IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

Vehicle Overspeed Detection using YOLOV5 in Machine Learning

Kalakonda Shashank[1], Anumandla Sahithya[2], Shaik Shakeel[3], K. Damodhar Rao[4]
B.TECH Scholars, Department of Computer Science and Engineering[1,2,3]
Assistant Professor, Department of Computer Science and Engineering[4]
Sreenidhi Institute of Science & Technology, Hyderabad, India

Abstract: *The main reason of many road accidents that are occurring in modern days are due to over speed and negligence driving . Existing system uses many approaches and needed improvement in performance. Many Systems requires several specialized Hardware's and Sensors which makes it less practical and costly. More sensors increase the overall cost of the operation/ procedure becomes high and dependency of manual workforce is increased. There are several image processing techniques which uses edge detection to detect object and uses simple formula to detect the vehicle and calculate it's speed. These methods are very unreliable and the system works on hard-coded rules. The project uses computer vision and deep learning algorithms such as yolo to detect speeding violations and report violations to law enforcement officers. When speeding is detected, an image of the offending vehicle is captured and emailed to law enforcement. The system requires little manual effort and can run continuously. We are using opencv yolo machine learning algorithms and required python modules to identify any type of vehicles . These System requires almost no manual work and can work continuously. The Final output of the project will be an App like system which can be deployed easily and can be used easily. The system here understands the vehicle and it's type based on the training data provided.*

Keywords: Vehicle detection, speed check, Machine Learning, Yolov5

I. INTRODUCTION

Vehicle overspeed detection systems have become increasingly crucial in recent years promote road safety and prevent accidents caused by high-speed driving. This is because speeding is a major contributor to accidents on the road, resulting in severe consequences such as injuries, fatalities, and significant economic and social costs. The purpose of vehicle overspeed detection systems is to identify and alert drivers and authorities to instances of overspeeding to prevent accidents, improve road safety, and ultimately save lives. Overspeeding is a common problem, and there are several reasons why drivers exceed the speed limit. It could be due to poor road conditions, driver distraction, or simply a lack of awareness of the dangers of overspeeding. Therefore, vehicle overspeed detection systems are crucial to promoting safe driving behavior and reducing the incidence of accidents caused by overspeeding. These systems utilize technologies such as Doppler radar, LiDAR, and cameras to accurately measure the speed of moving vehicles and identify those that are exceeding the speed limit. They can be installed on highways, city streets, and other busy roadways, and are typically integrated with traffic control systems to manage traffic flow more efficiently. One of the key benefits of vehicle overspeed detection systems is that they provide real- time data on traffic patterns, driver behavior, and road conditions. This information can be utilized to optimize traffic flow, improve road design, and implement more effective speed limits to enhance road safety further. These systems also aid law enforcement in identifying and prosecuting drivers who violate the speed limit, thereby promoting greater accountability and adherence to traffic laws. Recent advancements in technology have led to more advanced vehicle overspeed detection systems, which incorporate artificial intelligence and machine learning algorithms to provide more accurate and efficient speed detection capabilities. Some of these systems can even recognize individual vehicles and drivers, enabling authorities to monitor repeat offenders more closely and take appropriate action. As the number of vehicles on the road continues to rise, speeding remains a significant contributor to accidents. Therefore, vehicle overspeed detection systems are becoming increasingly crucial in ensuring public safety on the road. By investing in these systems, we can create a safer and more sustainable transportation system for everyone. Nowadays AI

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

have grown very fast in recent times which making life easy. In our project, we are using Machine learning algorithm YOLOV5 to detect vehicle overspeeding by training the model with COCO dataset. This project helps to detect overspeeding vehicles and report a mail to officialsto take severe actions. These systems require almost no manual work and can work continuously.

II. METHODOLOGY

Objective

In general we use high level bandwith capture sensors to detect object motion .Here we are using machine learning algorithm YOLOV5 .This model was pretrained with over ten thousand images .The model was trained and tested with COCO dataset . After model was trained , a new input video has been given to model to detect speed .It checks thespeed , if speed >60 a screenshot is been captured and a mail will be sent along with captured picture to higher officials .It also detect animals or humans who are on road and immediately intimate to higher officials

Problem Statement

To check vehciles speed using Yolov5 machine learning algorithm and send a mail to higher authorities if vehicles cross its speed limit.

Existing System

Existing system uses many approaches and needed improvement in performance. Many Systems requires several specialized Hardware's and Sensorswhich makes it less practical and costly.More sensors increase the overall cost of the operation/ procedure becomes high and dependency of manual workforce is increased.Thereare several image processing techniques which usesedge detection to detect object and uses simple formula to detect the vehicle and calculate it's speed.These methods are very unreliable and the system works on hard-coded rules.

Proposed System

In this project, by the use of computer vision and Deep Learning based Object Detection YOLO algorithms, we are trying to detect over speeding ofCars and report the violation to the law enforcement officer. If an over speeding is detected, pictures of the particular Car will be taken and sent to Law Enforcement via mail.These System requires almostno manual work and can work continuously. The Final output of the project will be an App like systemwhich can be deployed easily and can be used easily.The system here understands the vehicle and it's type based on the training data provided.

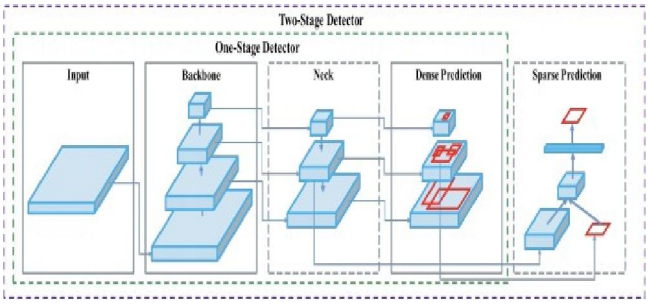
III. ALGORITHM

YOLOV5 is a modern object detection algorithm designed by Ultralytics, which has been receiving a lot of attention due to its accuracy and high speed indetecting objects in images and videos. Thisalgorithm is an updated version of the YOLO series,famous for its capability to detect multiple objects inreal-time in a single image or video frame.The latestversion of the YOLO series, YOLOV5, has introduced several key improvements over its predecessors, which include its speed, accuracy, and architecture. The algorithm is significantly faster than YOLOv4, its predecessor, while providing higher accuracy in real-time object detection. This enhancement makes it an attractive option for various applications that require fast and precise object detection, such as robotics, autonomous driving, and surveillance. The new architecture of YOLOV5 is optimized for detecting objects of different sizes, shapes, and orientations, using anchor boxes that are adjustable to match the objects' characteristics in the image. This feature allows YOLOV5 to detect objects more accurately, even when they are partially obscured or have complex shapes.Apart from its architecture, YOLOV5 also introduced new features such as mosaic data augmentation, which combines multiple images into a single image for training, and self-adversarial training, which improves the algorithm's robustness to variations in the input data. These features enhance the algorithm's performance and reliability, making it more effective for real-world applications.In conclusion, YOLOV5 is a powerful and efficient object detection algorithm thatoutperforms its predecessors in terms of accuracy, speed, and versatility. Its development and refinement continue to make it a valuable tool for various applications in the fields of computer vision and objectdetection.

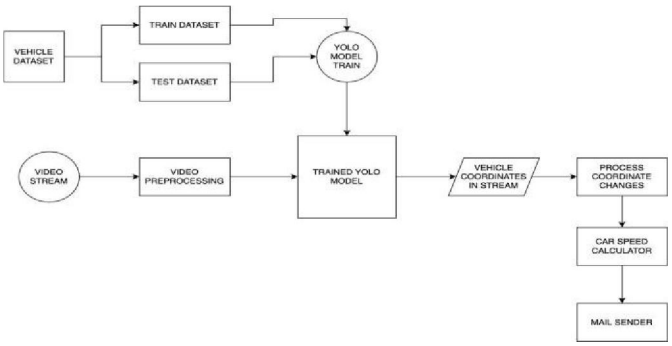
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

Finally, YOLOV5 refines the predicted bounding boxesto improve their accuracy. The algorithm uses a technique called Non-Maximum Suppression (NMS) to remove redundant bounding boxes and merge overlapping ones. This process ensures that only one bounding box is assigned to each object detected in theimage.Overall, YOLOV5's object detection process is highly efficient and accurate, thanks to its use of a deepconvolutional neural network and advanced techniques such as Non-Maximum Suppression. Its speed and versatility make it an ideal choice for real-time object detection applications, including robotics, surveillance.Input Image: YOLOv5 takes an input image or video frame and feeds it into the neural network.

- Backbone: The neural network first extracts high-level features from the input image using a convolutional neural network backbone, such as CSPNet or EfficientNet.
- Neck: Next, a neck network is applied to further refinethe features extracted by the backbone network.
- Head: Finally, a detection head is applied to generate bounding boxes around the objects of interest and classify them into different object categories. The detection head consists of convolutional layers thatpredict the object's location, width, height, and confidence score, as well as the object class probabilities.
- Non-Maximum Suppression: The model then applies non-maximum suppression to the detected bounding boxes to remove duplicate detections and keep only themost confident ones.
- Output: The final output is a set of bounding boxes, each with a class label and confidence score, indicatingthe presence and location of the object in the input image



IV. ARCHITECTURE



1. The above architecture can be seen as 3-tierarchitecture in the following way.
2. Intially, the vehicle(speedy vehicles) dataset issplitted into train and test datasets in order to avoid overfitting and for easy evaluation. This trained and tested dataset together will be sent to YOLO model for trainingthe datatset and loading. This comes into Client/application layer.
3. Secondly, the video recorded or captured by the camera will be sent into processing for pixilation and trained YOLO modelwill be working on it. This comes into User-Interface layer.
4. Thirdly, user requests or user required data willbe processed and routes will be accessed. In this manner, car over speeding will becalculated and final result will be stored in the database.
5. Eventually, the stored data will be sent todepartment officials via mails and this comes into storage layer

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

V. IMPLEMENTATION

```
import os
import systemcheck
import cv2
import torch
import time

from models.common import DetectMultiBackend
from utils.datasets import LoadImages
from utils.general import check_img_size, non_max_suppression, scale_coords
from utils.plots import Annotator, colors
from utils.torch_utils import select_device

# from playsound import playsound
from mailsend import sendmail
```

```
#Get the Speed of vehicle based on the X movement of Car
def get_speed(x_movement):
    return (x_movement * (metre_each_px/1000))/hour_per_frame
```

```
def get_lane_of_car(coord, image):

    detect_speed = 0
    if  vertical_line_x[0] < int(coord[2]) < vertical_line_x[1]:
        cv2.putText(image, "*", (int(coord[2])-125,int(coord[3])-50), cv2.FONT_HERSHEY_PLAIN, 3, (20,250,20), 3)
        detect_speed = 1

    if detect_speed:
        print()
        # print("Received Coords:", coord)

    top_y_lane = None
    top_y_lane_diff = None
    bottom_y_lane = None
    bottom_y_lane_diff = None

    #check of top y
    for lane_y in lanes_y:
        if coord[1] < lane_y:
            top_y_lane = lanes_y.index(lane_y)
            top_y_lane_diff = abs(int(lane_y - coord[1]))
            break
    image = cv2.circle(image, (int(coord[2]),int(coord[1])), 5, (255,255,255), 5)

    #check of bottom y
    last_coord = 0
    for lane_y in lanes_y:
        if coord[3] < lane_y:
            bottom_y_lane = lanes_y.index(lane_y)
            bottom_y_lane_diff = abs(int(last_coord - coord[3]))
            last_coord = lane_y
            break
```

```
if detect_speed:
    print("Present in Lanes:", top_y_lane, bottom_y_lane, top_y_lane_diff, bottom_y_lane_diff)


if top_y_lane == bottom_y_lane:
    cv2.putText(image, str(top_y_lane), (int(coord[2])-100,int(coord[3])-50), cv2.FONT_HERSHEY_PLAIN, 3, (20,250,20), 3)
    return top_y_lane, image, detect_speed
else:
    if bottom_y_lane_diff < top_y_lane_diff:
        cv2.putText(image, str(top_y_lane), (int(coord[2])-100,int(coord[3])-50), cv2.FONT_HERSHEY_PLAIN, 3, (20,250,20), 3)
        return top_y_lane, image, detect_speed
    else:
        cv2.putText(image, str(bottom_y_lane), (int(coord[2])-100,int(coord[3])-50), cv2.FONT_HERSHEY_PLAIN, 3, (20,250,20), 3)
        return bottom_y_lane, image, detect_speed
```

```
#Get the Speed of vehicle based on the X movement of Car
def get_speed(x_movement):
    return (x_movement * (metre_each_px/1000))/hour_per_frame
```

```
def detect(model_yolo, image_path, conf_thres=0.50, iou_thres=0.45):

    model, imgsz, device = model_yolo
    coords, class_list = list(), list() #Will be used to store coordinates and names

    loaded_image = LoadImages(image_path, img_size=imgsz, stride=model.stride)
    for path, im, raw, vid_cap, _ in loaded_image:
        im = torch.from_numpy(im).to(device)
        if device.type != 'cpu':
            im = im.half()/ 255
        else:
            im = im.float()/255  # uint8 to float and normalise

        if len(im.shape) == 3:
            im = im[None] # [R][G][B] -> [[R][G][B]]

        final_preds = non_max_suppression(model(im), conf_thres, iou_thres,max_det=100)
```

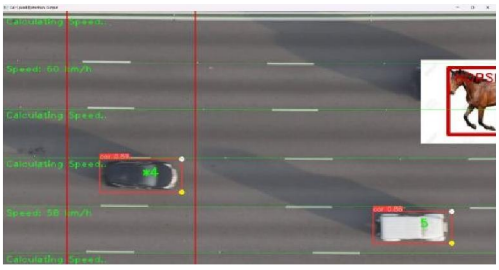
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

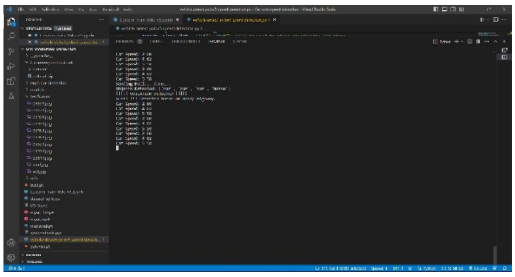
```python
if __name__ == '__main__':

    model_yolo = get_model("best.pt") # Yolo Model for detection of Car
    model_animal_yolo = get_model("yolov5x.pt") # Yolo Model for Detection of Human, Animal
    temp = 0
    vid = cv2.VideoCapture("input_1.mp4") # 1080 x 1920 video
    ret = True
    while ret:
        final_coords = []
        final_class = []

        ret, img = vid.read()
        car_detected = False
        if ret:
            img = cv2.resize(img, (1920, 1080)) #Default size
            cv2.imwrite("testframes/vid.jpg", img)

            image, coords, class_list = detect(model_yolo, image_path = f"testframes/vid.jpg", conf_thres=0.70)
            image2, coords2, class_list2 = detect(model_animal_yolo, image_path = f"testframes/vid.jpg")

            for coord in coords:
                final_coords.append(coord)
            for cls in class_list:
                final_class.append(cls)

            for coord in coords2:
                final_coords.append(coord)
            for cls in class_list2:
                final_class.append(cls)
```

```python
voil = 0
for voilation in VOILATE:
    if voilation in final_class:
        print("!!!!! VOILATION DETECTED !!!!!")
        print(f"ALERT !!! Detected {voilation} on Road/ Highway.")

        coord = final_coords[final_class.index(voilation)]
        image = cv2.rectangle(image, (int(coord[0]),int(coord[1])), (int(coord[2]),int(coord[3])), (0,0,200), 10)
        cv2.putText(image, str(voilation).upper(), (int(coord[0])+20,int(coord[1])+50)), cv2.FONT_HERSHEY_PLAIN, 3, (0,0,2

        final_coords.remove(coord)
        final_class.remove(voilation)

        voil = 1

if voil:
    pass
```

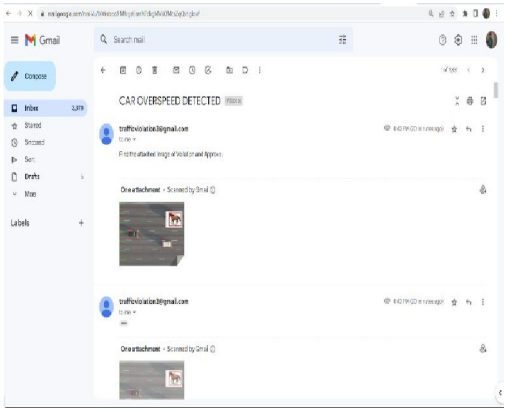
```python
if do_calculation:
    # Car went out of Detection Range
    if last_condition[current_lane] == 1 and detect_speed == 0:
        print("A Car Exited in Detection Area of Lane:", current_lane)
        if len(speed_dict[current_lane]) > 2:
            final_speed_dict[current_lane] = int(sum(speed_dict[current_lane])/len(speed_dict[current_lane]))
        else:
            final_speed_dict[current_lane] = None

        speed_dict[current_lane] = []
        last_position_dict[current_lane] = None

    #New Car Arrived in the Detection range
    if last_condition[current_lane] == 0 and detect_speed == 1:
        print("A Car Entered in Detection Area of Lane:", current_lane)
        final_speed_dict[current_lane] = None
        speed_dict[current_lane] = []
        last_position_dict[current_lane] = None


    last_condition[current_lane] = detect_speed
```

VI. OUTPUT SCREENS





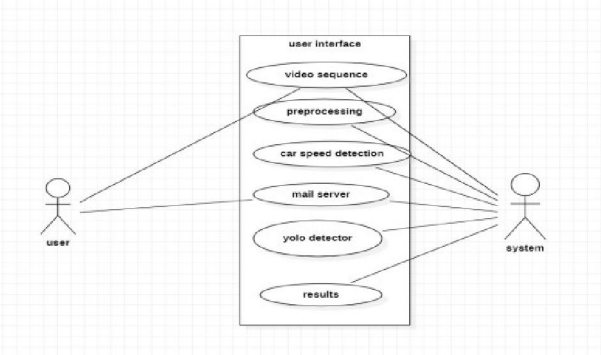
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023



UML DIAGRAMS

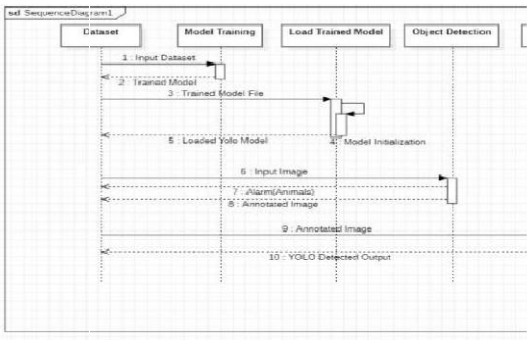
Use case diagram

Use case diagrams helps in summarizing the system users data and their interactions with system. Use case are used to represent system and user interactions. It is used in defining and organizing the functional requirements in a system



Sequence diagram

Sequential diagrams are one of the interaction diagrams that deals thith how operations are carriy out. The interaction between objects are captured in the context of collabaration



VIII. ADVANTAGES OF SYSTEM

In general we use sensors to detect objects and use some different processors to calculate speed .World is growing fast with AI .AI making life simple and decreasing the spending cost . This system can detect objects, calculates speed and also informs to higher offcials if any obstacles happens on road. It need less manual work and cost effective.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 14, May 2023

IX CONCLUSION

The speed detection system that was designed can continuously monitor the speed of approaching vehicles. The accuracy of the output is higher when there are no other moving objects around. The output screen displays the speed value of each vehicle that passes by the system. Whenever a vehicle exceeds the speed limit, the camera is triggered and the image is sent to the officials. This project is helpful in detecting vehicles that exceed the speed limit and alerts law enforcement officials when people or animals cross the road, ultimately leading to a reduction in accidents, especially on highways. Additionally, it is cost-effective and requires less manual work

REFERENCES

[1] ArasRad, Abbas Dehghani, Mohamed Rehan Karim, Vehicle speed discovery in videotape image sequences using CVS system, International Journal of the Physical loresVol. 5(7)
[2] Mohit Chandorkar, Shivam,Dr. Sachin, Vehicle Detection and Speed Tracking, International Journal of Engineering Research & Technology(IJERT)Vol. 10 Issue 05
[3] Lakshitaa Sehgal, Anshal Aggarwal, Sarthak Sood, Aryan Aggarwal, Real- Time Automated Overspeeding Detection and Identification System, International Research Journal of Engineering and Technology (IRJET) Volume Issue 02
[4] Shashwat tripathi, Vivek kumar Singh, Shahzad ahmed, Shivam srivastav,Mrs. Zainab Kamal Khan, Vehicle Detection and Speed Tracking System, International journal of advance exploration and innovative ideas in education(IJARIIE) Vol- 8 Issue- 3
[5] Dr. Philip Heller,Dr. Robert Chun, Vyas Bhagwat, Samkit Patira, Over speed discovery using Artificial Intelligence, SJSU Scholar Works
[6] Akash Gaur, Aditya Vats, Akash Raturi,Ms. Priyanka Bhardwaj, Automatic Vehicle Plate Recognition And Over Speed Discovery Using Machine literacy, Journal of Emerging Technologies and Innovative exploration (JETIR), Volume 9 Issue 5
[7] Sowmya,V. andR. Radha. Heavy- Vehicle Discovery Grounded on YOLOv4 featuring Data addition and Transfer Learning ways. in Journal of Physics Conference Series. 2021. IOP Publishing.
[8] Addict,Q.,L. Brown, andJ. Smith. A near look at Faster R- CNN for vehicle discovery. in 2016 IEEE intelligent vehicles council(IV). 2016. IEEE.
[9] Fang,W.,L. Wang, andP. Ren, bitty- YOLO A real- time object discovery system for constrained surroundings. IEEE Access, 2019. 8p. 1935- 1944.
[10] Fang,L.,X. Zhao, andS. Zhang, Small- objectness sensitive discovery grounded on shifted single shot sensor. Multimedia Tools and Applications, 2019. 78(10)p. 13227- 13245.
[11] M.Y.Yang, W.Liao, X.Li, and B. Rosenhahn, " Deep learning for vehicle discovery in upstanding images, " in 2018 25th IEEE International Conference on Image Processing(ICIP),pp. 30793083, Athens, Greece, 2018.
[12] Sowmya,V. andR. Radha. Heavy- Vehicle Discovery Grounded on YOLOv4 featuring Data addition and Transfer Learning ways. in Journal of Physics Conference Series. 2021. IOP Publishing