Aquatic Species Detection and Classification using Deep Learning

CAPSTONE 3

Overview

- Introduction
- Dataset
- Modeling
- Evaluation
- Conclusion

Introduction

- 436 zoo and aquarium in US
- Population and health monitoring of marine species is essential
- Image sensors better alternative to invasive techniques
- Convolutional neural network is applicable
- Goal: Develop aquatic species detection and classification system from aquarium images

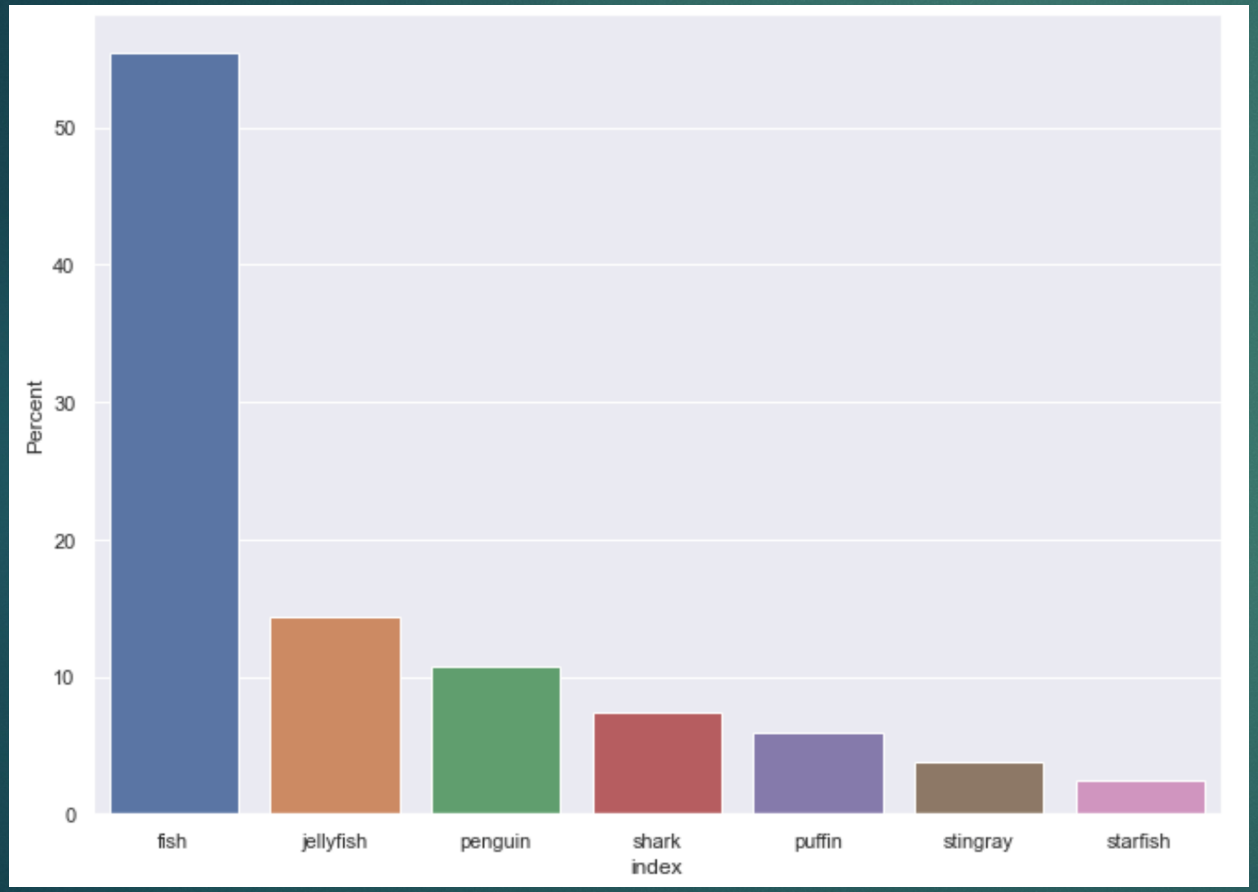
Contribution

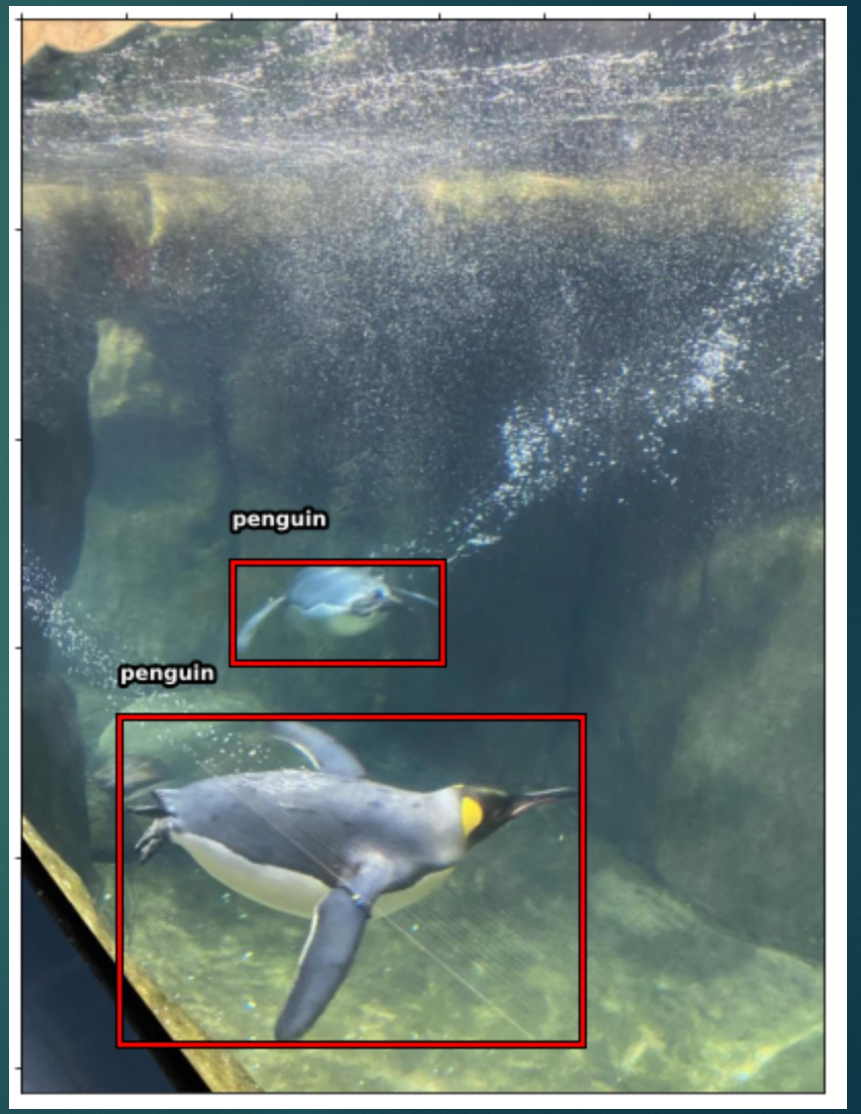
- Custom train a deep learning model for detection and classification task
- Implement the trained model in form of a web application

Dataset

- Public dataset available from Roboflow
- 637 images and annotations
- 7 classes
  - Fish
  - Jellyfish
  - Penguin
  - Shark
  - Puffin
  - Stingray
  - Starfish

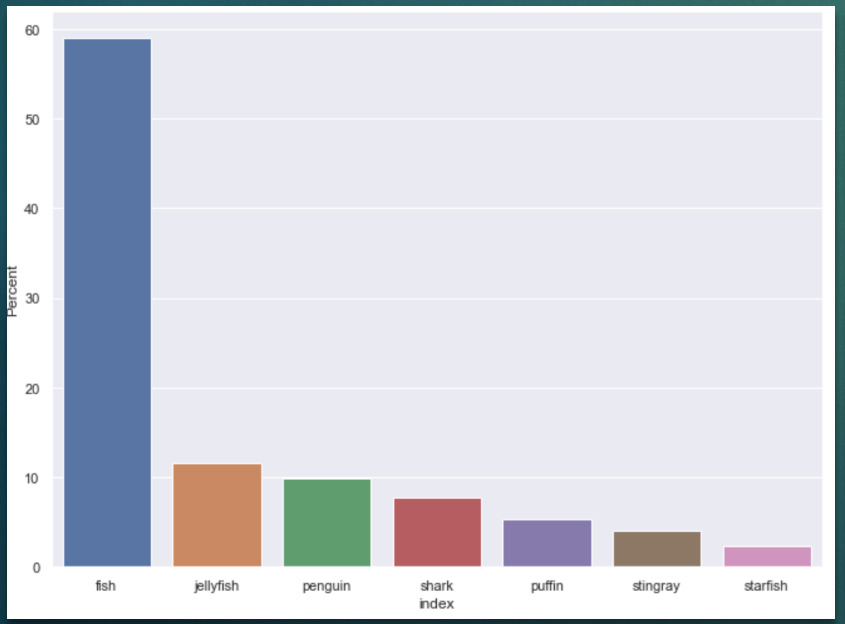
Dataset


Percent
50
40
30
20
10
0
fish
jellyfish
penguin
shark
puffin
stingray
starfish
index

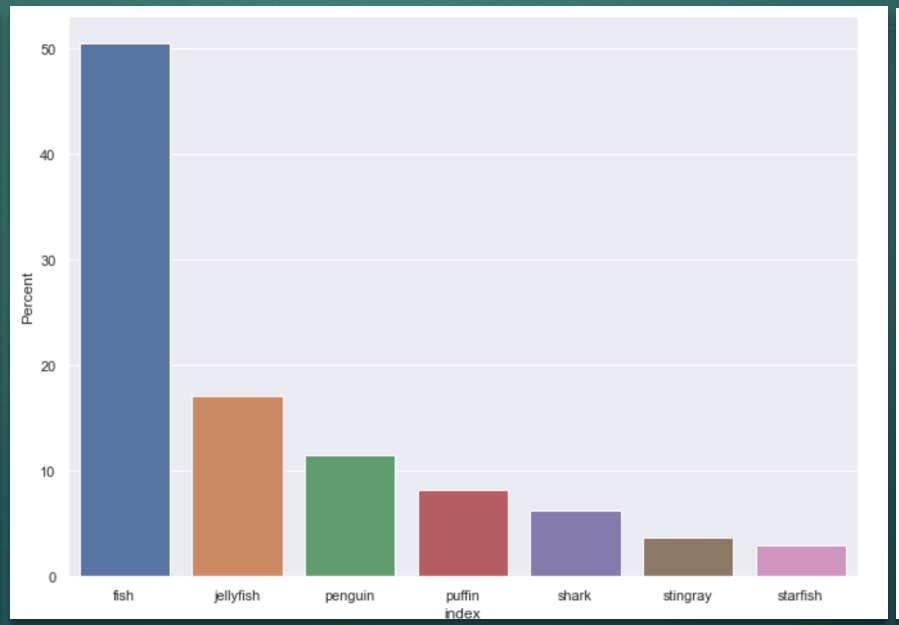

penguin
penguin

Dataset

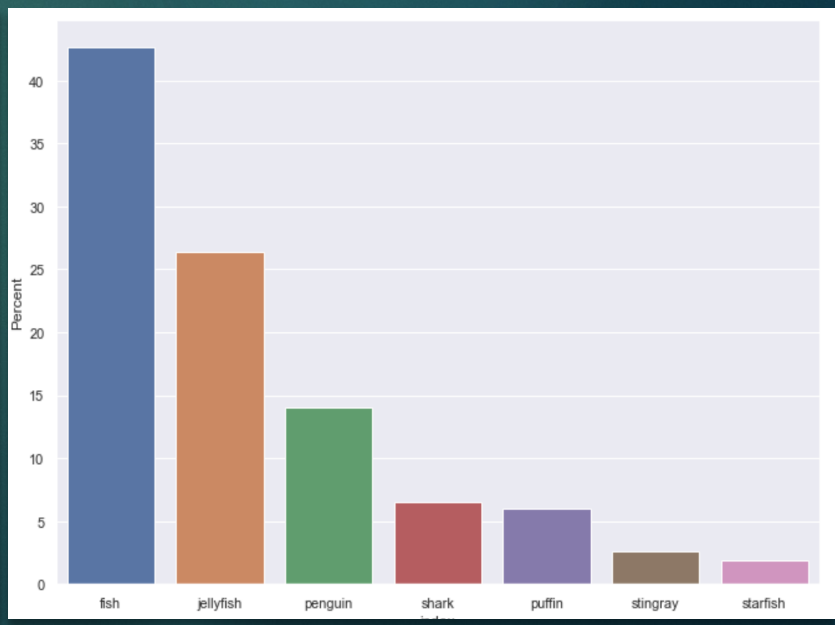
- Already divided into training, validation and test sets
- 447 images in training set
- 127 images in validation set
- 63 images in test set



Training



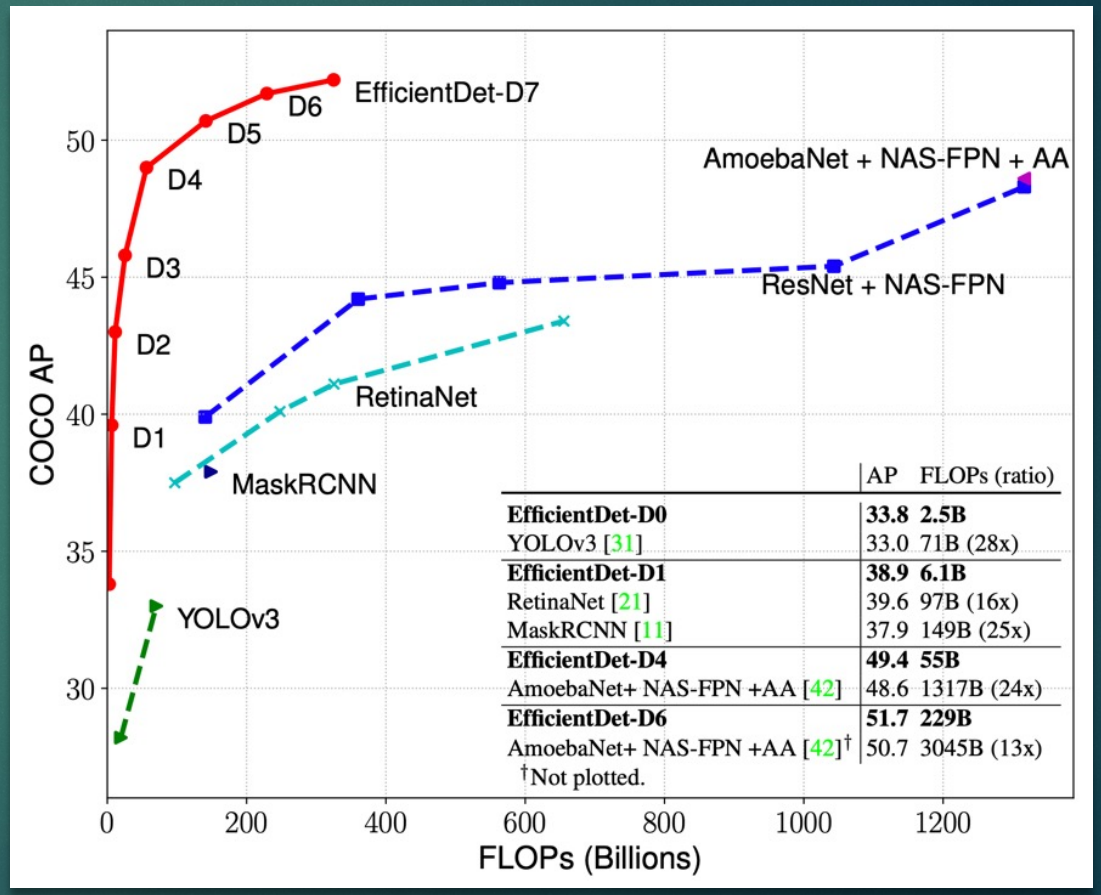
Validation



Test

Modeling

- EfficientDet model
- Developed by Google Brain team
- Built on top of EfficientNet
- Available in Tensorflow 2 Object Dection API
- There around 40 models
- Pretrained on COCO 2017 dataset

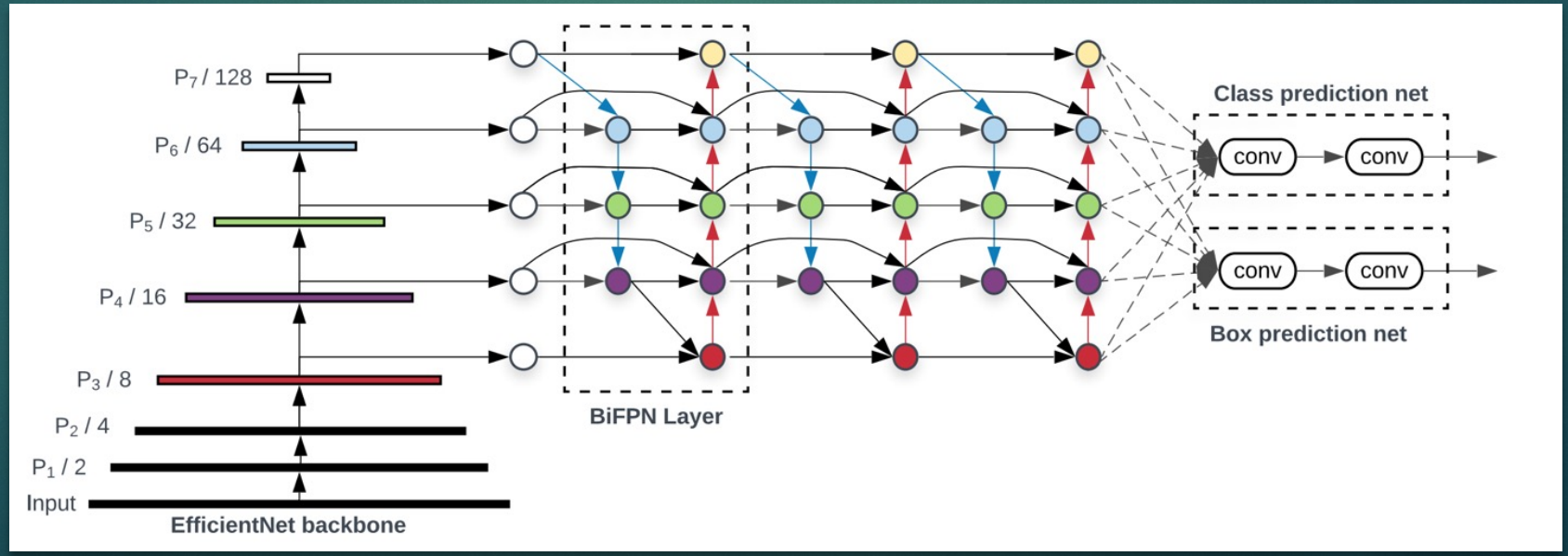


	AP	FLOPs (ratio)
EfficientDet-D0	**33.8**	**2.5B**
YOLOv3 [31]	33.0	71B (28x)
EfficientDet-D1	**38.9**	**6.1B**
RetinaNet [21]	39.6	97B (16x)
MaskRCNN [11]	37.9	149B (25x)
EfficientDet-D4	**49.4**	**55B**
AmoebaNet+ NAS-FPN +AA [42]	48.6	1317B (24x)
EfficientDet-D6	**51.7**	**229B**
AmoebaNet+ NAS-FPN +AA [42]†	50.7	3045B (13x)

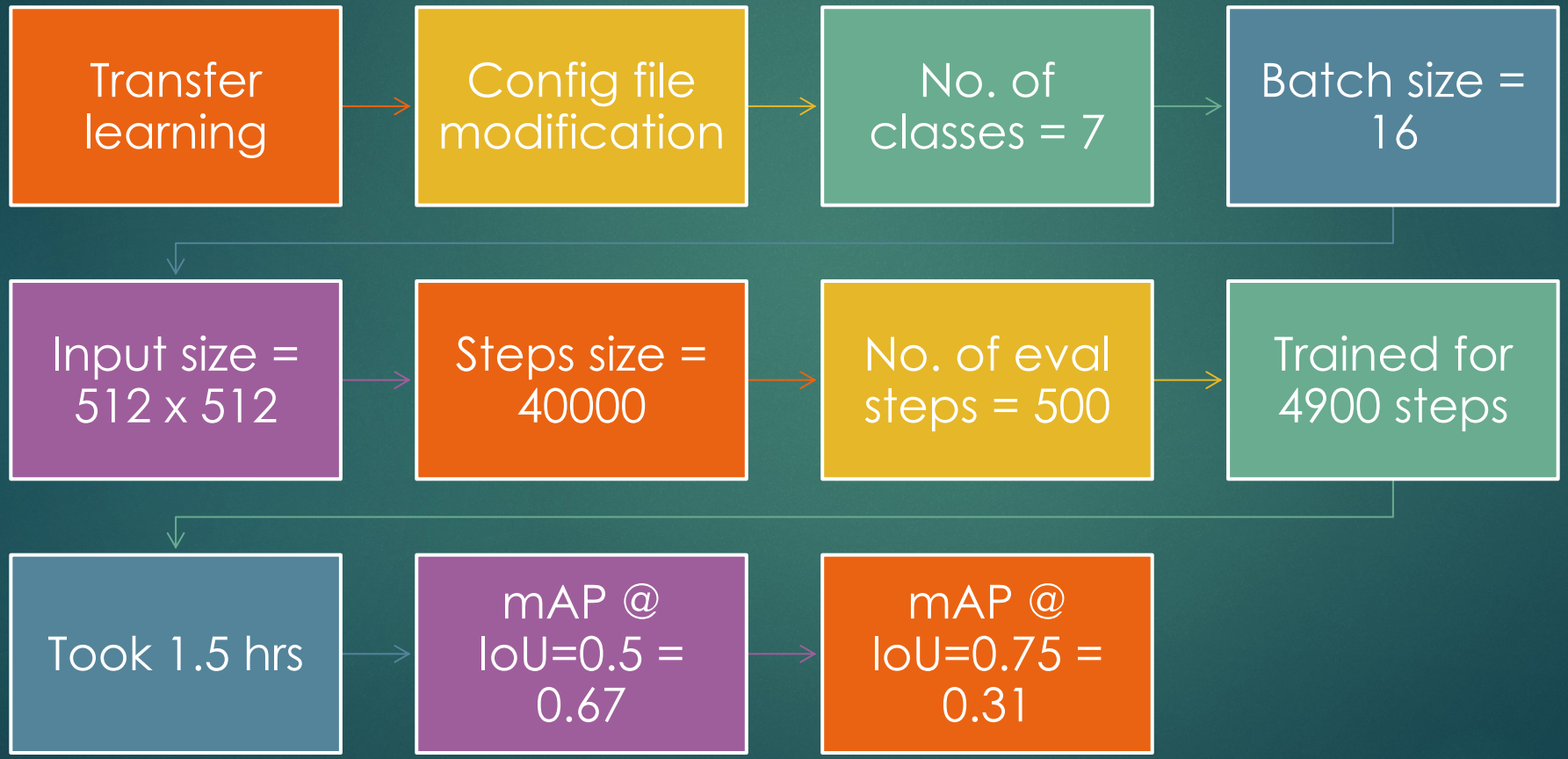
†Not plotted.

Modeling

- Pretrained ImageNet as backbone
- BiFPN as feature network
- Class and box network



Training

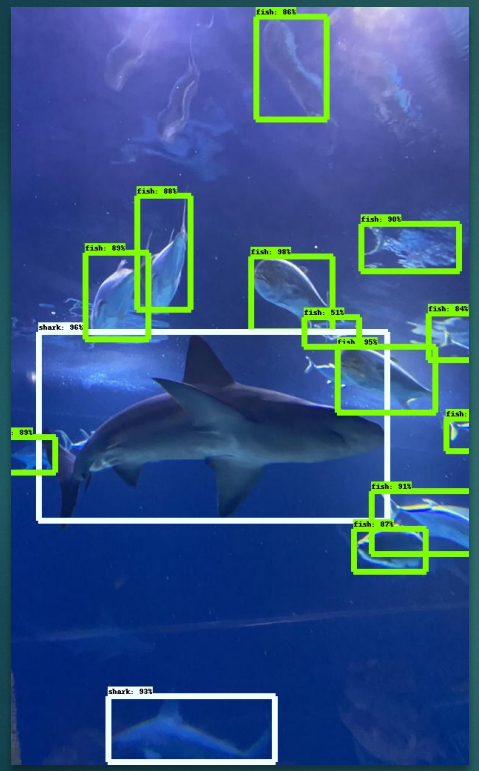

Transfer learning
Config file modification
No. of classes = 7
Batch size = 16
Input size = 512 x 512
Steps size = 40000
No. of eval steps = 500
Trained for 4900 steps
Took 1.5 hrs
mAP @ IoU=0.5 = 0.67
mAP @ IoU=0.75 = 0.31

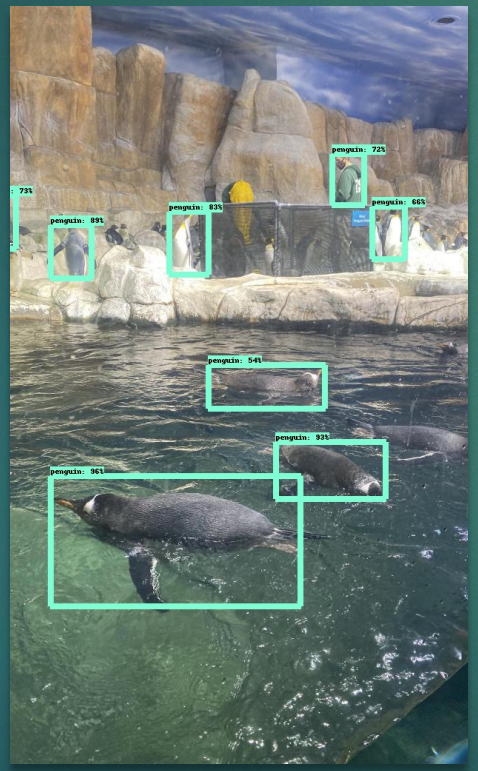
Evaluation

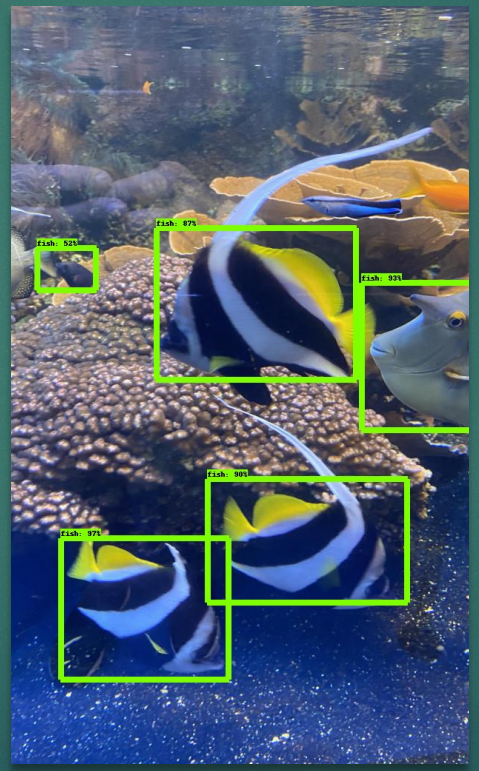
- Model was evaluated using test data
- IoU, Precision and Recall

category	precision_@0.5IOU	recall_@0.5IOU	F1
puffin	0.08	0.26	0.12
starfish	0.24	0.82	0.38
stingray	0.06	0.80	0.11
fish	0.05	0.44	0.08
shark	0.07	0.61	0.12
jellyfish	0.03	0.80	0.06
penguin	0.06	0.32	0.10

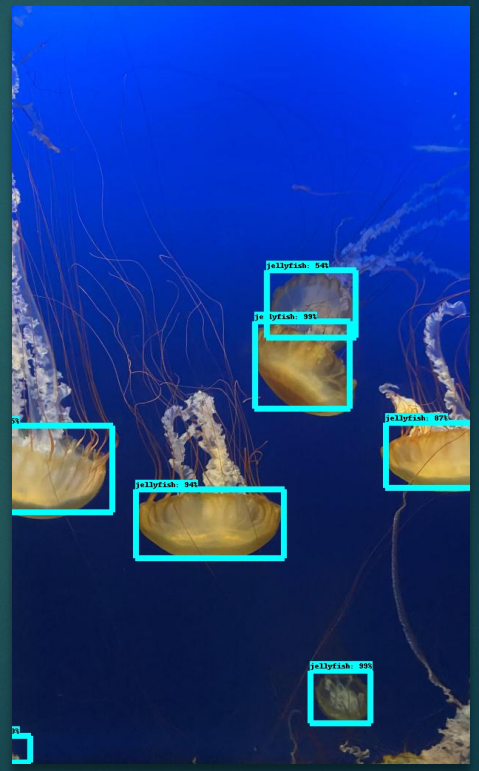
Results


fish: 86%
fish: 88%
fish: 89%
fish: 90%
fish: 98%
fish: 51%
fish: 84%
shark: 96%
fish: 95%
fish: 91%
fish: 87%
shark: 93%


penguin: 72%
penguin: 89%
penguin: 83%
penguin: 66%
penguin: 54%
penguin: 93%
penguin: 96%

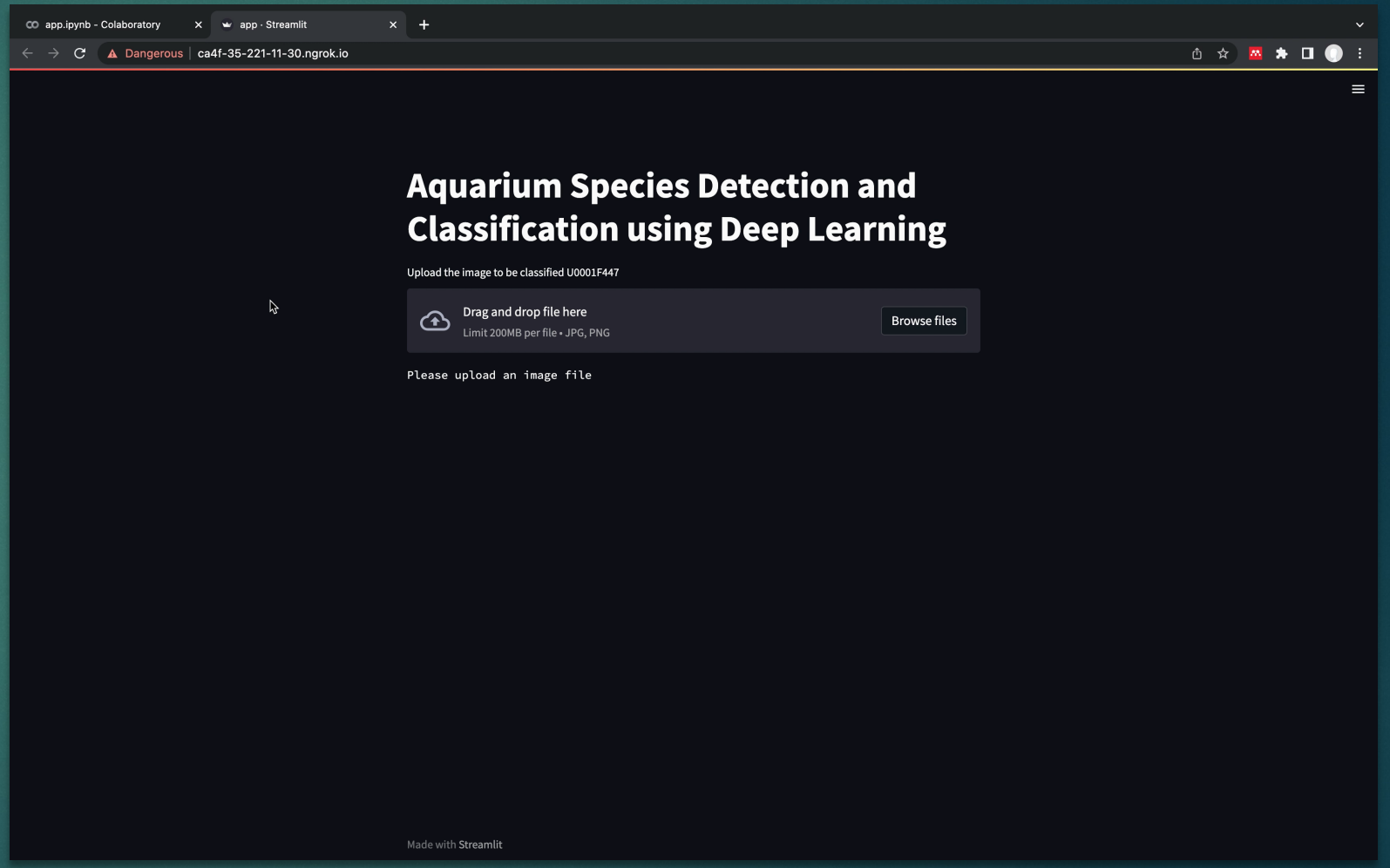

fish: 87%
fish: 52%
fish: 93%
fish: 90%
fish: 97%


jellyfish: 92%
jellyfish: 93%
jellyfish: 96%
jellyfish: 95%
jellyfish: 90%
jellyfis : 97%
jellyfish: 96%


jellyfish: 54%
jellyfish: 87%
jellyfish: 94%
jellyfish: 99%

Deployment

- Web-app made using Streamlit
- Upload an image and webapp outputs the detections



Conclusions

- Custom train deep learning model using TF2OD API
- Detection and classification of aquatic species
- Model has low precision
- Takes longer to load on the website