Identifying Underwater Trash Using Neural Networks and Natural Language Processing

Project 5

Julian Cheng


100,000 marine animals die every year
5.25 trillion plastic pieces in the ocean

The Problem:

Can we develop a model that can help clean up the ocean floor of sunken debris?

My Answer:

I wanted to develop a model that could assist autonomous robots in cleaning up ocean debris from the sea floor. This system would be driven by a neural network, with the goal of consistently differentiating trash from organisms living in the vicinity. Additionally, I wanted to leverage natural language processing to identify the most prevalently predicted images.

Process - CNN

Seafloor images

↓

Standardize dimensions

↓

Convert grayscale

↓

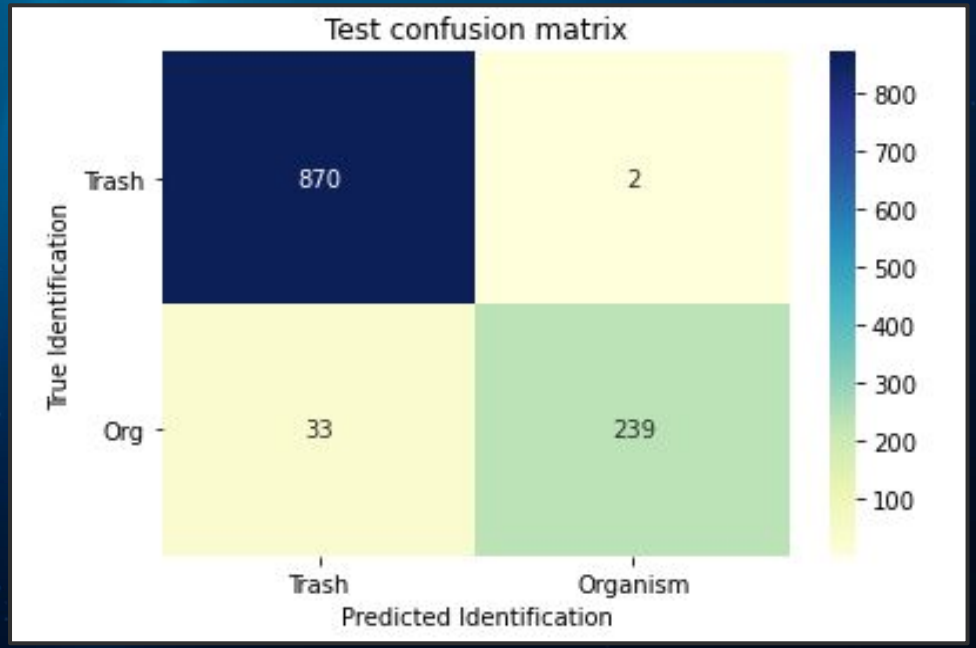
Label classification

Example "Object" Images

Example "Biology" Images

Performance - CNN

Accuracy	0.969
Precision	0.963
Recall	0.998
F1 Score	0.980



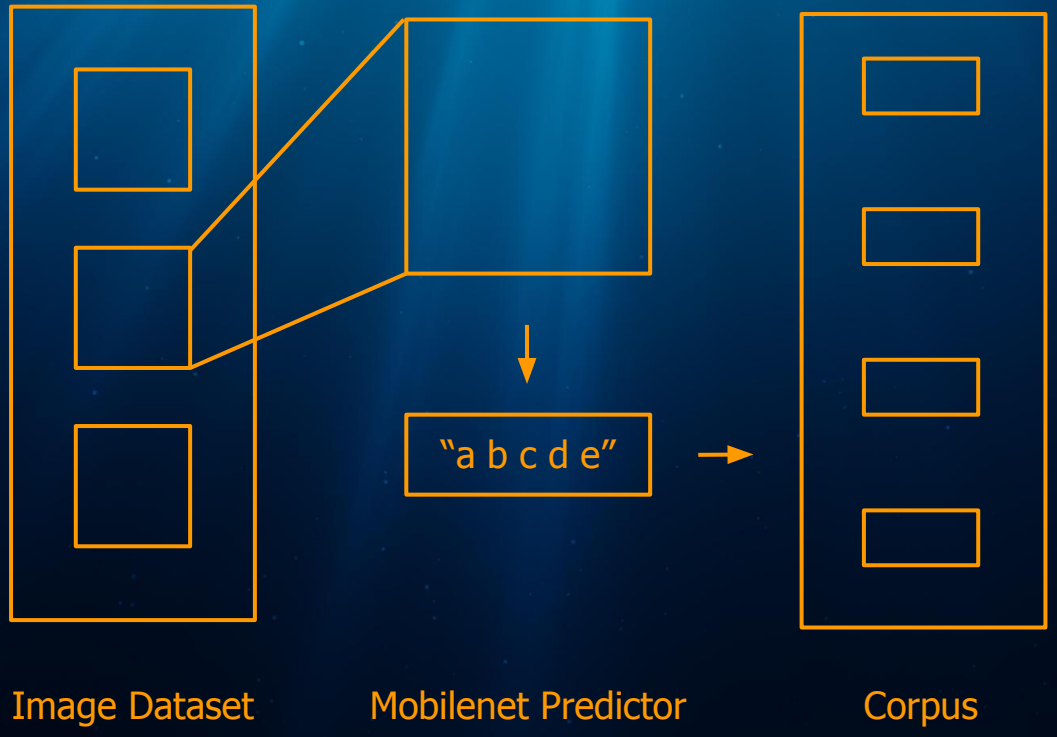
Transfer Learning & NLP - Motivation

Many autonomous robots use broad strokes which are erroneous and limited

Advanced robots limited by the transportable tools

I want to predict the most commonly encountered objects and prepare robots accordingly

Transfer Learning & NLP - Process



Transfer Learning & NLP - Results

Trash Groupings/"Topics"
0: book_jacket binder envelope
1: stage dishwasher monitor
2: monitor screen television
3: electric_ray dugong hammerhead
4: plastic_bag shower_cap packet
5: nematode isopod scorpion
6: menu washer dishwasher
7: water_bottle pop_bottle beaker
8: wreck scuba_diver coral_reef
9: jellyfish hammerhead digital_clock

Organism Groupings/"Topics"
0: tiger_shark great_white_shark dugong
1: barn_spider rock_crab king_crab
2: water_snake sea_snake thunder_snake
3: gar coho sturgeon
4: monitor television screen
5: axolotl banded_gecko eel
6: fiddler_crab scorpion hair_slide
7: tailed_frog fiddler_crab eel
8: electric_ray dugong stingray
9: quill nematode ladle

Conclusion

In this project, I developed:

An accurate convolutional neural network

A prediction list of the most commonly encountered objects

Looking forwards

- KNN Classifier
- Treatment Priority List

Appendix

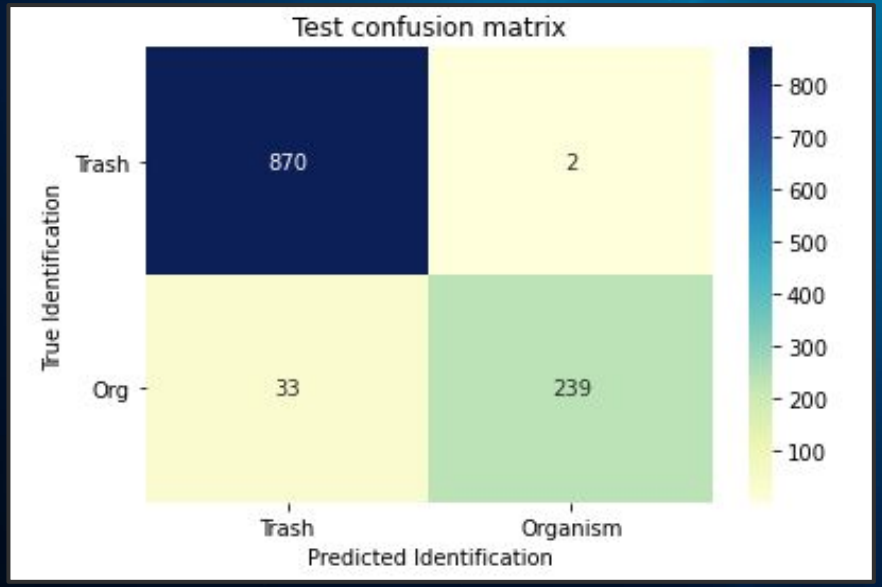
Sources: https://conservancy.umn.edu/handle/11299/214366

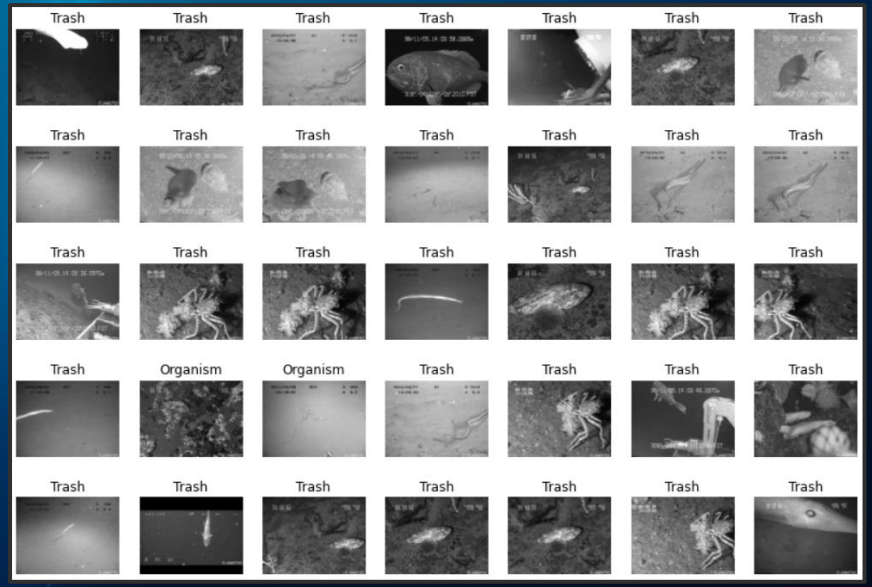
NMF Groupings

Trash Groupings/"Topics"
0: book_jacket binder envelope
1: stage dishwasher monitor
2: monitor screen television
3: electric_ray dugong hammerhead
4: plastic_bag shower_cap packet
5: nematode isopod scorpion
6: menu washer dishwasher
7: water_bottle pop_bottle beaker
8: wreck scuba_diver coral_reef
9: jellyfish hammerhead digital_clock

Organism Groupings/"Topics"
0: tiger_shark great_white_shark dugong
1: barn_spider rock_crab king_crab
2: water_snake sea_snake thunder_snake
3: gar coho sturgeon
4: monitor television screen
5: axolotl banded_gecko eel
6: fiddler_crab scorpion hair_slide
7: tailed_frog fiddler_crab eel
8: electric_ray dugong stingray
9: quill nematode ladle

Appendix






```
test precision:  0.9634551495016611
test recall:  0.9977064220183486
test F1:  0.980281690140845
```

Erroneous Classifications

Appendix

NN Summary

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 300, 400, 5)       50
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 100, 133, 5)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 100, 133, 10)      460
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 33, 44, 10)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 33, 44, 20)        1820
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 11, 15, 20)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3300)              0
_________________________________________________________________
dense_3 (Dense)              (None, 30)                99030
_________________________________________________________________
dense_4 (Dense)              (None, 10)                310
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 22
=================================================================
Total params: 101,692
Trainable params: 101,692
Non-trainable params: 0
_________________________________________________________________
```