**Obstacle Detection Algorithm (Python Code) :**

from imageai.Detection import ObjectDetection

import os

import sys

import tensorflow as tf

import matplotlib.pyplot as plt

tf.logging.set\_verbosity(tf.logging.ERROR)

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

execution\_path = os.getcwd()

detector = ObjectDetection()

detector.setModelTypeAsRetinaNet()

detector.setModelPath( os.path.join(execution\_path ,"model.h5"))

detector.loadModel()

left = list()

height = list()

label = list()

inputimage = input("Enter the input image name : ")

if(os.path.exists(inputimage)):

pass

else:

print("Error!!!Image is not found")

sys.exit(0)

outputimage = input("Enter the output image name to be created : ")

if(os.path.exists(outputimage)):

print("Error!!!Image alredy present")

sys.exit(0)

else:

detections = detector.detectObjectsFromImage(input\_image=os.path.join(execution\_path , inputimage), output\_image\_path=os.path.join(execution\_path , outputimage))

i=0

for eachObject in detections:

label.append(eachObject["name"])

height.append(eachObject["percentage\_probability"])

left.append(i)

i+=1

print(eachObject["name"] , " : " , eachObject["percentage\_probability"] )

plt.plot(height,color="blue")

plt.xlabel('Detected Objects')

plt.ylabel('Percentage of correctness in detection')

plt.xticks(rotation=90)

plt.xticks(left,label)

plt.title('Object Detection Chart')

plt.savefig('graph'+os.path.splitext(outputimage)[0]+'.jpg',bbox\_inches='tight',pad\_inches=1)