Celestine Tan | DSI-SG-26 Capstone

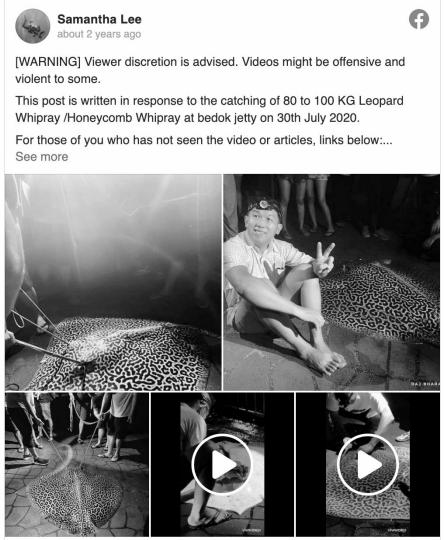
WHAT THE FISH?

(Sustainable fishing via image classification)

TABLE OF CONTENTS

80kg stingray caught and "cut up" at Bedok Jetty: Netizens, Acres slam killing of vulnerable species

See 23 More Photos



"Honeycomb rays are vulnerable because they are long-lived, have few young, and take four to five years to reach sexual maturity. Mothers carry their young for 12 months before giving birth to them live."

Native to Singapore, critically endangered ray prized for fins to get better protection

The shovelnose ray (Rhynochobatus australiae), a Singapore native also known as the bottlenose wedgefish, will soon get better protection. PHOTO: ISTOCKPHOTO

Critically endangered rays chopped up & sold in S'pore supermarkets for S$1.30 per 100g

Restaurants in Singapore serve it as a 'shark head' delicacy when it is not even a shark.

Ashley Tan | September 05, 2019, 01:20 AM

"Shovelnose rays have been classified as "Critically Endangered" by the International Union for Conservation of Nature (IUCN) since Aug. 28, 2019. They are threatened by commercial fishing and harvesting."

01 BACKGROUND

The onset of Covid-19 has increased the popularity of recreational fishing in Singapore.

Along with this, there have been more reports of people reeling in endangered species (e.g. honeycomb rays, eagle rays, shovelnose rays).

Singapore is rich in biodiversity, and it can be difficult to differentiate between species. However it is of utmost importance that catch and release is practiced, especially when it comes to the creatures that are more vulnerable.

Challenges

Lack of education about the marine biodiversity in the sea surrounding Singapore

People who fish recreationally do not practice catch and release, vulnerable species put at risk

Solution

Ability to quickly identify marine life once it has been hooked

Leverage on supervised deep learning to alert users if species caught is endangered, to facilitate catch and release

01 BACKGROUND
(8 Target Classes)

Commonly caught by fishermen, juveniles should still be released

To make the model more complex, honeycomb grouper and blue spot ray were included due to similarity in shape and pattern to hybrid grouper and honeycomb ray respectively

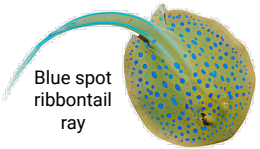
Invasive Species

(A result of fish farm breeding in Johor. Hybrid groupers compete with native fish for food causing strain on the natural ecosystem.)

Hybrid grouper

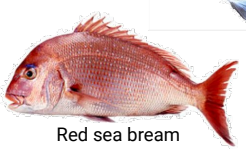
Common Species

Honeycomb grouper



Blue spot ribbontail ray

Seabass

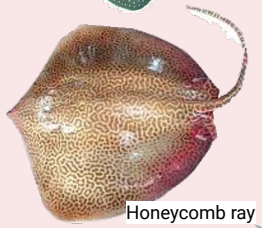


Red sea bream

Queenfish

Vulnerable Species

Shovelnose ray



Honeycomb ray

Dinner!

Release at all costs

02 METHODOLOGY

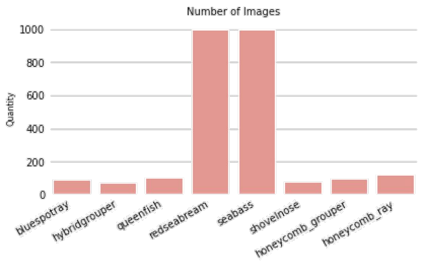
Scraped from Google images using an API

Pre-trained models chosen based on top-1 accuracy, and time taken per step

Models run:
- Baseline CNN
- 3 pre-trained models
- 1 ensemble NN

Feature extraction (freeze all layers)
Fine tuning (unfreeze last 2 layers)
Evaluate based on accuracy

Data Collection | Data Preparation | Model Selection and Training | Hyperparameter Tuning | Evaluation and Predictions



Discard irrelevant images
Train, test, validation split into different sub-folders
Create image pipeline
Resize and augment images (imbalanced classes)
Pre-processing (based on pre-trained model inputs)

More details in 03 Modelling

03 MODELLING

Common across all 4 models:
Loss: categorical cross entropy
Metric: accuracy

For 3 pre-trained models:
Batch normalization layers not unfrozen during fine tuning
Regularization also included: dropout at last 2 layers

Vanilla CNN (Baseline)
(No feature extraction done)

Architecture: 2 convolutional layers (256, 128) with max pooling

Optimizer: adam (default lr 1e-3)

Regularization:
early stopping (patience = 5, min delta = 1e-3)

ResNet50V2
Feature extraction (accuracy - 99.1%)

Unfreeze layers : 'conv5'

Optimizer: adam (lr = 1e-4, epsilon = 1e-3)

Regularization:
early stopping (patience = 10, min delta = 1e-5)

VGG16
Feature extraction (accuracy - 98.9%)

Unfreeze layers : 'block5'

Optimizer: RMSprop (lr = 1e-4)

Regularization:
early stopping (patience = 8, min delta = 1e-5)

EfficientNetB0
Feature extraction (accuracy - 99.9%)

Unfreeze layers : 'block7a', 'top"

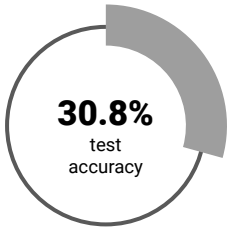
Optimizer: adam (lr = 1e-4, epsilon = 1e-7)

Regularization:
early stopping (patience = 10, min delta = 1e-6)

03 MODELLING
(Evaluation Summary)

Best performing model after fine tuning was: **EfficientNetB0**

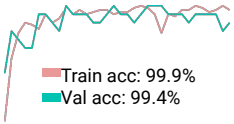
Vanilla CNN (Baseline)



ResNet50V2

99.5%
test accuracy

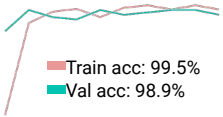
4
images misclassified



VGG16

99.6%
test accuracy

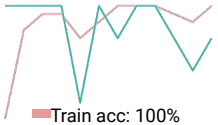
3
images misclassified

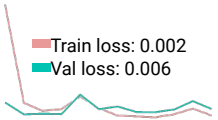


EfficientNetB0

99.7%
test accuracy

2
images misclassified

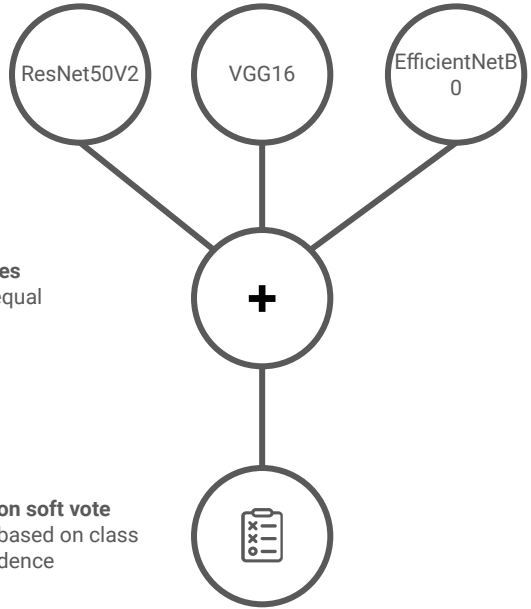




03 MODELLING
(Ensemble model)

Can we get better results?

Models run



Sum of confidences
All models given equal weightage

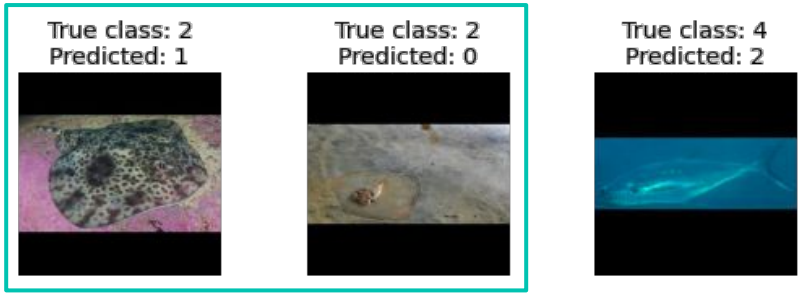
Prediction based on soft vote
Final target class based on class with highest confidence

Resulting accuracy was **slightly worse** than our best performing model

All 3 models classified the first 2 images wrongly.

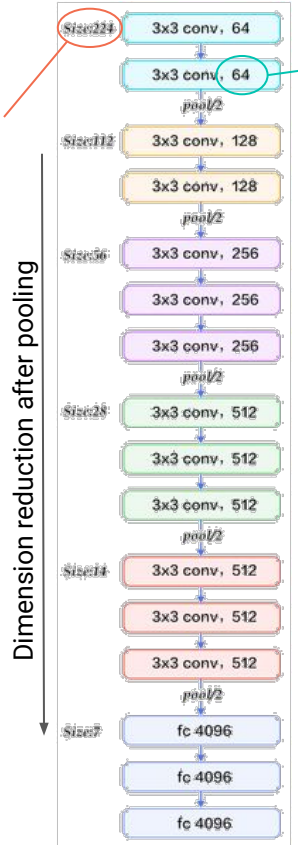
99.6%
test accuracy

3
images misclassified



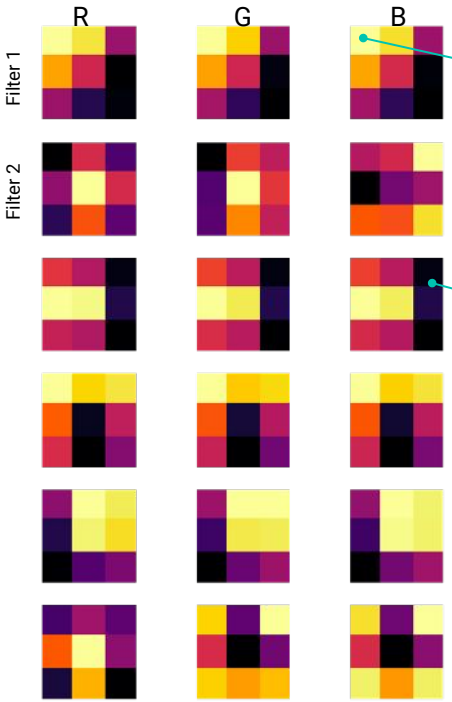
04 DECODING
(VGG16 Architecture)

Input size:
224 x 224 pixels



Dimension reduction after pooling

Number of filters in convolution
(Example is only 6 out of the 64)



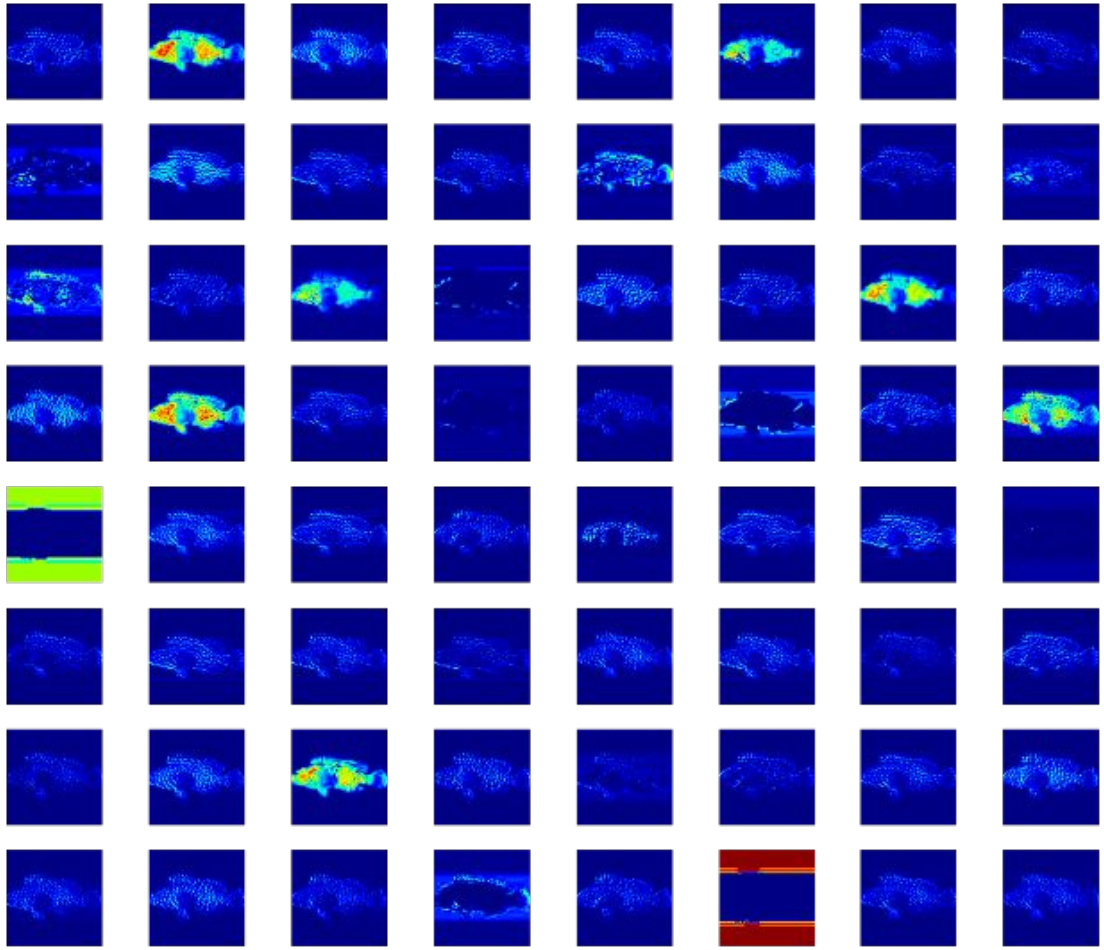
Excitatory / large weights
- activates processing elements

Inhibitory / small weights
- neurons don't get activated

04 DECODING

(Visualizing the first layer)

Original image



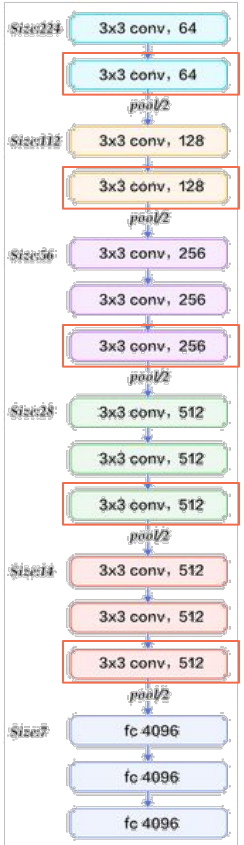
04 DECODING

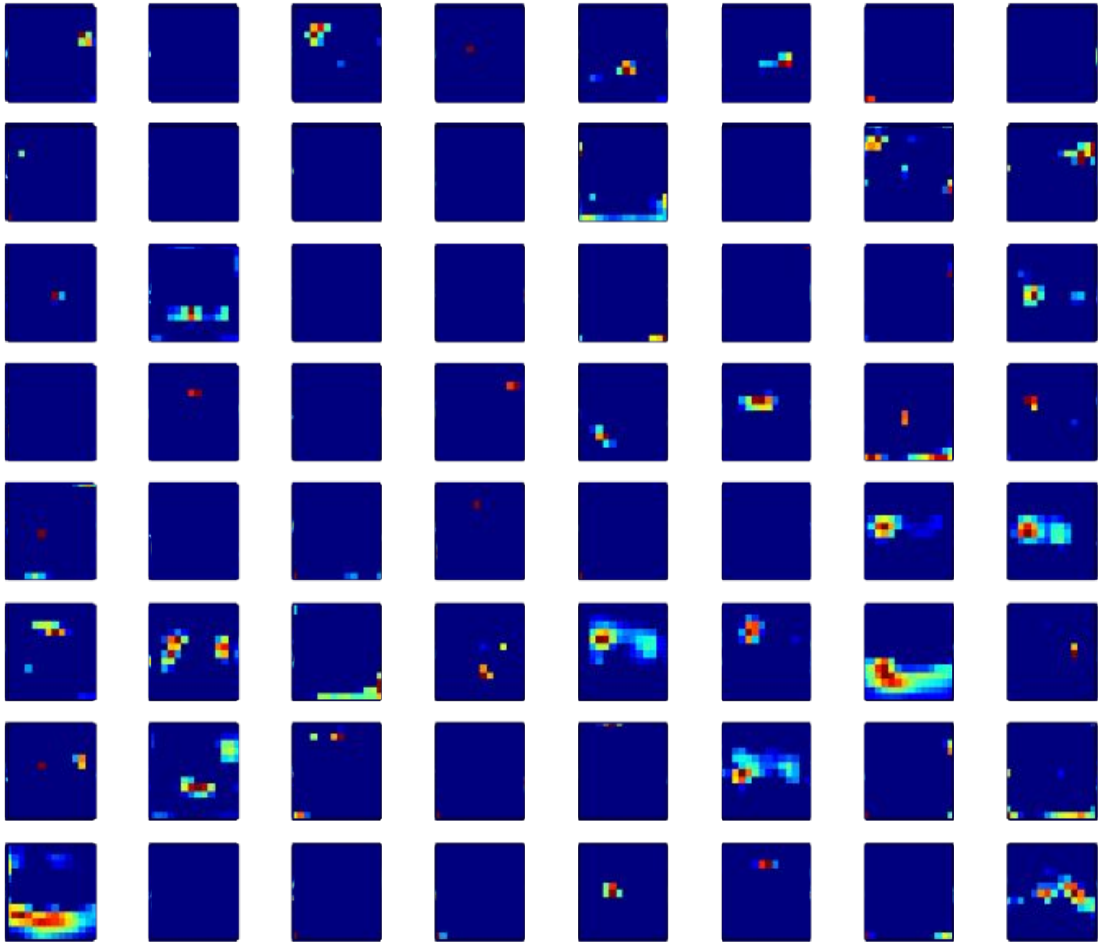
(Visualizing feature extraction)

First few layers extract general features (e.g. shape)

The last few layers extract more specific features

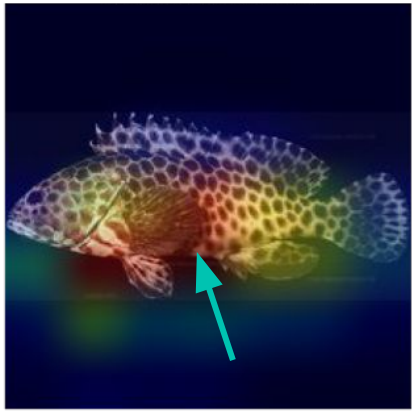
This is due to the filters in the convolutions



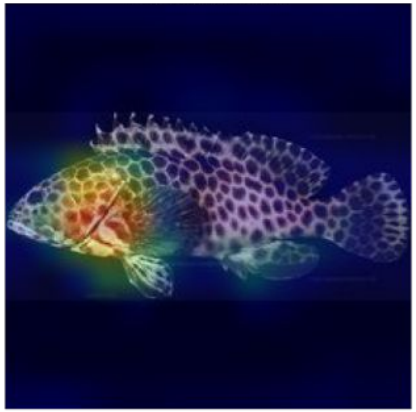


04	DECODING (Grad-CAM: model performance)

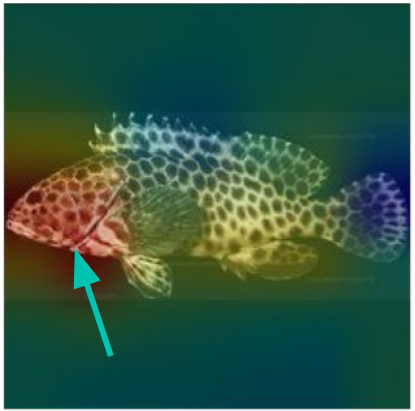
99.5%
test accuracy
Grad-CAM: ResNet50V2



99.6%
test accuracy
Grad-CAM: VGG16



99.7%
test accuracy
Grad-CAM: EfficientNetB0

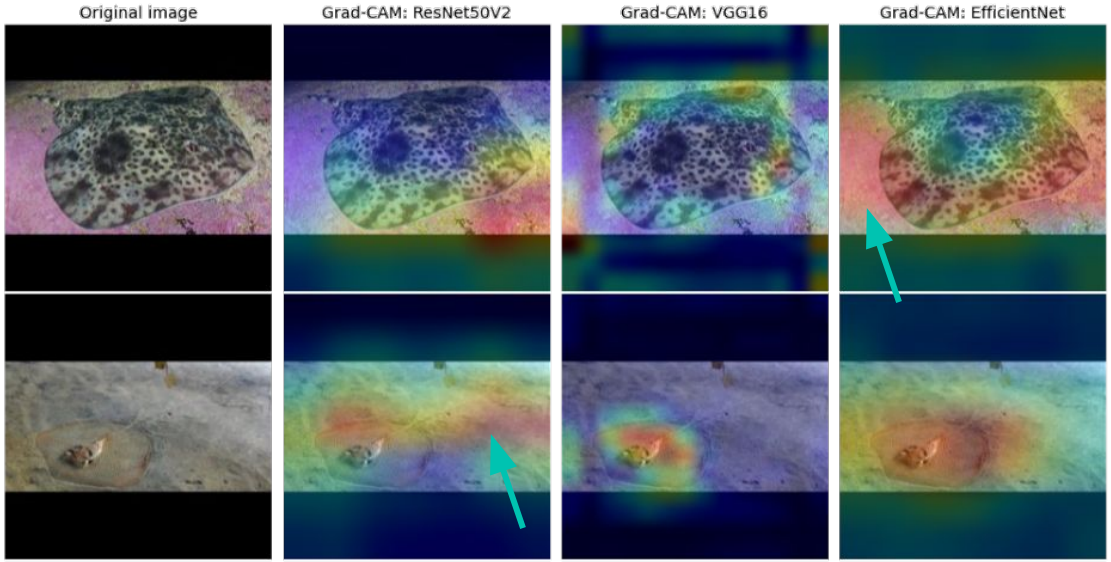


Gradient class activation maps (Grad-CAM) tell us which features were most important for predictions

EfficientNetB0 performs better as it identifies the fish's head, instead of random fish parts

04 DECODING

(Grad-CAM: misclassification)

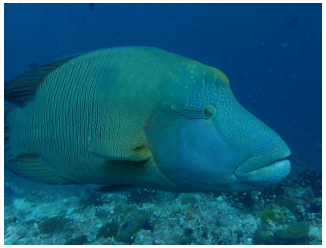


Models seem to be confused as to what is an important feature
Misclassification happens for stingray subjects that are not distinct from background

Pass in more images of ‘camouflaged’ stingrays to improve accuracy

05 CONCLUSION
(Future Improvements)

Before

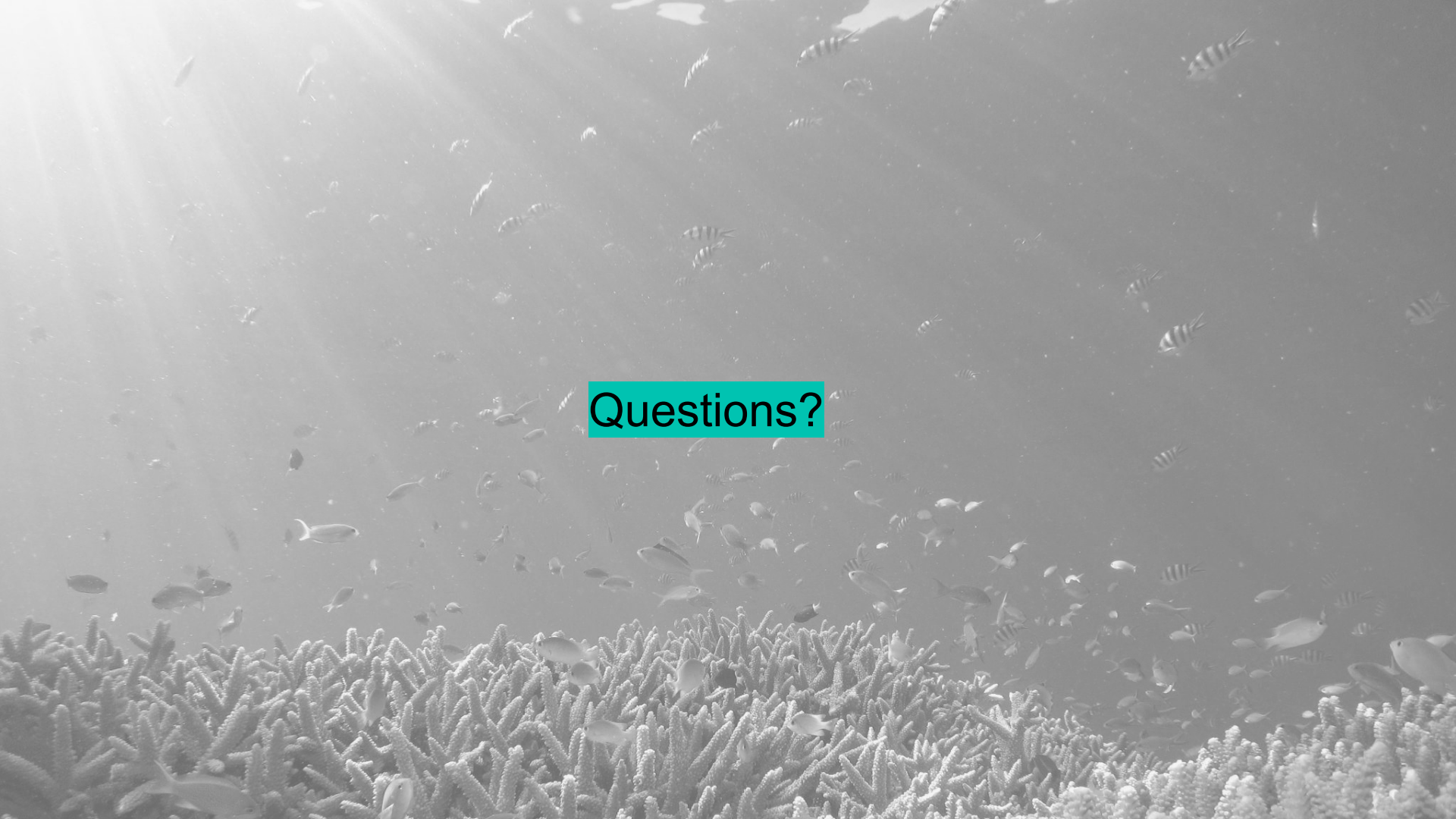


After

- Passing more training data for ‘camouflaged’ subjects
- Adding in more classes for identification
- Work on a mobile application that people can use on the go (WIP)
- Explore colour correction for photos that were taken underwater, as these lack a red filter
- Explore how to identify length of catch, as this would help differentiating juveniles from adults

05 CONCLUSION

01	02	03	04	05
To identify vulnerable species caught while fishing	1 baseline CNN	Baseline accuracy: 30.8%	Explore how to improve accuracy by visualizing the CNN black box	Work on deploying on edge
Encourage catch and release	3 pre-trained models	EfficientNetB0 acc: 99.7%	Pass in more images of camouflaged subjects for training	Expand number of classes, improve accuracy
	Ensemble NN	Ensemble NN acc: 99.6%		


Questions?