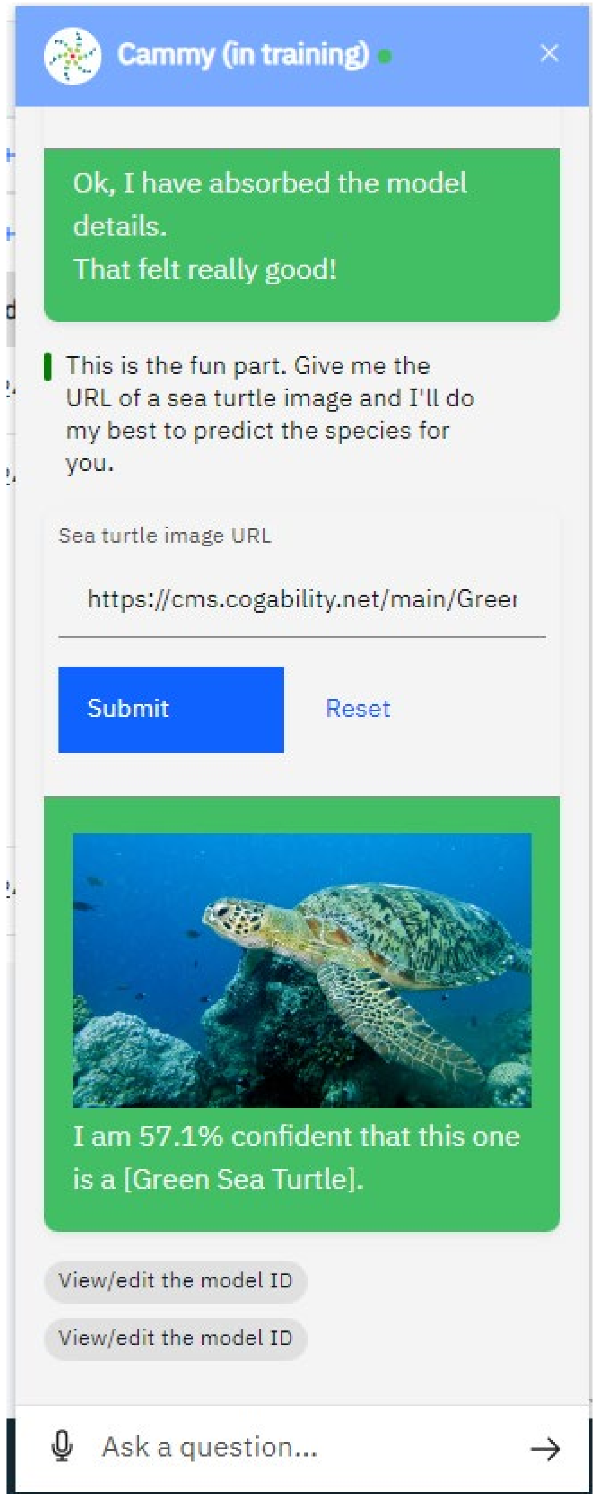


Correct prediction for Ridley Sea Turtle

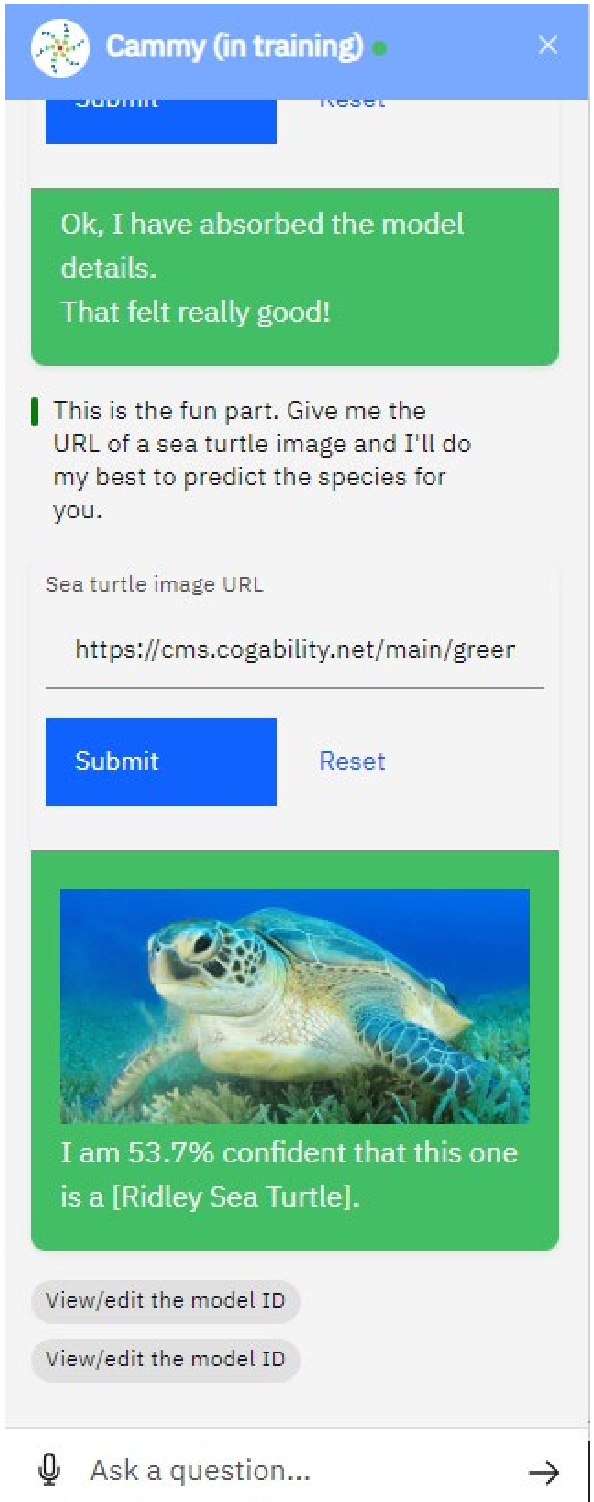
Cammy had greatest success at recognizing the Hawksbill Sea Turtles, with accuracy over 85%, and was also able to identify the Ridley Sea Turtle, with accuracy 60 – 85%.

Sea Turtle Test Images	Prediction	Accuracy
Hawksbill sea turtles		
https://cms.cogability.net/main/hawksbill_sea_turtle.jpeg	TRUE	87.70%
https://cms.cogability.net/main/himages.jpeg	TRUE	95%
Loggerhead sea turtles		
https://cms.cogability.net/main/La29582bef3bdf958ffcb31801a9639cd.jpeg	False - Hawksbil	84.60%
https://cms.cogability.net/main/L218933603_78ec25bc66R.jpg	False - Green	47.90%
Ridley sea turtles		
https://cms.cogability.net/main/Rimages.jpeg	TRUE	85.90%
https://cms.cogability.net/main/ridley_sand.jpeg	TRUE	61%
Green sea turtles		
https://cms.cogability.net/main/Green_Turtle_Chelonia_mydas_6133097542.jpeg	TRUE	57.10%
https://cms.cogability.net/main/green_weeds.jpeg	False - Ridley	53.70%
https://cms.cogability.net/main/green-sea-turtle_australia_shutterstock.jpeg	False - Hawksbb	49.50%

Cammy failed to recognize the Loggerhead Sea Turtle images, predicting the images were of HawksBill at 85% accuracy, and also a 47% accuracy predicton of HawksBill as a Green Sea Turtle. There were mixed results with predicting the Green Sea Turtle images, with only 1 of the 3 images being identified correctly with 57% accuracy, and the other 2 images being identified at 50% accuracy as Ridley and Hawksbill Sea Turtles.

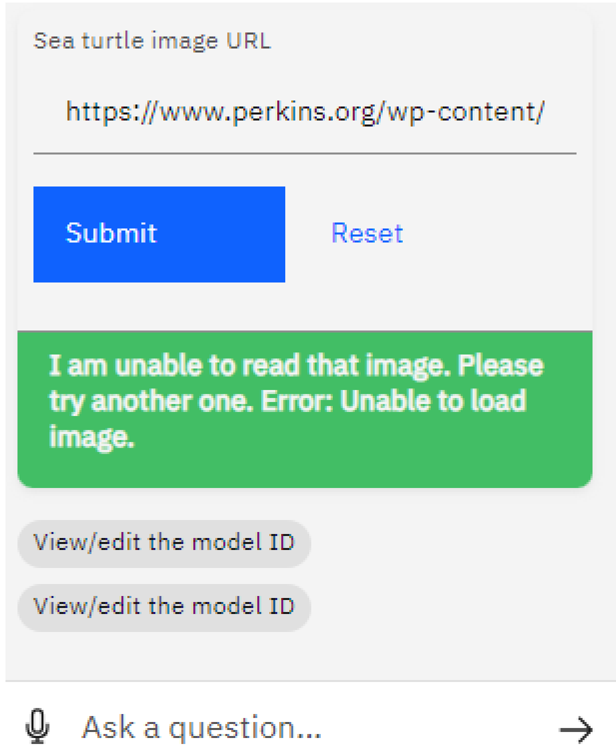


Correct prediction for Green Sea Turtle



Incorrect prediction for Green Sea Turtle

Incorrect prediction for Loggerhead Sea Turtle



There was an error encountered when trying to use an external image URL that was not part of the test images provided by CogUniveristy.

Summary and Recommendations

Since the image recognition had mixed results, it indicates that the image dataset used had better and worse images used to train the model for the species. The Hawksbill Sea Turtles were highly accurate predicted, and their training images were highest quality and easy to identify with contrast and color. The Ridley Sea Turtles were also accurate, because they had distinct color difference of greyer scales, and the photos were of lowest resolution. Both the Green Sea Turtle and Loggerhead Sea Turtle training images were of too many varieties of angles and backgrounds, and included I mages and close ups, making it difficult for the AI model to properly identify what the Sea Turtle looks like.

Since external photos were unable to be used for testing, it narrows the experiment to just the 9 photos provided. As mentioned earlier, the estimated use case for Cammy would be to help save sea turtles with target audience as the public. Ordinary visitors to beaches and aquariums would most likely use smartphone cameras to photograph the creatures, and wouldn't take photographs like the majority of the training dataset, besides the photos of the Ridley Sea Turtle which were taken on land.

To improve the accuracy of the training model, the sea turtle images should be organized before being uploaded to the labeling program. It might help to isolate the sea turtle from its background.so that the background does not become a factor in identifying the actual subject. The photos should contain the entire sea turtle and not be cropped. The upclose photos of the sea turtle where it is just the head should be removed from the dataset. All copyright images should also be removed, but that might make the training impossible.

The time to train the images is based on the number of images trained on. If want to make the process faster, could use a smaller dataset, but the results might not be of highest accuracy. To use copyright free images for testing, could use online repositories of copyright-free photos such as Pexels, Unsplash, and Kaggle. Another option is to take the photos yourself or to hire a photographer, so all of the training is original and authentic.

Something to consider bias is the geo-location, time of day that the photos were taken, and if the photos were taken underwater or on land. There may be a difference in male or female sea turtles, that are not discernable from this dataset.

To get more accurate image recognition, the images should be a much larger data set, at least 100 or 1000 images would help the training process. The subject should be clearly defined in the training images and remove all cropped out subjects. Centering the subject and removing background will also improve identification of the actual subject.

References

"MakeSense." MakeSense.ai, n.d., www.makesense.ai/. Accessed 3/28/24

"CogAbility." CogAbility, n.d., www.cogability.com/. Accessed 3/28/24