Project Report On

Triple Ride Detection

Submitted By

C Praveen Kumar Reddy (R170128),

D Suryaprakash Reddy (R170393),

S Challa Rao (R170131),

P Venkat Sumanth(R171231).

Under The Guidance of

Ms. P. Udaya Sree

Assistant Professor

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the report entitled "Triple Ride Detection" submitted by C. Praveen Kumar Reddy (R170128), S. Challa Rao (R170131), D. Suryaprakash Reddy (R170393), G.Venkata Sumanth (R171231) in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science Engineering is a bonafide work carried out by her under supervision and guidance.

The report hasn't been submitted previously in part or in full to this or any other university or institution for the award of any degree. Under the Guidance of P Udaya Sree (Assistant Professor, Computer Science &Engineering, RGUKT, R.K Valley).

GUIDE

P. Udaya Sree

HEAD OF THE DEPARTMENT

P. Harinadha

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution. I also express my sincere gratitude to our respected Head of the Department Mr. P. Harinadha for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Ms. P. Udaya Sree for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract

Violations in traffic laws are very common in a highly populated country like India. The accidents associated with these violations cause a huge loss to life and property. Since utilization of bikes is high, mishaps associated with bikes are additionally high contrasted with different vehicles.

To decrease the accident rate and traffic levels, strict implementation of the rules and continuous monitoring of the traffic is mandatory. The main objective of this work is to identify the Triple Riding. To detect the triple riders Deconvolutional neural network-based YOLO (You Only Look Once) algorithm is used. For detection of the number of persons riding a bike, the system classifies the vehicle as to the rule-breach vehicle or not. The surveillance cameras at traffic signals acting as the data collections center, collects the data.

Introduction

All over the world around 1.35 million lives are lost each year, 50 million people are getting injured due to road accidents, according to a report titled " The Global status Revised Manuscript Received on December 05, 2019 report on road safety 2018" released by world health organization. It is very hard to imagine that this burden is unevenly borne by motorcyclists, cyclists, and pedestrians. This report noted that a comprehensive action plan must be set up in order to save lives.

Two-wheeler is a very popular mode of transportation in almost every country. However, there is a high risk involved because of less protection. When a two-wheeler meets with an accident, due of sudden deceleration, the rider is thrown away from the vehicle. If head strikes any object, motion of the head becomes zero, but with its own mass brain continues to be in motion until the object hits inner part of the skull. Sometimes this type of head injury may be fatal in nature. Most of the accidents are happening by not following the traffic rules like triple ride, helmetless driving and many more. So we come up with an idea to detect triple ride using CNN in python.

.

Purpose

In this Project Work, Triple Ride detection system is built which attempts to satisfy the automation of detecting the traffic violation. Implementation of rules and continuous monitoring of the traffic is mandatory in order to curtail the rate of accidents and traffic levels. Riders in India commit traffic offenses like riding without helmet, riding wrong way, parking in wrong zone, obstructing the free left on junctions. The gravest of these violations is triple riding. In triple riding, the vehicle is overloaded that leads to stress on the structural rigidity of the vehicle. The load displaces the center of gravity of the vehicle and it looses balance easily, with 3 people riding it. Triple riding violation is not only dangerous for the riders, it is menacing for the people around them too, and jeapordizes public safety.

Requirements

Hardware Configuration

Ram	512MB
Hard disk	10GB
Processor	1.0GHz
OS	Any version of ubuntu or windows

Software Requirements:

Python:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases

CV2:

Cv2 is a library of Python bindings designed to solve computer vision problems. cv2. imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

Shapely.geometry:

Shapely is a Python package for set-theoretic analysis and manipulation of planar features using (via Python's ctypes module) functions from the well known and widely deployed GEOS library.

System Implementation

One important element of deep learning and machine learning at large is dataset. A good dataset will contribute to a model with good precision and recall. In the realm of object detection in images or motion pictures.

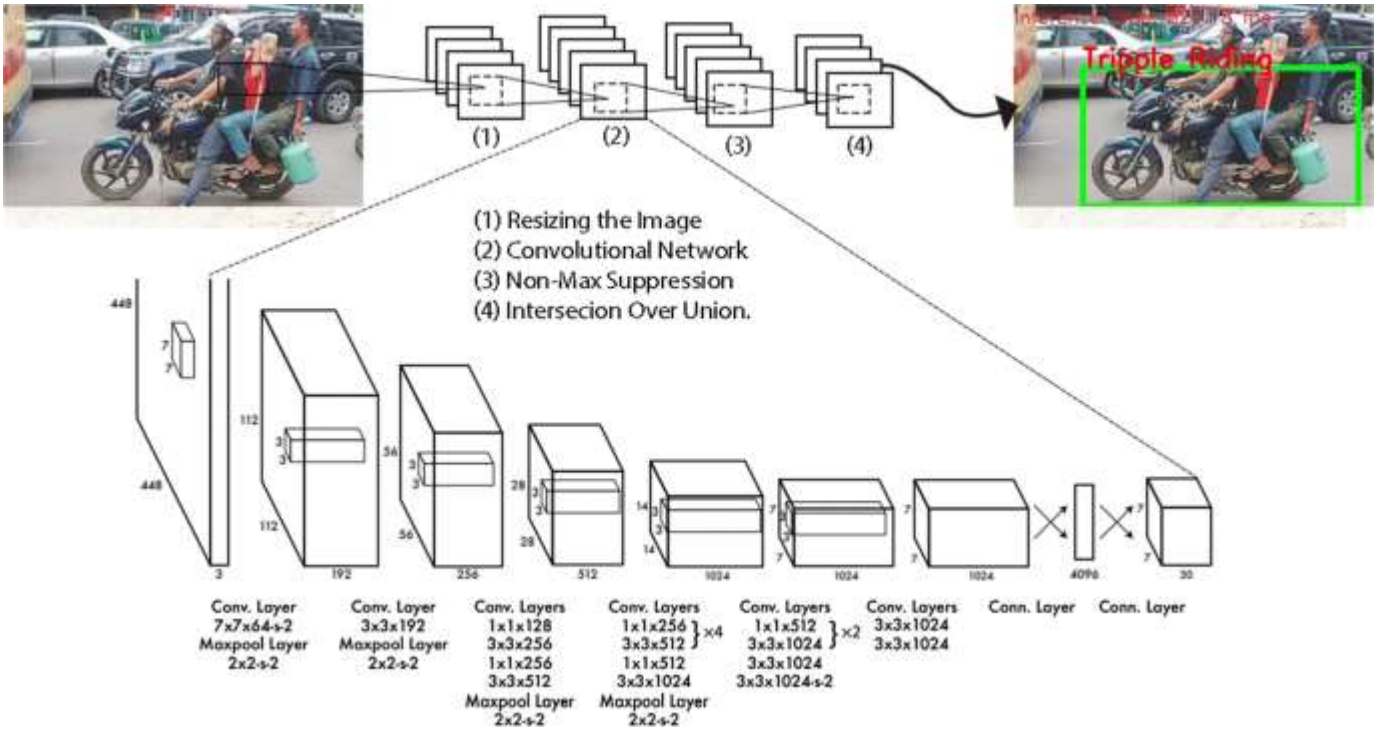
For Motorcycle detection: we used trained model with COCO Dataset with accuracy of 99%.

Model for Motorcycle Detection:

Coco Dataset for Motorcycle Detection: COCO is a large-scale object detection, segmentation, and captioning dataset. This version contains images, bounding boxes " and labels for the 2017 version. Coco defines 80 classes. COCO stands for Common Object Context. The COCO dataset contains the images which are captured in our daily life scenes. COCO provides multi-object labeling, segmentation mask annotations, image captioning, key-point detection and panoptic segmentation annotations with a total of 80 categories, making it a very versatile and multi-purpose dataset. As we have mentioned above that the COCO dataset contains a total of 80 categories, we import the required libraries that are required to access the COCO dataset. The input contains many objects like Buses, Cars, and Motorcycles etc. The libraries imported will helps divide the required objects from all other objects and the detected objects will be given certain IDs to access them later. Coco.GetObjectIds is a function used to get the IDs for the differentiated objects. The 80 categories are as follows

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

Triple Riding Detection Proposed Model The system consists of a main subsystem: setting up the environment, training, testing the model and getting the accurate coordinates of the vehicle that comes along with the system. The Proposed system of triple riding is illustrated in following figure.



Working of yolo

The system mainly uses the deconvolutional approach of deep learning along with the deep learning framework Darknet and the object detection algorithm YOLOv3 performs the various functionalities like detection, recognition, and Identification, followed by classification. The subsystem having the support of the image, video and live feed as the input, enables the system to process all kinds of data. Having the image and video input assessment as an extension, the live feed of the camera modules deployed at the junctions processes every single frame. Every single frame is subjected to the detection of various objects. In our prototype model, the various objects that were subjected to the process of the training are the bike and the person. The two objects bike, and person are the most important aspects of the focus. The detection process of the YOLO model makes use of the target regression as a regression problem for the spatially separate target box and confidence.

1. The YOLO first divides the image into convolutions of size NxN like 13x13, and the size of each of the NxN cells depends on the size of the input.
2. Each cell of these NxN cells is responsible for predicting the number of bounding boxes in the input image.
3. For every box in the input, the deconvolutional network predicts the confidence that the bounding box contains the object and the probability of the enclosed object being from one of the classes mentioned in the configuration file.
4. After that, it applies the Non-Max Suppression to remove the bounding boxes with less confidence value.
5. The processed images are subject to the Intersection over union function developed to find out the relation between the persons detected in the frame along with the motorbike. Thereby, marking the rectangular bounding box with triple riding as the label.

Program to detect triple ride

```
import cv2
import numpy as np
from shapely.geometry import Polygon
net = cv2.dnn.readNet('yolov3.weights','yolov3.cfg')


classes =[]
with open('coco.names','r') as f:
   classes= f.read().splitlines()


pict=input("Enter the pic name ")
img = cv2.imread(pict)
height, width, _ = img.shape


blob = cv2.dnn.blobFromImage(img, 1/255, (416,416), (0,0,0), swapRB=True,
crop=False)


net.setInput(blob)
output_layers_names = net.getUnconnectedOutLayersNames()
layerOutputs = net.forward(output_layers_names)


boxes = []
confidences = []
class_ids = []
```

```
for output in layerOutputs:
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence> 0.5:
            center_x = int(detection[0]*width)
            center_y = int(detection[1]*height)
            w = int(detection[2]*width)
            h = int(detection[3]*height)

            x = int(center_x - w/2)
            y = int(center_y - h/2)

            boxes.append([x,y,w,h])
            confidences.append(float(confidence))
            class_ids.append(class_id)


indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
font = cv2.FONT_HERSHEY_PLAIN
colors = np.random.uniform(0, 255, size = (len(boxes), 3))


bike_list=[]
```

```
persons_list = []
new_bike_list = []



for i in indexes.flatten():


    x,y,w,h = boxes[i]
    label = str(classes[class_ids[i]])
    confidence = str(round(confidences[i],2))
    color = colors[i]
    if label =='motorbike':
        new_bike_list.append([x,y,w,h])
        m1 = (x,y+h)
        m2 = (x+w,y+h)
        m3 = (x+w,y-40)
        m4 = (x,y-40)
        m_p = Polygon([m1,m2,m3,m4])
        bike_list.append(m_p)


    elif label == 'person' :


        p1 = (x,y+h)
        p2 = (x+w,y+h)
        p3 = (x+w,y)
        p4 = (x,y)
        p_p = Polygon([p1,p2,p3,p4])
```

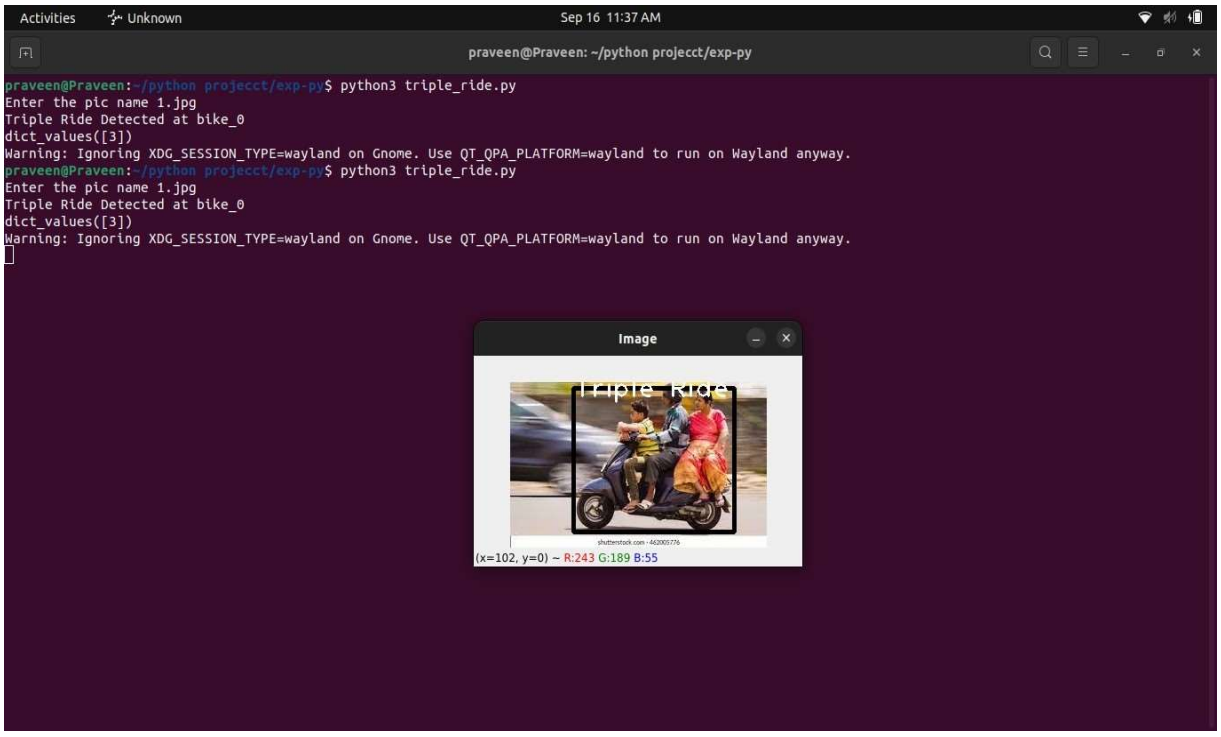
```
        persons_list.append(p_p)
    else:
        pass



bike_dict = {}
for i in range(len(bike_list)):
    intersection_list = []
    for j in persons_list:
        intersect = bike_list[i].intersection(j).area
        intersection_list.append(intersect)
    if 'bike_'+str(i) not in bike_dict:
        bike_dict['bike_'+str(i)] = intersection_list



new_dict ={}
for k,v in bike_dict.items():
    c=0
    for i in v:
        if i>=0.5:
            c+=1
        else:
            c=c
    if k not in new_dict:
        new_dict[k]=c
```

```
for k in new_dict.keys():
    if new_dict[k]>=3:
        print(f'Triple Ride Detected at {k}')
    else:
        print(f'Triple Ride not Detected at {k}')



print(new_dict.values())
l=list(new_dict.values())
for i in range(len(l)):
    x,y,w,h = new_bike_list[i]
    if l[i]<=2:
        cv2.rectangle(img, (x,y-40), (x+w,y+h), (255,255,255),3)
        cv2.putText(img, "No Triple Ride",(x,y+20), font, 2, (0,255,0), 2)
    else:
        cv2.rectangle(img, (x,y-40), (x+w,y+h), (0,0,0),3)
        cv2.putText(img,"Triple Ride",(x,y-30), font, 2, (255,0,0), 2)

cv2.imshow('Image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

INPUT AND OUPUT:

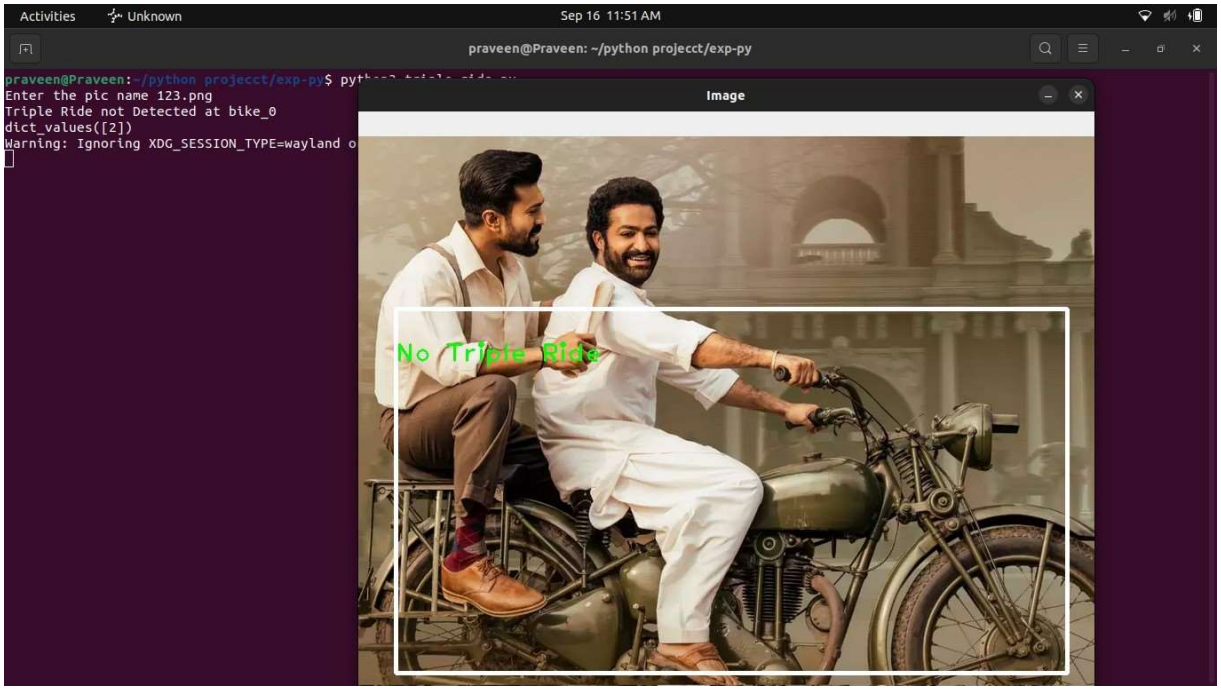
Input:

Output:



Input:

Output:



Conclusion

Our project is an automatic approach for detecting the cases of the triple riding. Our model only calculates Intersection, whereas others use IoU so, in our view our work is faster because of less calculations. The image of the vehicle classified as the Triple Ride is stored along with the data such as vehicle's license plate number. Then police can impose fines easily. This can completely wipe out the human effort in monitoring the traffic system.

References

Coco model and yolo algorithm:

https://appsilon.com/object-detection-yolo-algorithm/

NumPy and cv2:

https://www.w3schools.com/python/numpy/numpy_intro.asp

https://www.geeksforgeeks.org/python-opencv-cv2-imread-method/