




AI Visual Inspection system for road deflection (L3)

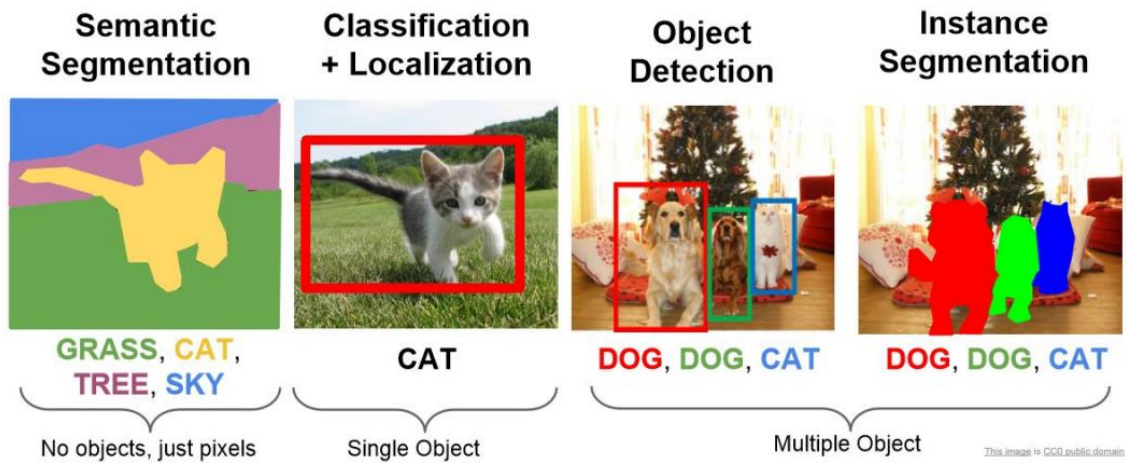
YOUTH COLLEGE (INTERNATIONAL)

Reference Websites

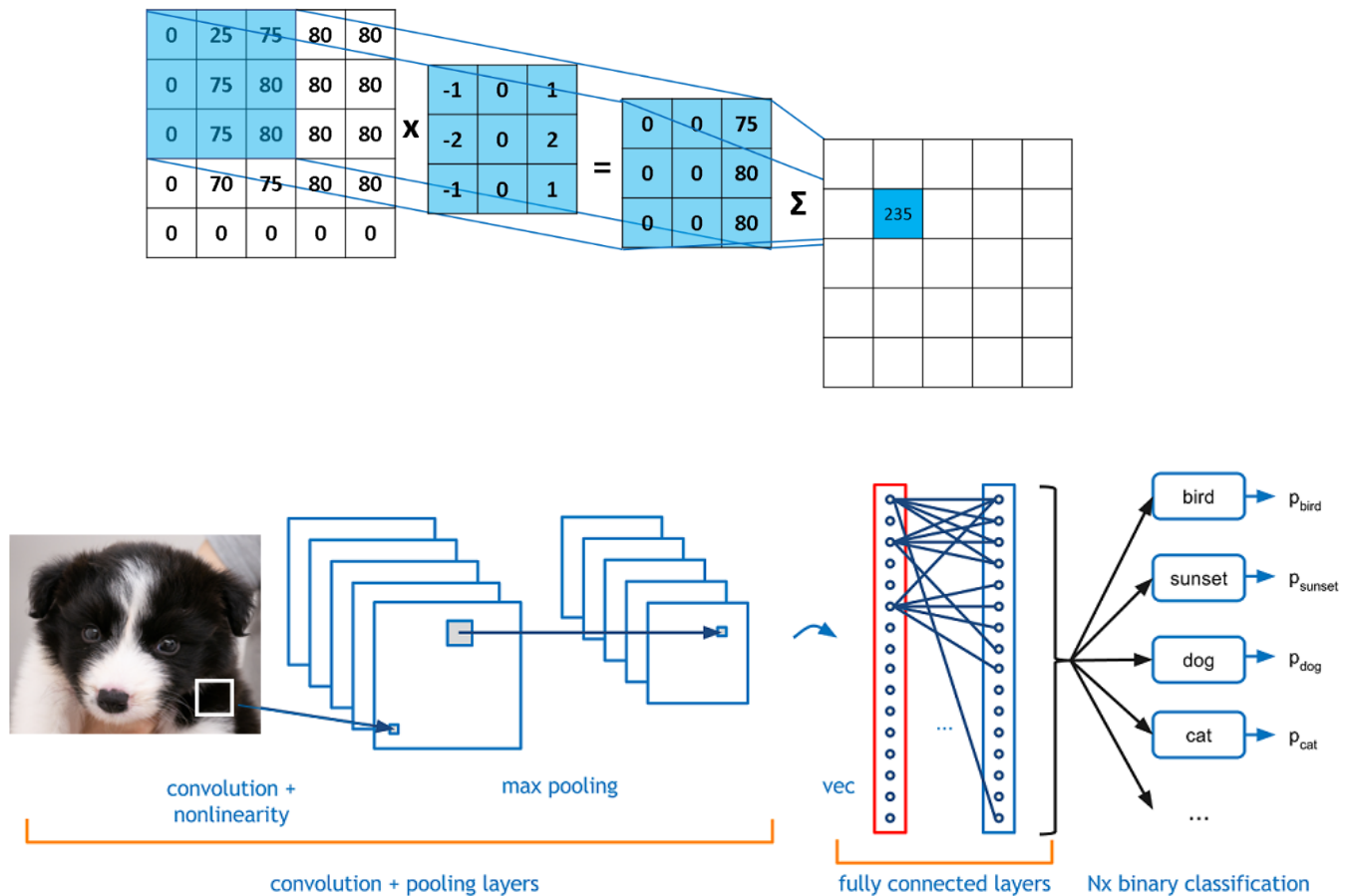
Python Official	
https://www.python.org/	
Google Colab	
https://colab.research.google.com/	
Python Exercises	
https://www.w3resource.com/python-exercises/	
https://www.w3schools.com/python/default.asp	
GitHub	
https://github.com/garyprojects/road_detect	

1. Introduction

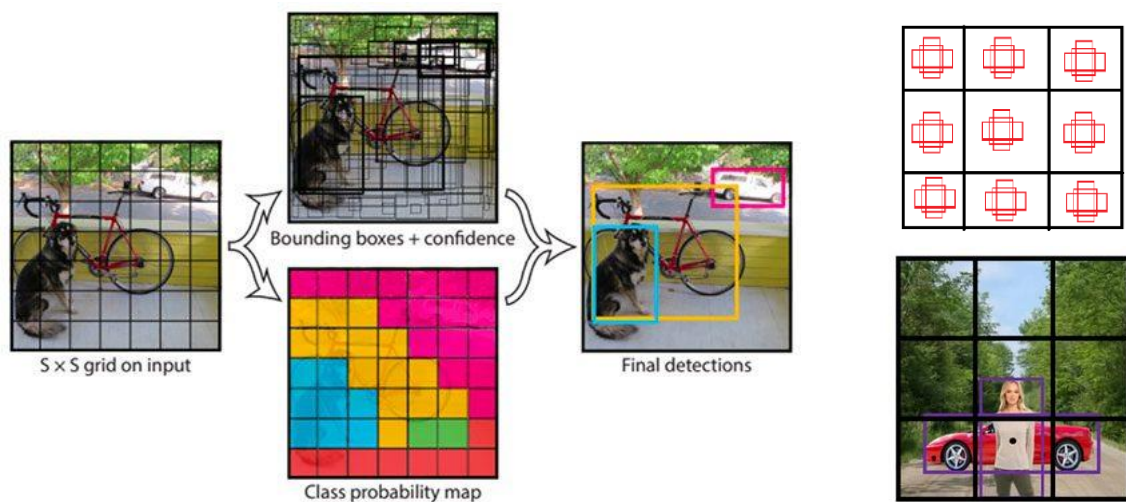
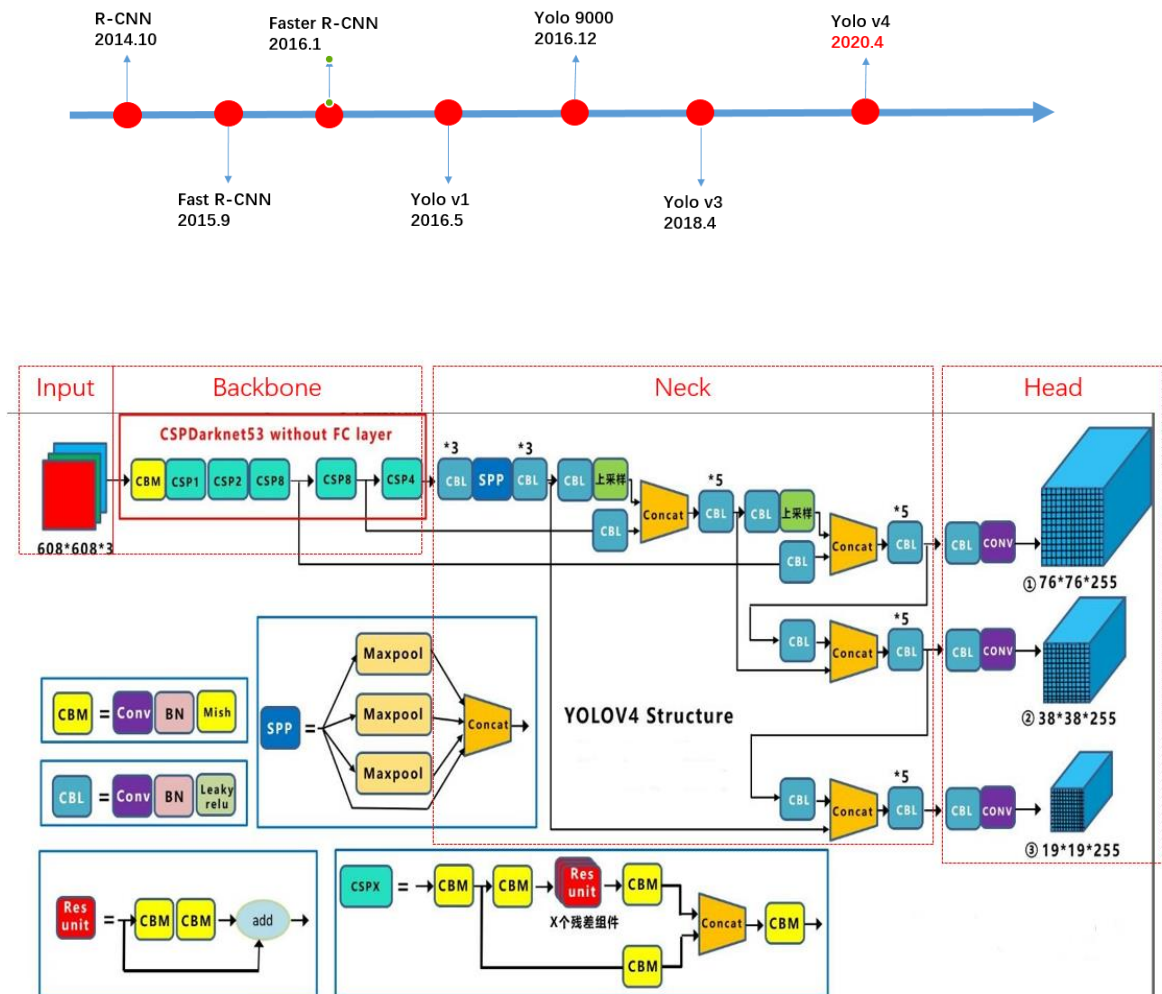
Computer Vision Tasks

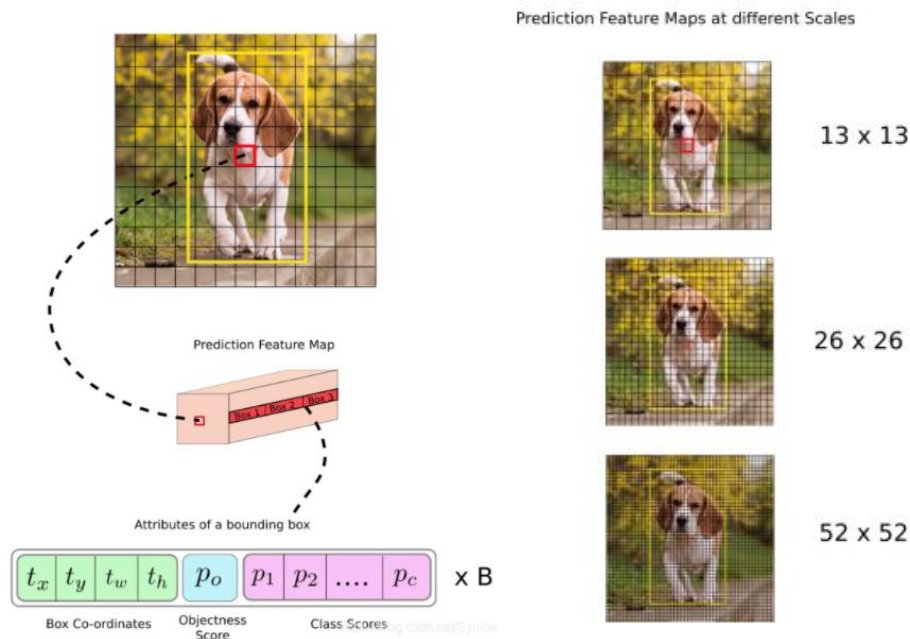


Convolutional Neural Network (CNN)



YoloV4





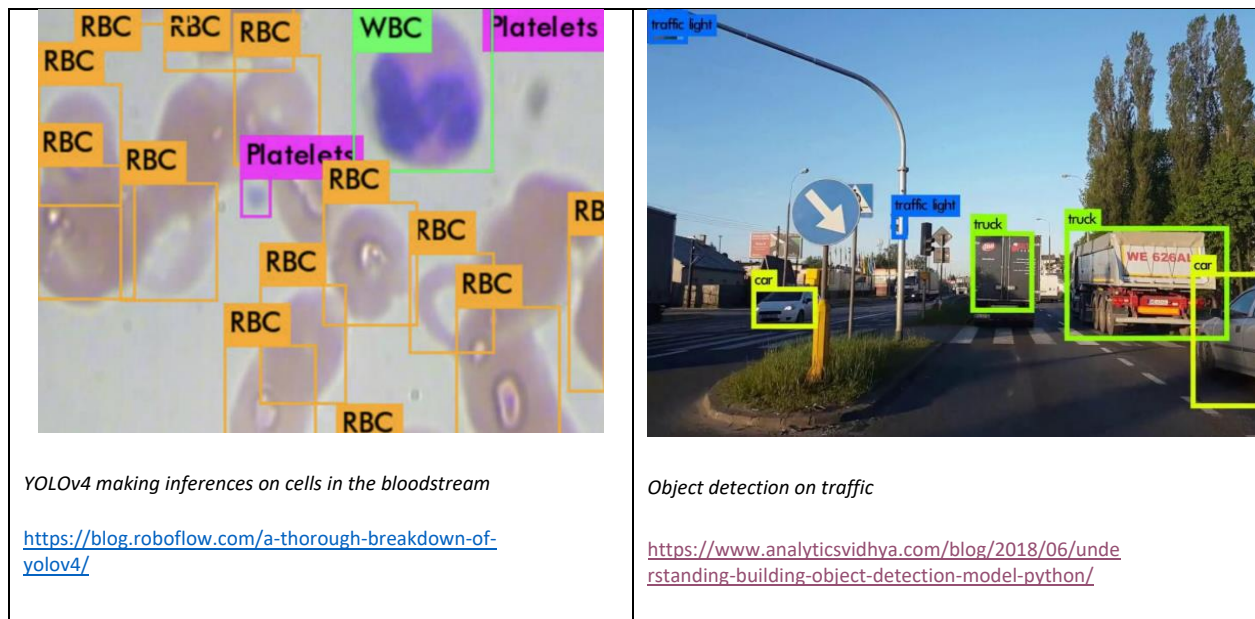
Reference blogs: <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>

<https://arxiv.org/pdf/2004.10934.pdf>

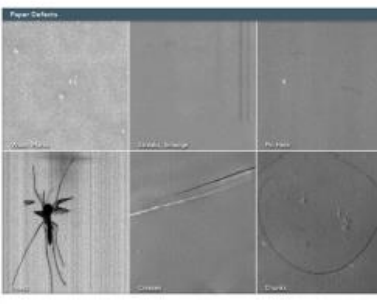
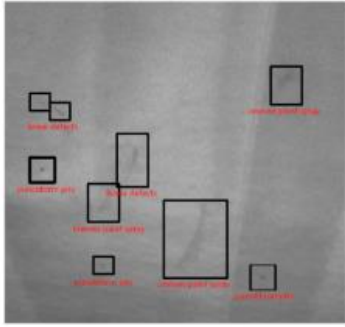
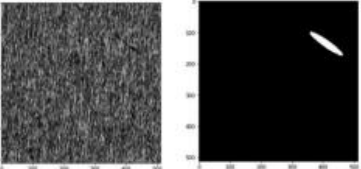
YoloV4 Keras Example: <https://paperswithcode.com/paper/yolov4-optimal-speed-and-accuracy-of-object>

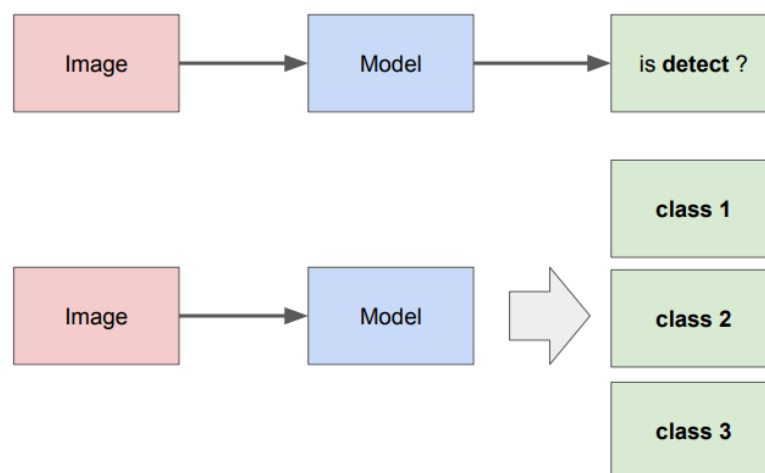
<https://github.com/AlexeyAB/darknet#yolo-v4-in-other-frameworks>

Examples of object detection

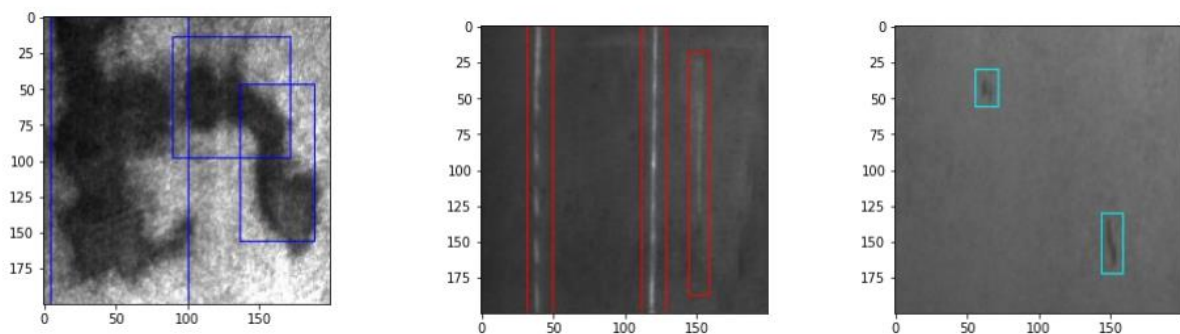


Defect Detection/ Anomaly Detection

		
Classification	Object detection	Segmentation



Examples of defect detection



Examples of Road Surface Defects

Potholes



Cracking



Decolored



I) Data preparation

Software: labelImg

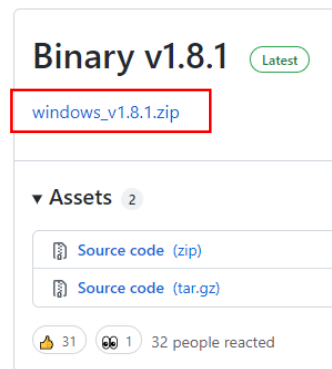
For windows:

windows_v1.8.1.zip

<https://github.com/tzutalin/labelImg/releases>

Others:

<https://github.com/wkentaro/labelme>



[./Lesson3_materials/dataset_all/xml/decolord](#)

File structure of the dataset

名稱	類型
Test	檔案資料夾
Train	檔案資料夾
xml	檔案資料夾
names.txt	文字文件

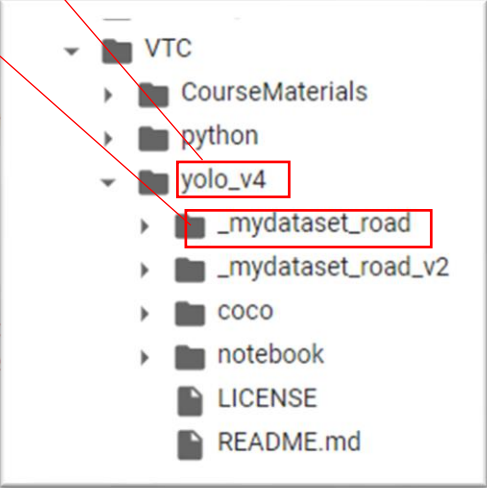
II) Training customized detector

File: yolo_detector.ipynb

```

1 # Your YOLO folder path in google drive
2 FOLDER_PATH = "/content/drive/MyDrive/VTC/yolo_v4"
3
4 MY_DATASET_PATH = "/content/drive/MyDrive/VTC/yolo_v4/_mydataset_road"
5
6 # Your XML Path (Output from LabelImg)
7 XML_PATH = os.path.join(MY_DATASET_PATH, 'xml')
8
9 # The path of labels for classification
10 CLASSES_PATH = os.path.join(MY_DATASET_PATH, 'names.txt')
11
12 # The output annotation.
13 TXT_PATH = os.path.join(MY_DATASET_PATH, 'anno.txt')
14
15 # Training and testing Image
16 IMG_FOLDER_PATH = os.path.join(MY_DATASET_PATH, 'Train')
17 TEST_FOLDER_PATH = os.path.join(MY_DATASET_PATH, 'Test')
18
19 # Set the number of epochs of the training
20 epochs = 1000
21 initial_epoch = 0
22
23 # Number of epochs with no improvement after which training will be stopped.
24 train_patience = 0.05 # 5% of epochs
25
26 # Train the model
27 model = YOLOv4.from_names(CLASSES_PATH, IMG_FOLDER_PATH, XML_PATH)
28 model.train(patience=train_patience, epochs=epochs, initial_epoch=initial_epoch)
29
30 # Save the model
31 model.save('yolo_v4_model.pth')
32
33 # Load the model
34 model = YOLOv4.from_names(CLASSES_PATH, IMG_FOLDER_PATH, XML_PATH)
35 model.load('yolo_v4_model.pth')
36
37 # Test the model
38 model.test(IMG_FOLDER_PATH, TEST_FOLDER_PATH, XML_PATH)
39
40 # Save the test results
41 model.save('yolo_v4_test_results.txt')
42
43 # Load the test results
44 model.load('yolo_v4_test_results.txt')
45
46 # Print the test results
47 print(model.test_results)
48
49 # Close the model
50 model.close()
51
52 # End of the script
53

```



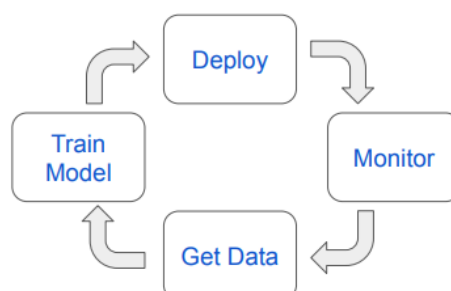
```

1 # Set the number of epochs of the training
2 epochs = 1000
3 initial_epoch = 0
4
5 # Number of epochs with no improvement after which training will be stopped.
6 train_patience = 0.05 # 5% of epochs
7

```

Tips for training model:

- As correct as possible
- Even Google can't do it 100%
- At least a clean and correct train/test dataset
- Clean train/test data to correctly show whether the model is progressing



III) Experimental Results

Decolored Dataset

```

Epoch 197/1000
4/4 [=====] - 7s 2s/step - loss: 63.1016 - val_loss: 593.5283
Epoch 198/1000
4/4 [=====] - 7s 2s/step - loss: 82.0183 - val_loss: 539.7248
Epoch 199/1000
4/4 [=====] - 7s 2s/step - loss: 74.4464 - val_loss: 517.0244
Epoch 200/1000
4/4 [=====] - 7s 2s/step - loss: 63.2695 - val_loss: 517.1257
Epoch 201/1000
4/4 [=====] - 7s 2s/step - loss: 52.7205 - val_loss: 529.2762
***
Training Completed! ... saving model

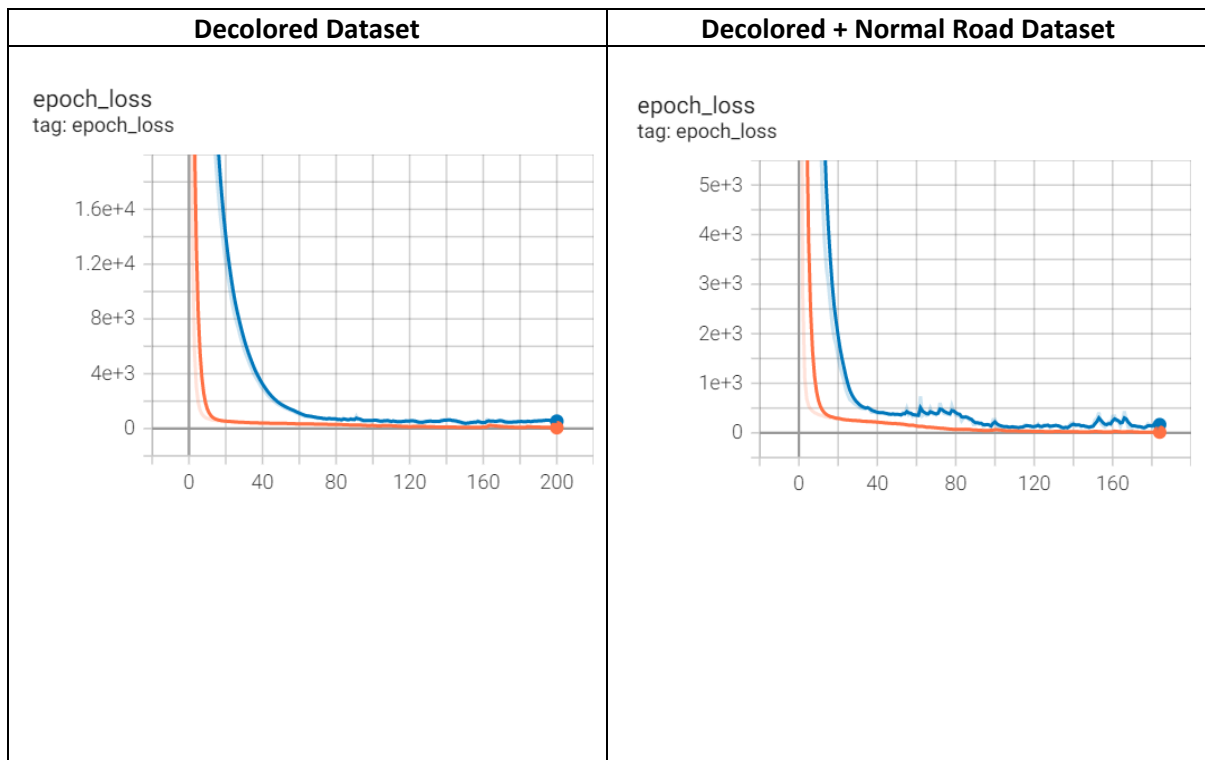
```

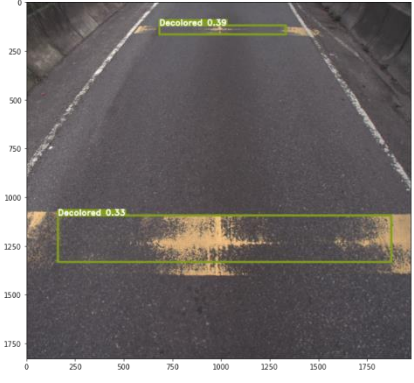
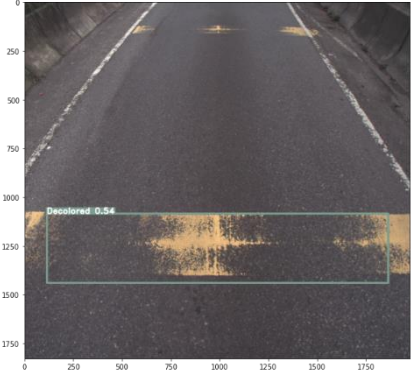
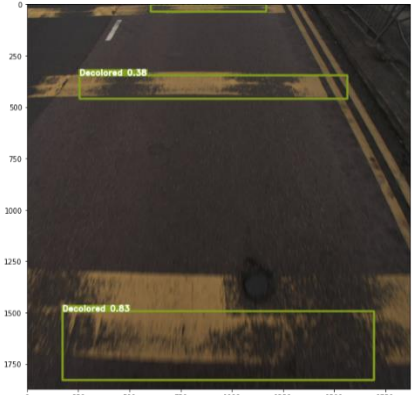
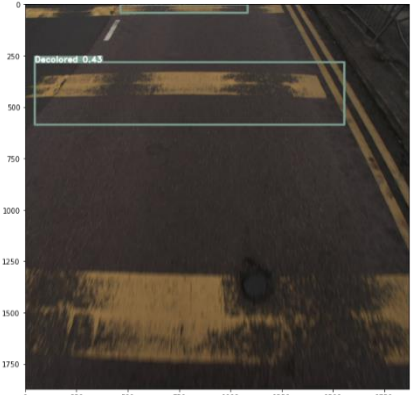
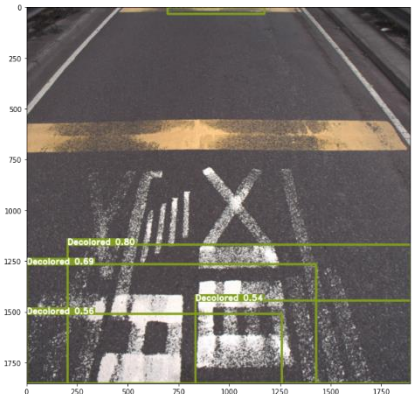
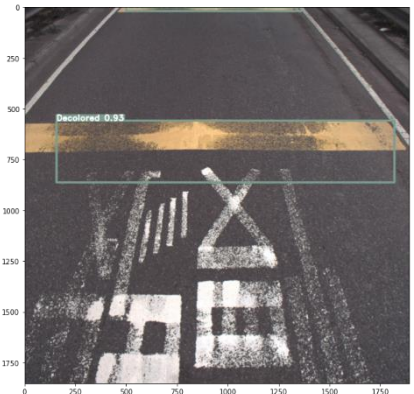
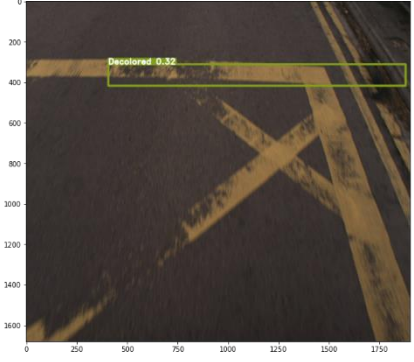
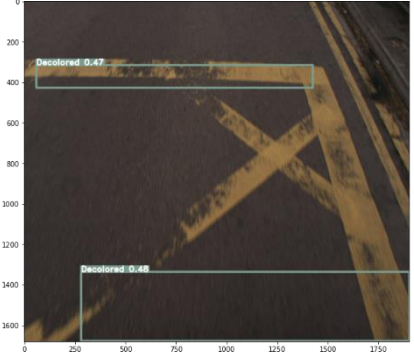
Decolored + Normal Road Dataset

```

Epoch 181/1000
8/8 [=====] - 12s 1s/step - loss: 11.4681 - val_loss: 127.4318
Epoch 182/1000
8/8 [=====] - 12s 1s/step - loss: 12.1996 - val_loss: 200.9842
Epoch 183/1000
8/8 [=====] - 12s 1s/step - loss: 11.2288 - val_loss: 199.6996
Epoch 184/1000
8/8 [=====] - 12s 2s/step - loss: 14.9343 - val_loss: 113.4704
Epoch 185/1000
8/8 [=====] - 12s 1s/step - loss: 14.7769 - val_loss: 187.5705
Training Completed! ... saving model

```



Decolored Dataset	Decolored + Normal Road Dataset
 <p>Decolored 0.39</p> <p>Decolored 0.33</p>	 <p>Decolored 0.54</p>
 <p>Decolored 0.39</p> <p>Decolored 0.33</p>	 <p>Decolored 0.43</p>
 <p>Decolored 0.80</p> <p>Decolored 0.69</p> <p>Decolored 0.55</p> <p>Decolored 0.54</p>	 <p>Decolored 0.93</p>
 <p>Decolored 0.32</p>	 <p>Decolored 0.47</p> <p>Decolored 0.48</p>