Water Pollution Detection using sensing and computer vision.

Guide : Dr Raja Vara Prasad

K.Sravani (S20170020216)
T.Chandana (S20170020246)

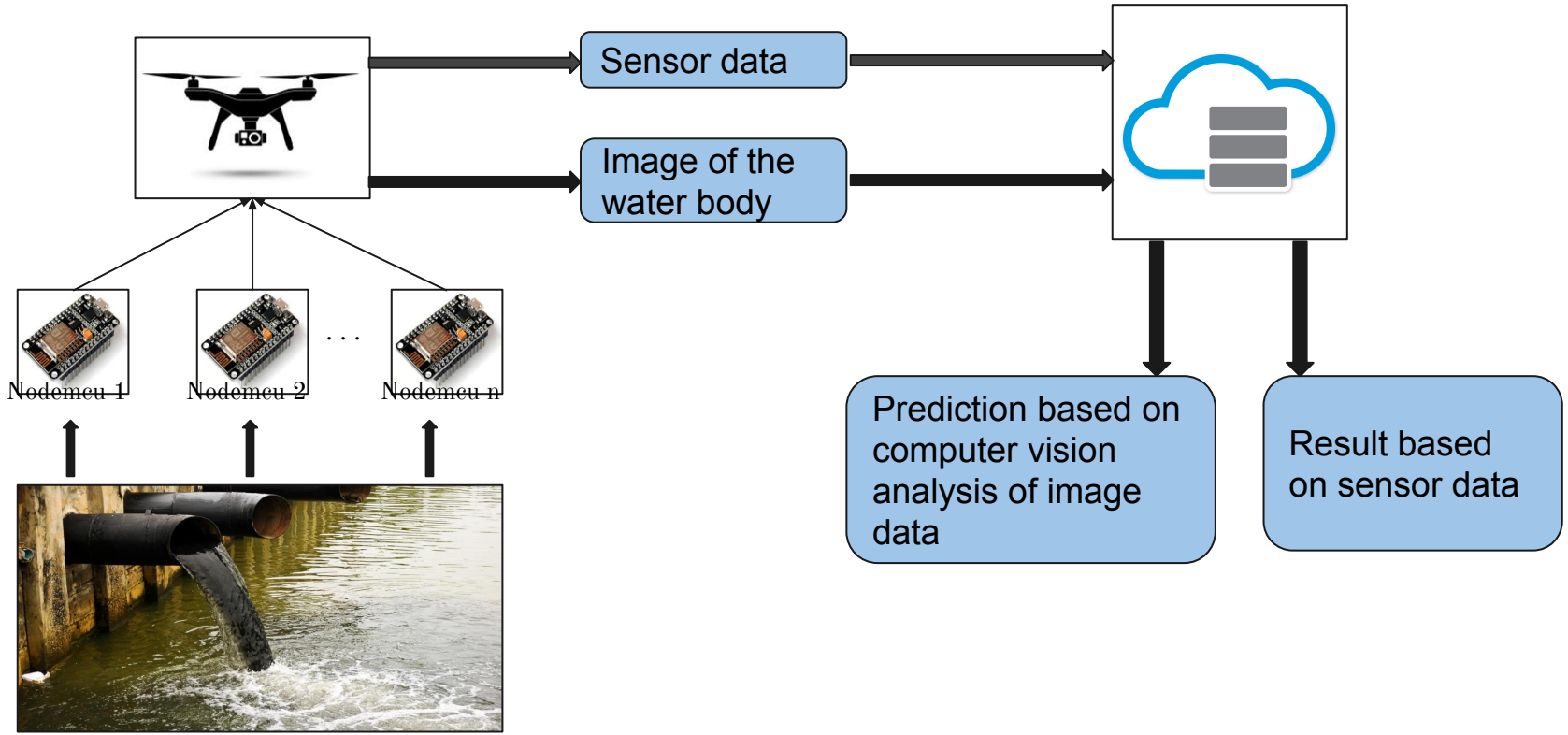
Overview

- Objective
- Workflow
- Literature Review
- Previous work
- Work done
- Results
- Challenges
- References

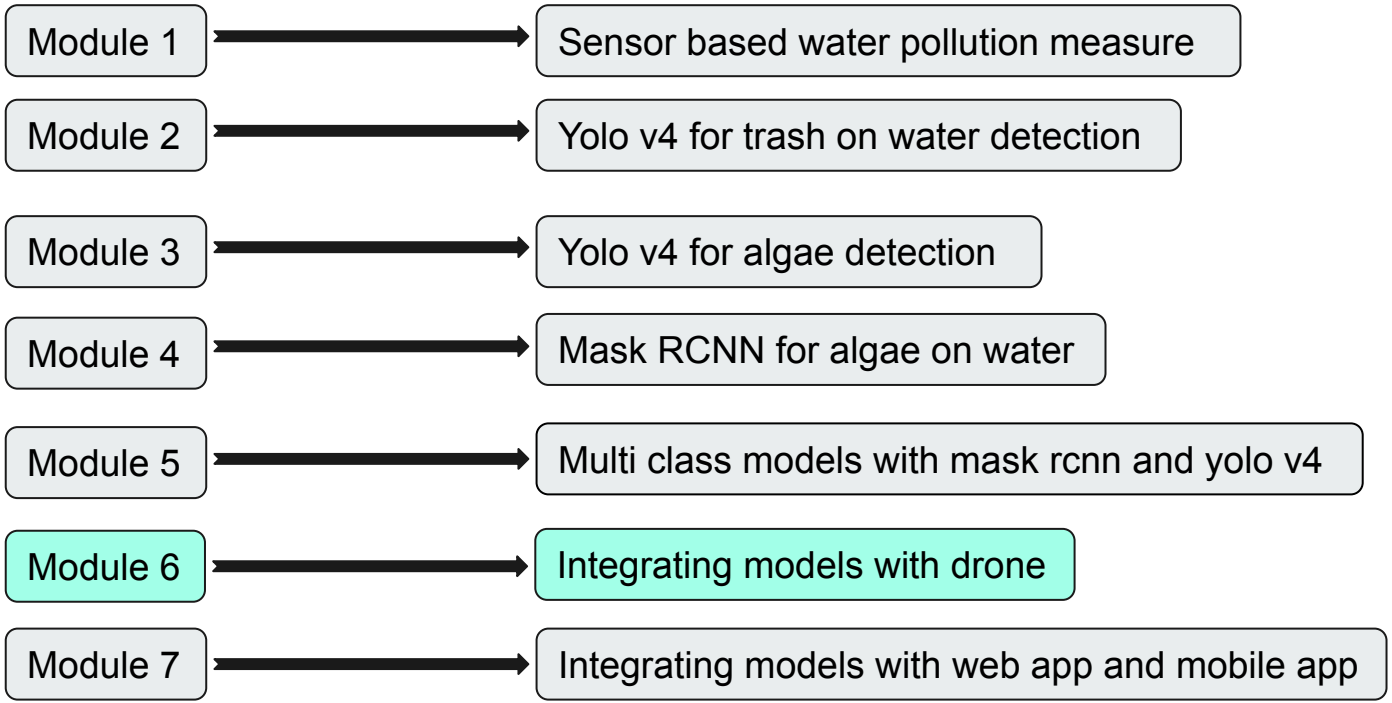
Objective

- To efficiently predict if the water body is polluted or not with the help of sensor data collected by placing sensors on water.
- To take a picture of the water body with the help of drone and to predict if it is polluted or not by running a computer vision model.

Workflow


Sensor data
Image of the water body
Nodemcu 1
Nodemcu 2
. . .
Nodemcu n
Prediction based on computer vision analysis of image data
Result based on sensor data

Deliverables

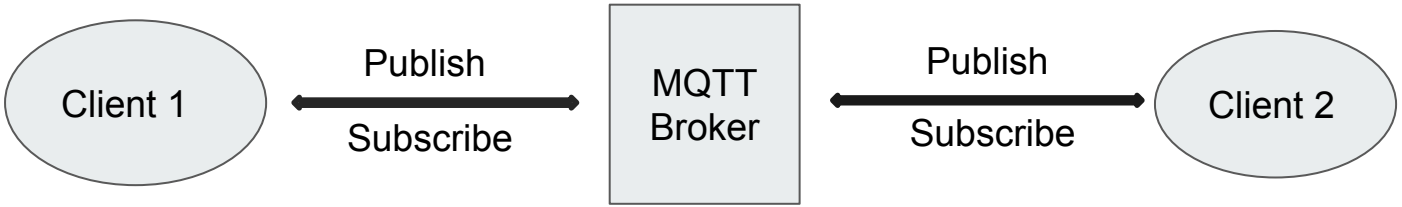


Literature Review

Protocol Explored:

Message Queueing Telemetry Transport(MQTT)

- Lightweight Publish and Subscribe Protocol.
- Designed for Constrained devices with low Bandwidth.
- Have different Quality of Service options.
- Both client and broker can publish and subscribe.



Literature Review

Object Detection

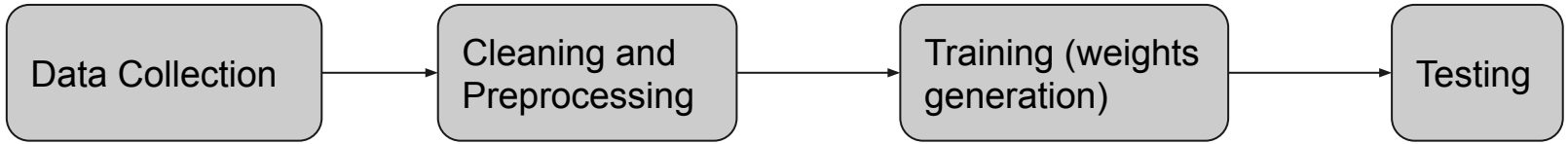
It has both Object Localization(Bounding boxes) and Class prediction(Label classification)
There are 3 well known object detectors:

1. RCNN' s and their variants (eg: Fast RCNN, Faster RCNN, Mask R-CNN).
2. YOLO-You only look once.
3. SSD-Single shot detector.

Process of object detection

There are three steps in any object detection task :

- Data Collection and labelling (with LabelImg).
- Prepare the dataset for training.
- Generate the weight files and use this ‘.h5’ or ‘.weights’ file for prediction on new image.

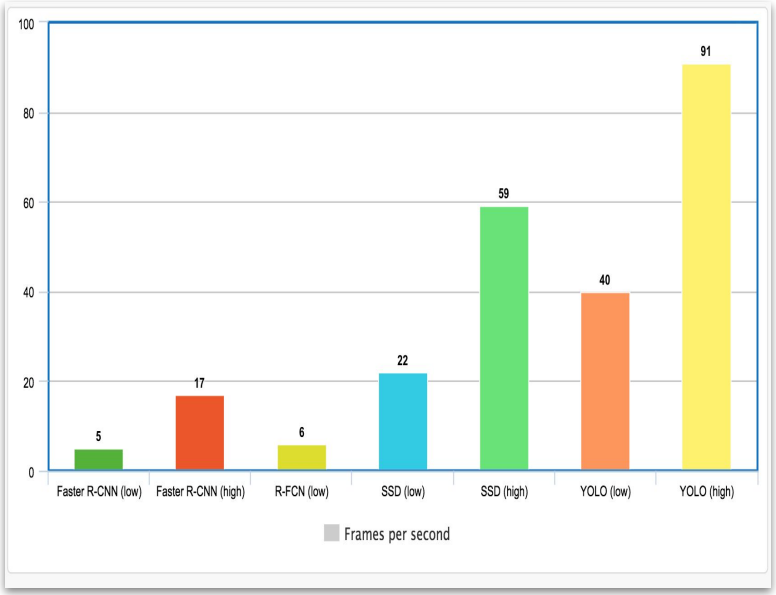


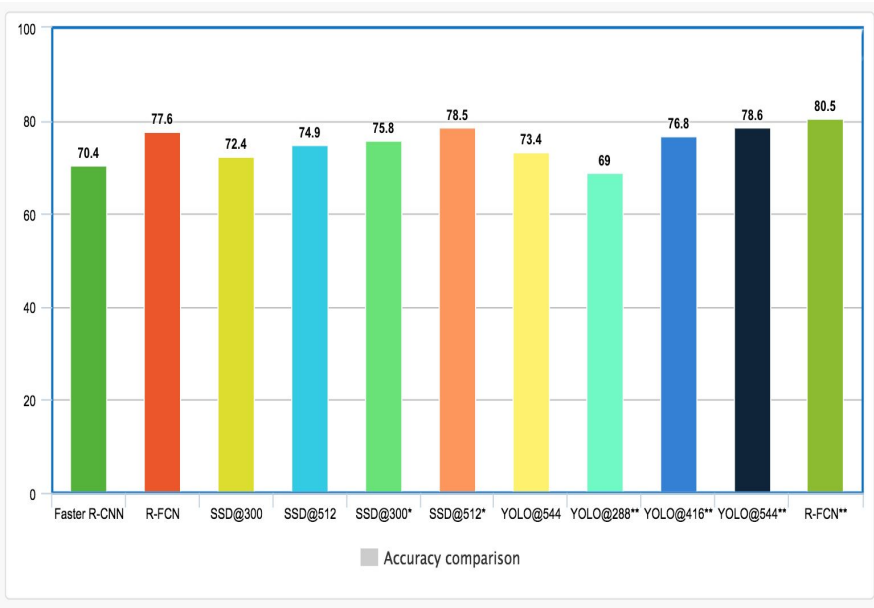
Literature Review(contd..)

YOLO object Detector:

- You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.
- YOLO v4 claims to have state-of-the-art accuracy while maintains a high processing frame rate. It achieves an accuracy of 43.5% AP (65.7% AP_{50}) for the MS COCO with an approximately 65 FPS inference speed on Tesla V100.

Speed Comparison Vs Accuracy Comparison





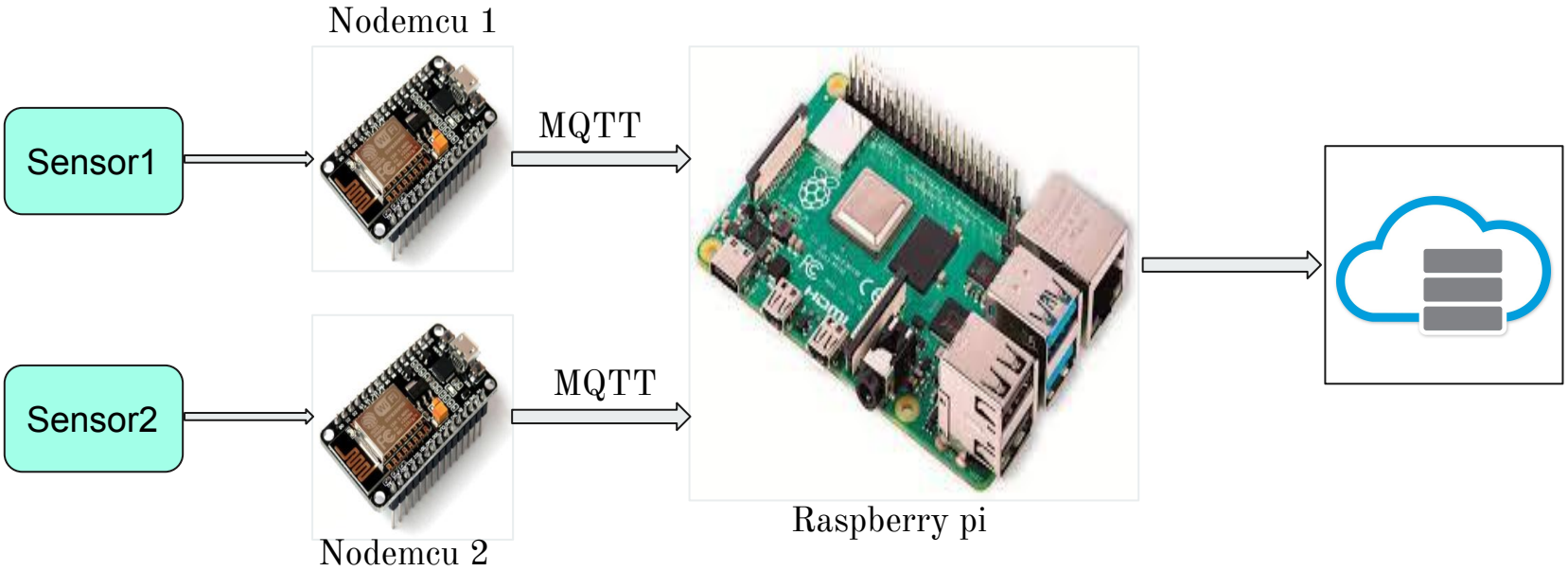
Mask R-CNN

- Mask R-CNN is a state-of-the-art framework for Image Segmentation tasks.
- The Mask R-CNN framework is built on top of Faster R-CNN.
- So, for a given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.

Previous work

- ➔ Data is collected from 2 different sensors through NodeMcu and it is sent to gateway node(raspberry pi).
- ➔ The protocol through which the data is sent from end nodes to gateway node is MQTT(Message Queueing Telemetry Transport).
- ➔ Multithreading is used to send data from multiple clients simultaneously without collision.
- ➔ Network Topology : STAR
- ➔ Data from this gateway is sent and saved in firebase realtime database.

Previous Work(Contd..)



Sensor data

Sensor Name	***Acceptable range***
Conductivity sensor	300 to 800 μS/cm
PH sensor	6.5 to 8.5
Turbidity sensor	≤ 5 NTU
Dissolved oxygen	≥ 8 mg/l

Previous work

Object detection models:

Data Collection

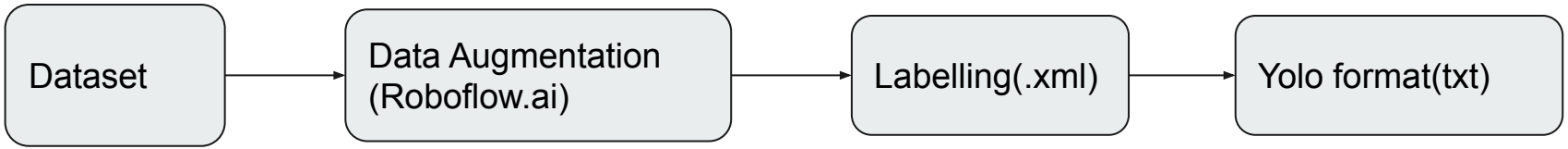
- ➔ Didn’t have proper labelled dataset available
- ➔ Did web scraping from different stock image sites like shutterstock, istock and collected images of algae and trash on water.
- ➔ Collected 200 images of water with trash on its surface and 200 images of algae on water.

Previous Work(Contd..)

Data labelling and preprocessing:

Since the dataset is very small, we tripled the dataset using data augmentation techniques like noise, crop, flip on roboflow.ai.

Labelled these 1200 images in voc format(.xml) using a tool called LabelImg.



Sample images

Trash on water:

Sample images:

Algae water

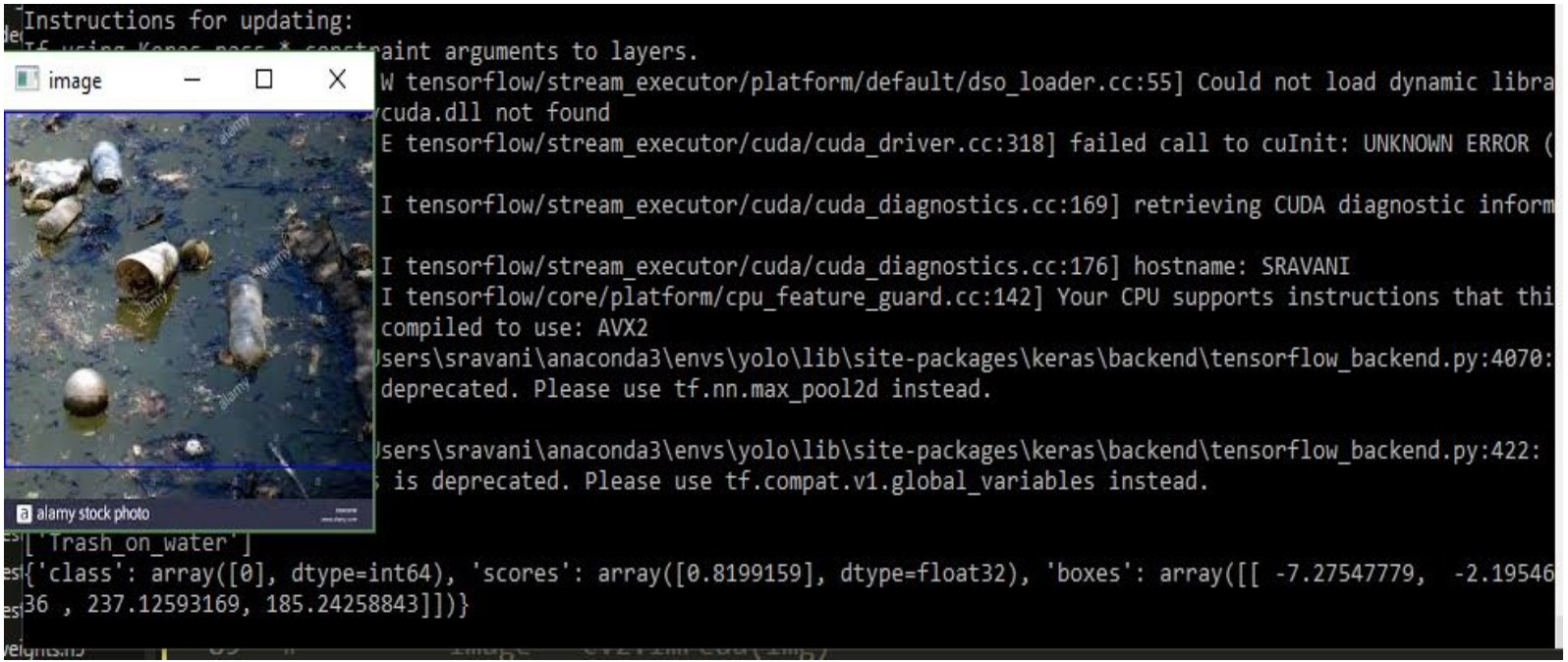
Previous Work(contd..)

Model 1 : Trash on water detection

Transfer learning on darknet model(darkent.conv.137) with YOLO V4.

- **Train test split**: 90% training data,10%test data
- Generated train.txt and test.txt files that contains the paths to all the images of training set and test set respectively.
- **Classes:** Trash_on_water(single class)
- **Iterations**: 5000
- **Batch size:** 32
- **Mean Average Precision(mAp)** : 70%

Results


Instructions for updating:
raint arguments to layers.
W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
cuda.dll not found
E tensorflow/stream_executor/cuda/cuda_driver.cc:318] failed call to cuInit: UNKNOWN ERROR (
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic inform
I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: SRAVANI
I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that thi
compiled to use: AVX2
sers\sravani\anaconda3\envs\yolo\lib\site-packages\keras\backend\tensorflow_backend.py:4070:
deprecated. Please use tf.nn.max_pool2d instead.
sers\sravani\anaconda3\envs\yolo\lib\site-packages\keras\backend\tensorflow_backend.py:422:
is deprecated. Please use tf.compat.v1.global_variables instead.
image
alamy stock photo
['Trash_on_water']
{'class': array([0], dtype=int64), 'scores': array([0.8199159], dtype=float32), 'boxes': array([[-7.27547779, -2.19546
36 , 237.12593169, 185.24258843]])}

Previous Work(contd..)

Model 2 : Algae Detection

Transfer learning on darknet model(darkent.conv.137) with YOLO V4.

- **Train test split**: 90% training data,10%test data.
- Generated train.txt and test.txt files that contains the paths to all the images of training set and test set respectively.
- **Classes:** algae(single class)
- **Iterations**: 5000
- **Batch size:** 32
- **Mean Average Precision(mAp)** : 81.43

Results

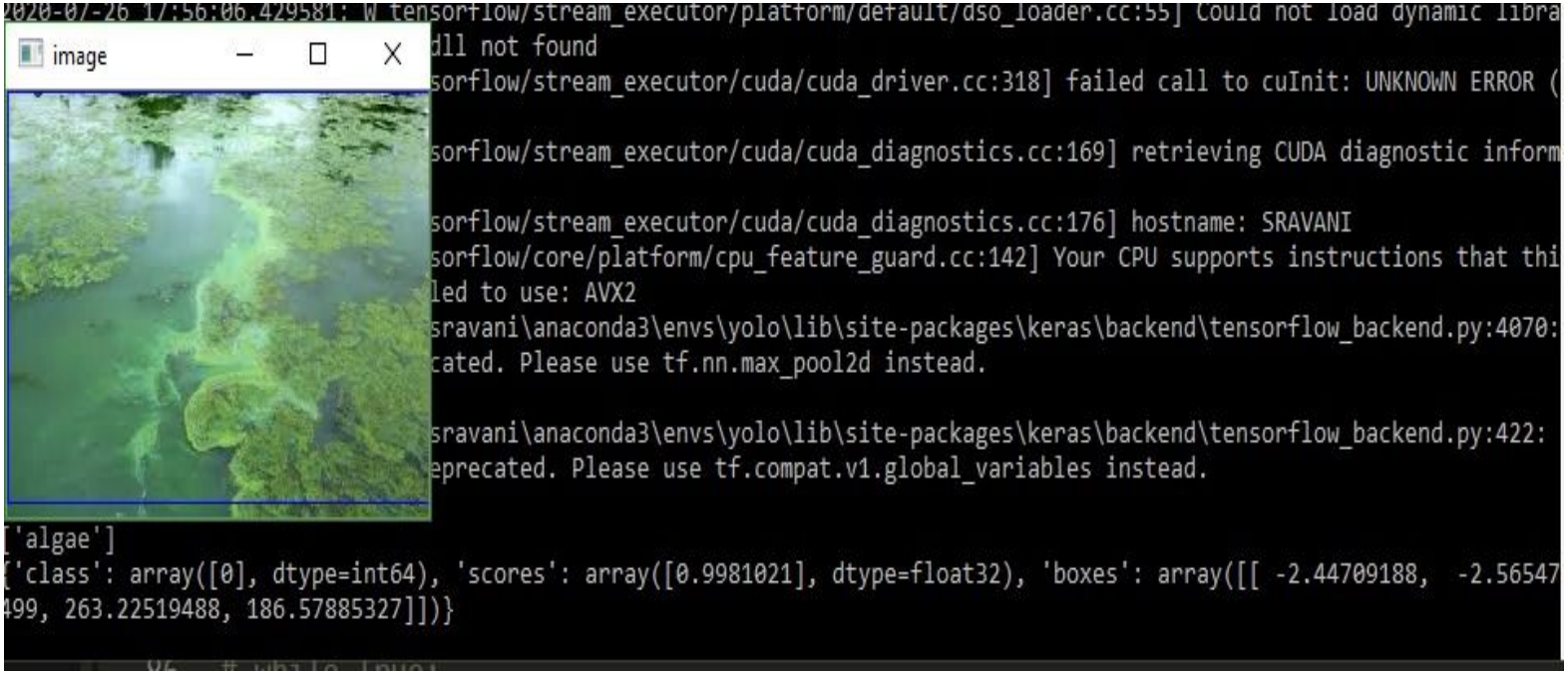


image
dll not found
sorflow/stream_executor/cuda/cuda_driver.cc:318] failed call to cuInit: UNKNOWN ERROR (
sorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic inform
sorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: SRAVANI
sorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that thi
led to use: AVX2
sravani\anaconda3\envs\yolo\lib\site-packages\keras\backend\tensorflow_backend.py:4070:
cated. Please use tf.nn.max_pool2d instead.
sravani\anaconda3\envs\yolo\lib\site-packages\keras\backend\tensorflow_backend.py:422:
eprecated. Please use tf.compat.v1.global_variables instead.
['algae']
{'class': array([0], dtype=int64), 'scores': array([0.9981021], dtype=float32), 'boxes': array([[-2.44709188, -2.56547
499, 263.22519488, 186.57885327]])}

Previous Work(Contd..)

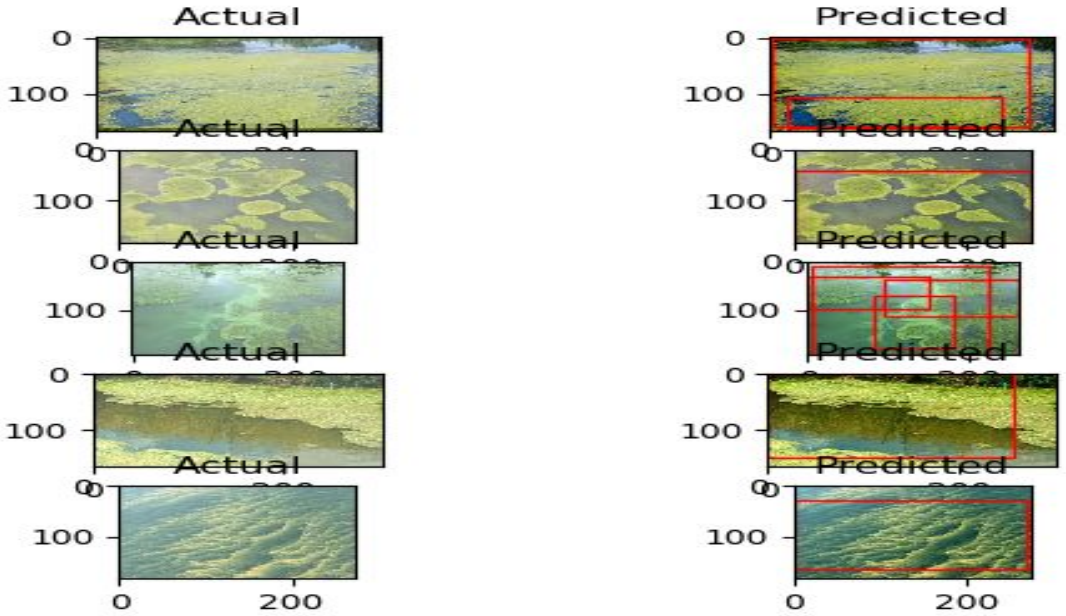
Model 3 : Algae on water detection

Transfer learning with pre trained MS COCO as base model with mask RCNN.

Initialized the last 3 layers weights of the pretrained MASK RCNN COCO model and retrained the last three layers with a new labelled dataset of algae.

- Training images:600
- Batch Size:40
- Epochs:50
- Train Accuracy: 87%
- Test Accuracy: 82.4%

Results

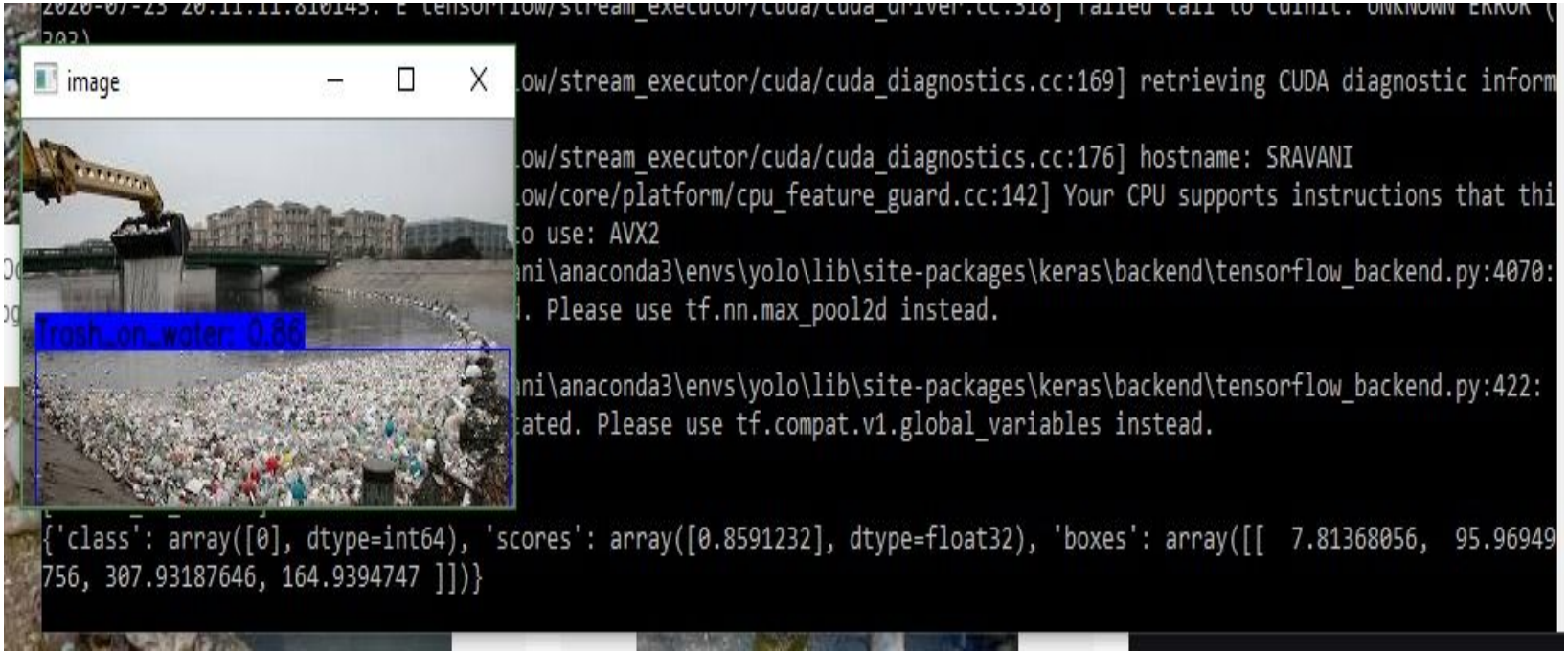


Previous Work (Multiclass Models)

Model 1: YOLO V4 with Darknet(multi class model)

- **Train test split**: 90% training data,10%test data
- Generated train.txt and test.txt files that contains the paths to all the images of training set and test set respectively.
- **Dataset size:** 1200
- **Classes:** algae,Trash_on_water(multi class)
- **Iterations**: 10,000
- **Batch size:** 32
- **Mean Average Precision(mAp)** : 80%

Results (Contd..)



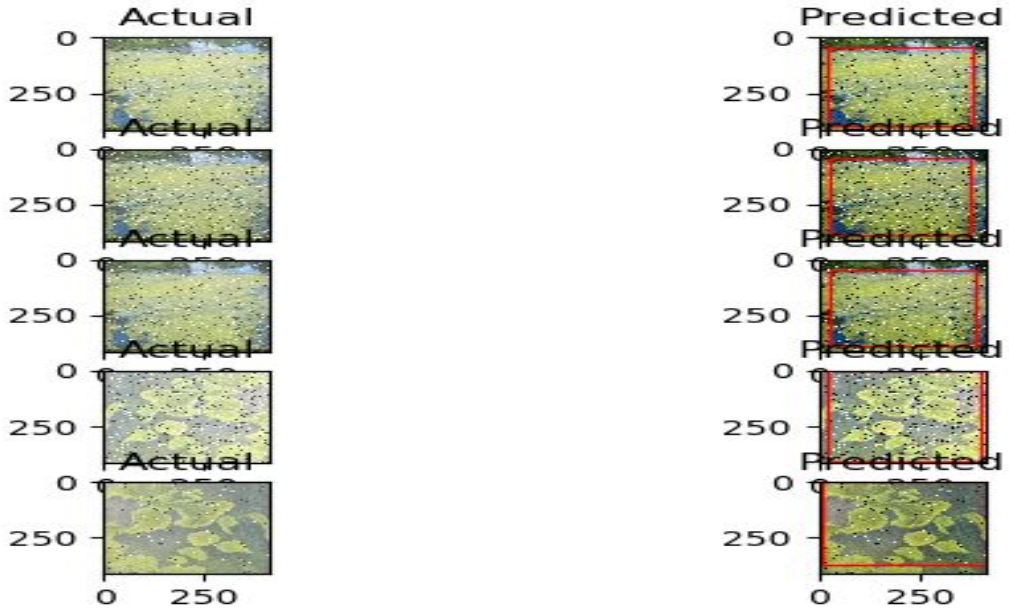
Results (Contd..)

Previous Work (Contd...)

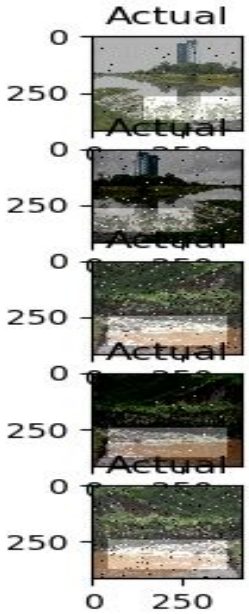
Model 2 : MASK RCNN with two classes.

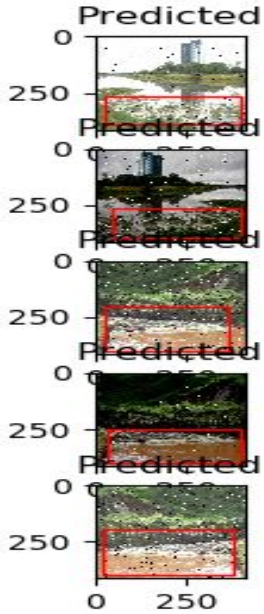
- **Train test split**: 90% training data,10%test data
- Generated dictionary of image data that contains the paths to all the images of training set and test set respectively by parsing xml files.
- Masked all the images with numpy.
- **Dataset size:** 1200
- **Classes:** algae,Trash_on_water(multi class)
- **Epochs**: 500
- **Batch size:** 32
- **Mean Average Precision(mAp)** : 92%

Results



Results(contd..)





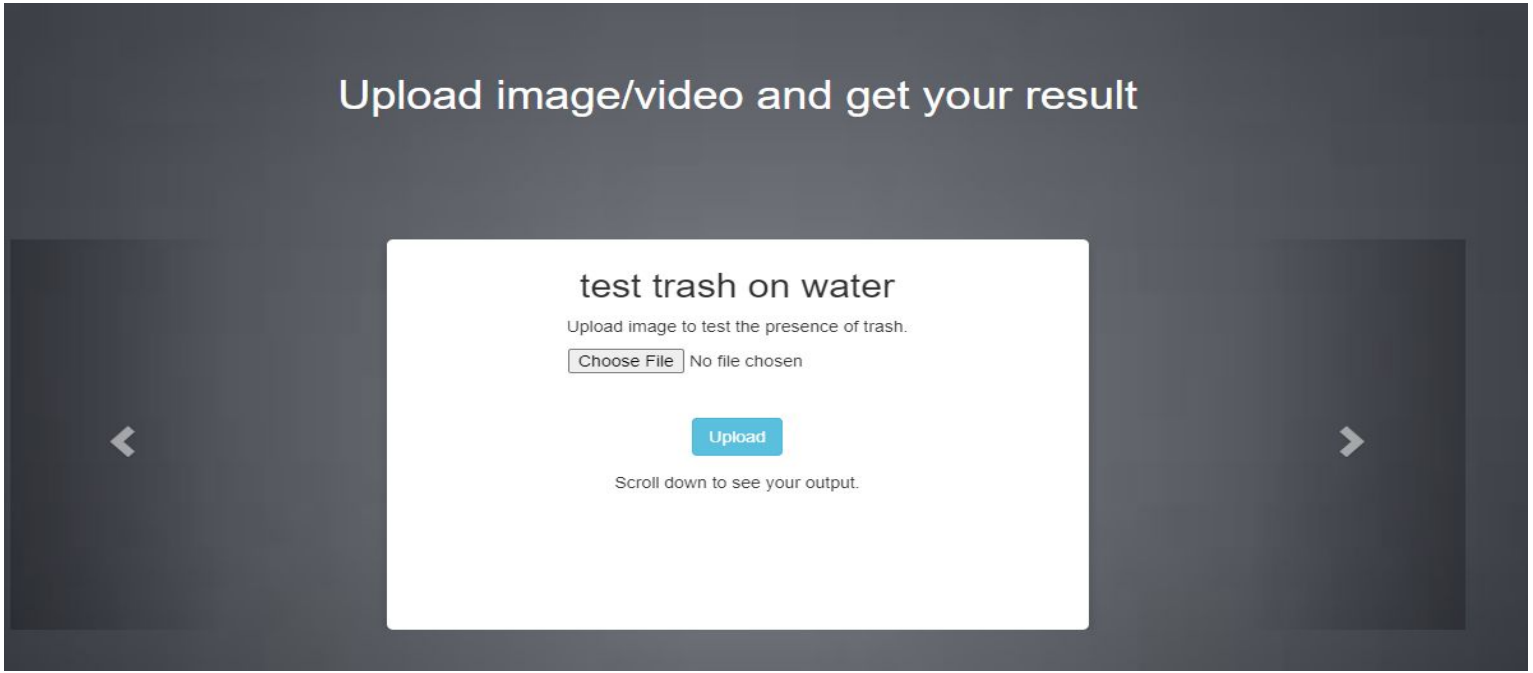
Integrated the models to Flask Api

- Built a basic flask web app
- Constructed api’s to integrate these models to flask app

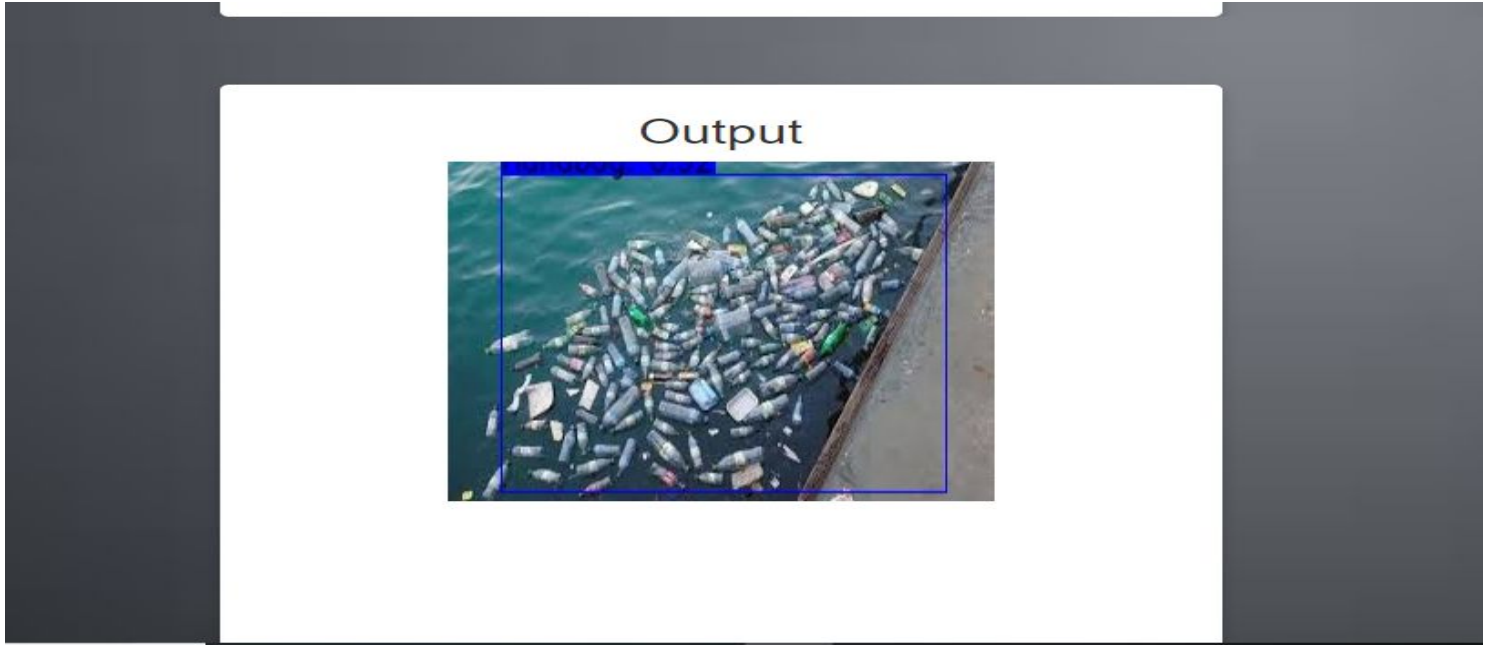
Requirements for testing :

- Cudnn
- Python 3.6
- Opencv
- Keras
- Tensorflow.

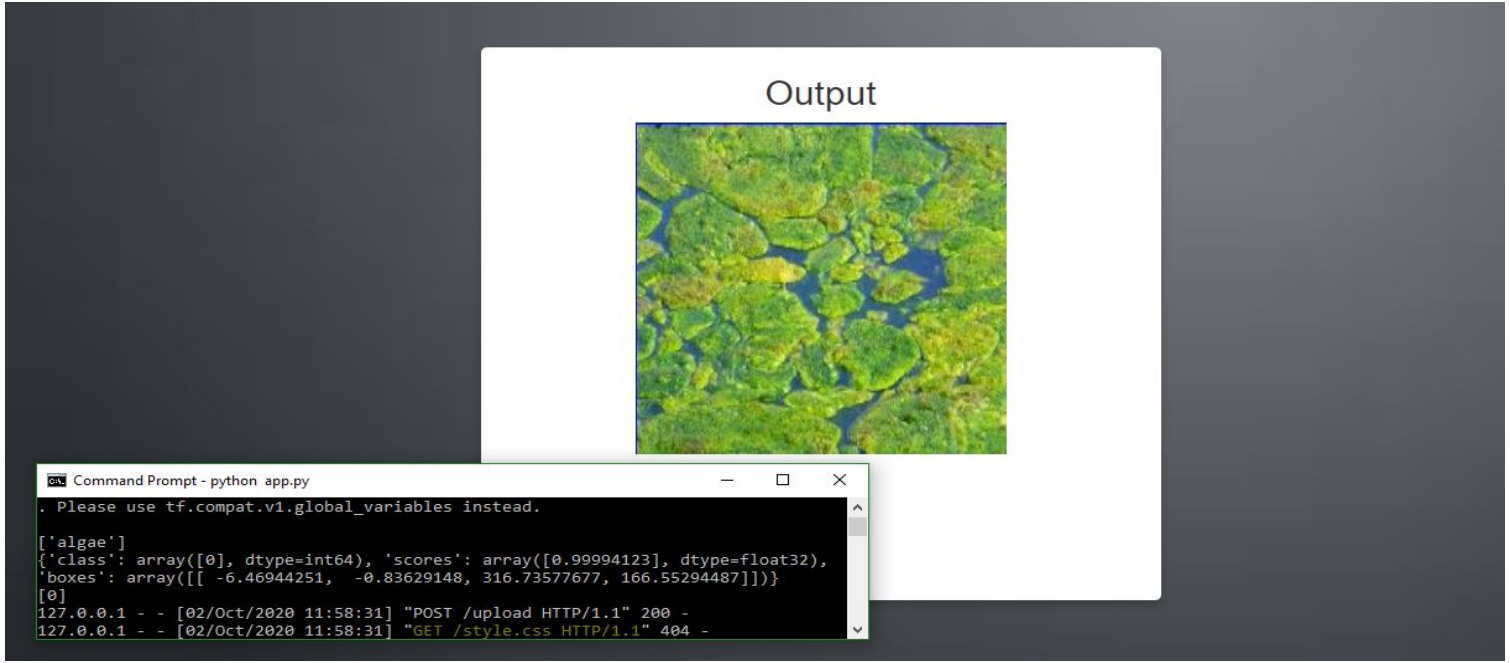
Results


Upload image/video and get your result
test trash on water
Upload image to test the presence of trash.
Choose File
No file chosen
Upload
Scroll down to see your output.

Results(Contd..)



Results(contd..)



Comparison and Analysis

Name	***Model***	***Class***	***Accuracy (mAp)***	***Time***
Trash on water	YOLO V4 Darknet	Single class	70	20.65sec
Algae detection	YOLO V4 Darknet	Single class	81.43	20.65 sec
Algae detection	Mask RCNN	Single class	82.4	1:10 sec
Algae and trash detection	YOLO V4 Darknet	Multiclass	80	21.65 sec
Algae and trash detection	Mask RCNN	Multiclass	92	1:10 sec

Work Done

YOLO V4- Tiny

In order to deploy the model in raspberry pi (lower end devices). We trained a lighter version model using YOLO V4 tiny.

We have to trade some accuracy in order to make the model lighter and for higher frame rate.

Work Done (Contd...)

Transfer learning on darknet model(darkent.conv.29) with YOLO V4 Tiny.

- **Train test split**: 90% training data,10%test data.
- Generated train.txt and test.txt files that contains the paths to all the images of training set and test set respectively.
- **Classes:** algae, trash on water(multi class)
- **Iterations**: 5000
- **Batch size:** 32
- **Mean Average Precision(mAp)** : 50.04

Work Done

Deployed the lighter version of the model in the mobile phone (android).

Developed the mobile app with flutter (android studio)

Steps :

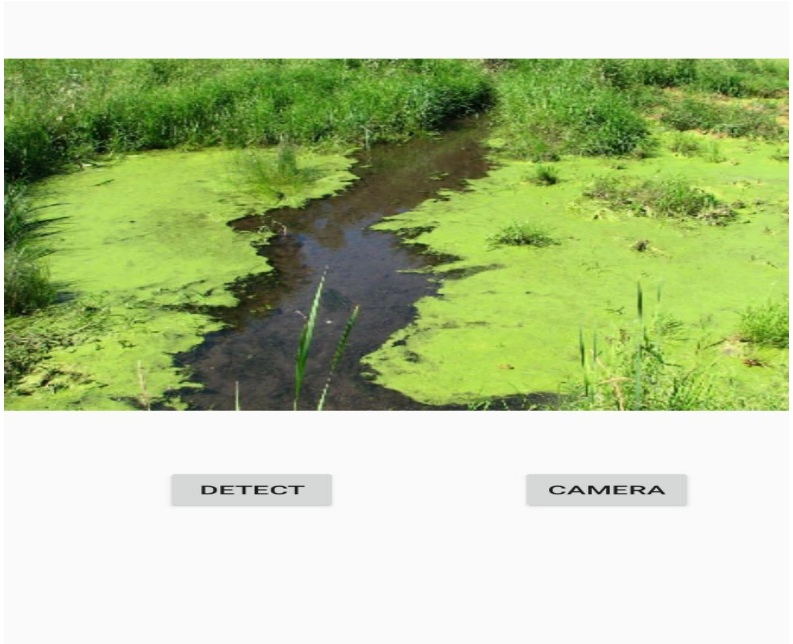
- Train the YOLO-Darknet model and save the weights.
- Convert Darknet Model to TensorFlow Lite
- Export Weights for Future Inference
- Deploy on Device

Work Done(cntd..)

Convert Darknet Model to TensorFlow Lite

- Darknet produces a .weights file for internal use in the Darknet framework. To use it with TensorFlow Lite, we need to convert it. For this, we used the tool TensorFlow-YOLOv4-TFLite.
- This tool uses the COCO dataset as a base, so we changed the classes to our custom ones.
- We converted the model from darknet to tensorflow SavedModel, then from TensorFlow Saved model to TensorFlow Lite, both with simple commands.

Results



Results (Contd..)

Challenges

- Difficulty in installing the required modules in raspberry pi due to lack of hardware.
- Pollution detection with image thresholding is not done due to lack of hardware(turbidity sensor) to prepare a dataset.
- Couldn’t integrate the models with drone due to lack of hardware.

References

[1] Samantaray, Arabinda & Yang, Baijian & Dietz, J. & Min, Byung-Cheol. (2018). Algae Detection Using Computer Vision and Deep Learning.

[2] U. Shafi, R. Mumtaz, H. Anwar, A. M. Qamar and H. Khurshid, "Surface Water Pollution Detection using Internet of Things," 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, 2018, pp. 92-96.

References (Contd...)

[3] V. Karnawat and S. L. Patil, "Turbidity detection using image processing," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 1086-1089.

- https://pjreddie.com/darknet/yolo/

Thank You