Vehicle Detection and Speed Tracking System Using YOLO v8

Mr. B. ARUNMOZHI M.E, Assistant Professor of Computer science Department,
Department of Computer Science and Engineering
Ms. G. SUBANANDHINI, B.E, Student of Computer Science Engineering
Ms. E. MONISHA, B.E, Student of Computer science and Engineering
St. Joseph College of Engineering, Sriperumbudur, Chennai.

Abstract

Speed Detection of Vehicle & its tracking plays an important role for safety of civilian lives, thus preventing many mishaps. This module plays a very significant role in the monitoring of traffic where efficient management and safety of citizens is the main concern. The Speed is determined using distance travelled by vehicle over number of frames and frame rate. Vehicle detection isa part of speed detection where the vehicle is located using various algorithms and later determination of speed takes place.

Machine learning is actively being used today, perhaps in many more places than o One of the root causes of road accidents is Speed. Extracting frames from the video and comparing the speed between two given points can be used to determine whether the car is moving above the permissible limit or not. Detection of vehicle and tracking of speed if the crucial part of town planning.In the last decade, vision-based traffic monitoring system has received considerable attention. Speed detection of vehicle and its tracking plays an important role for safety of civilian lives. Vehicle Speed surveillance is a predominant factor in enforcing traffic laws.

Traditionally vehicle speed surveillance was done using a radar technology which consists of a radar gun and a radar detector. With the help of vehicle detection and speed monitoring. The monitoring system gives various information about, vehicle count, traffic congestion and speed of the vehicle.

Key Terms: YOLO - You Only Look Once, HOG - Histogram Of Gradient, OCR – Optical Character Recognition, GRU – Gate Resurrection Units, VD – Vehicle Detection, ST – Speed Tracking, CNN – Convolutional Neural Networks

Introduction

Detection of vehicles and tracking of speed if the crucial part of town planning. In the last decade, vision-based traffic monitoring systems have received considerable attention. Speed detection of vehicle and its tracking plays an important role for safety of civilian lives. Vehicle Speed surveillance is a predominant factor in enforcing traffic laws. Traditionally vehicle speed surveillance was done using a radar technology which consists of a radar gun and a radar detector. With the help of vehicle detection and speed monitoring. The monitoring system gives various information about, vehicle count, traffic congestion and speed of the vehicle.

One of the root cause of road accidents is Speed. Extracting frames from the video and comparing the speed between two given points can be used to determine whether the car is moving above the permissible limit or not. Speed Detection of Vehicle & its tracking plays an important role for safety of civilian lives, thus preventing many mishaps. This module plays a very significant role in the monitoring of traffic where efficient management and safety of citizens is the main concern. The Speed is determined using distance travelled by vehicle over number of frames and frame rate.

Vehicle detection is a part of speed detection where, the vehicle is located using various algorithms and later determination of speed takes place. Radar is an acronym that stands for Radio Detection and Ranging. Radar systems create radio waves, a form of electromagnetic energy that can be directed out into the air where the signals produced travel at the speed of light – roughly 186,000 miles per second, or 3.08 x 108 meters per second. The transmission of these signals and the collection of returned energy that bounce off of objects in the path of the radar's transmission (called returned pulses) is what allows radar to be used to detect objects and range them, meaning establish their position and distance relative to the radar system's location.

Literature Survey

The paper "SC-YOLO Object Detection Model for Small Traffic Signs." By Yanli Shi Xiangdong Li, and Miapmiao Chen in 2023, this paper is devoted to improving the accuracy of small traffic sign recognition with a model of small number of parameters. Small traffic sign detection has been a challenge for object detection. Although previous methods have achieved good results in this direction, the complexity and accuracy of the model still need to reach a rational level. This paper proposes a high-performance object detection.

The paper “Automatic Social Distance Monitoring System using Deep Learning Algorithms.” By T.Shanthi,R.Anand,K.Hareesh,M.S.Jagan,V.Baskar,and Thanneru Bhanu Prakash in 2021, this paper provides a prominent solution for monitoring using the Open-Source Computer Vision Library (Open-CV) and Deep Learning Algorithm to course people in public areas and to avoid crowding.

The paper” CRAS – YOLO: A Novel Multi Category Vessel Detection and Classification Model Based on YOLOv5s Algorithm.” By Wenxiao Zhao,Muhammad Syafrudin and Norma Latif Fitriyani in 2023, helps to predict the Vehicle Count, Speed Tracking, License Plate.

Multi-category vessel detection and classification based on satellite imagery attract a lot of attention due to their significant applications in the military and civilian domains.

System Design

This Software helps you to detect the count of vehicles in the road and Speed of each vehicle with 98% accuracy.

After detection of Speed and Vehicle, with the help of EasyOCR and Tesseract module, we can able to find the License Plate and Text in the Vehicle without any interruption. Twilio API can help with sending SMS if respective vehicle is overspeeded at any cost.

With the help of this software, one can detect the count of vehicles and speed of the vehicles respectively. License Plate and SMS API can help with overspeeding or not.

For testing, the process user can use small set of videos smoothly without any delay. There are multiole videos which we can use for testing purposes.

For implementation, we created a yolo model to detect vehicles and tracking and created another YOLO model based on EasyOCR to recognise number plates, combined those two models and integrated in anaconda environment in addition with message system using Twilio.

Position of the individual blobs is identified and a recording table is created after identifying moving vehicles from the traffic scene. The number of row in the recording table represents number of blobs that found in given image while number of column represents the parameter pertaining to each of the identified blobs. The blobs will process regularly and match with the parameter in the table when the vehicular identification and tracking is in the progress.

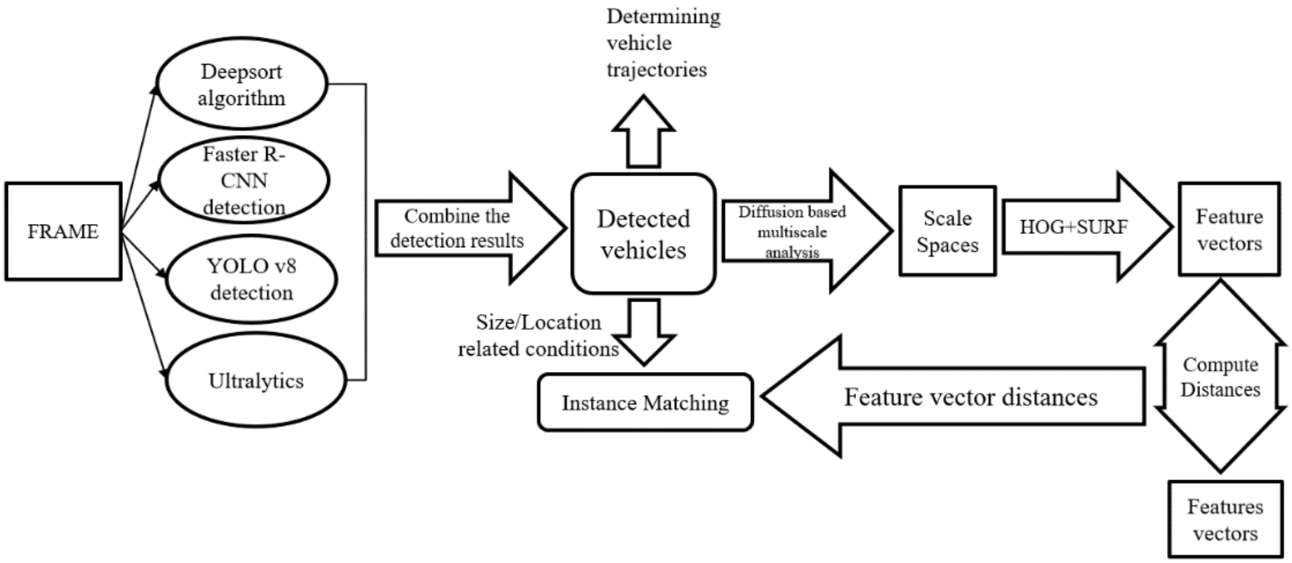
Detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behaviour are three main steps in video analysis. There are few examples of where object tracking apply to

- Motion-based recognition, that is, human identification based on gait, automated object detection, etc. - Automated surveillance, that is, monitoring a scene to detect suspicious activities or unlikely events.

- Video indexing, that is, automatic annotation and retrieval of the videos in multimedia databases

- Human-computer interaction, that is, gesture recognition eye gaze tracking for data input to computers.

- Traffic monitoring, that is, real-time gathering of traffic statistics to direct traffic flow.



IMPLEMENTATION

```
# Ultralytics YOLO 🚀, GPL-3.0 license
import hydra
import torch
from ultralytics.yolo.engine.predictor import BasePredictor
from ultralytics.yolo.utils import DEFAULT_CONFIG, ROOT, ops
from ultralytics.yolo.utils.checks import check_imgsz
from ultralytics.yolo.utils.plotting import Annotator, colors, save_one_box
import easyocr
import cv2
reader = easyocr.Reader(['en'], gpu=True)
def ocr_image(img,coordinates):
    x,y,w, h = int(coordinates[0]), int(coordinates[1]), int(coordinates[2]),int(coordinates[3])
```

```
    img = img[y:h,x:w]


    gray = cv2.cvtColor(img , cv2.COLOR_RGB2GRAY)
    #gray = cv2.resize(gray, None, fx = 3, fy = 3, interpolation = cv2.INTER_CUBIC)
    result = reader.readtext(gray)
    text = ""


    for res in result:
        if len(result) == 1:
            text = res[1]
        if len(result) >1 and len(res[1])>6 and res[2]> 0.2:
            text = res[1]
#       text += res[1] + " "
        print(text)
    return str(text)


class DetectionPredictor(BasePredictor):


    def get_annotator(self, img):
        returnAnnotator(img,line_width=self.args.line_thickness,
example=str(self.model.names))


    def preprocess(self, img):
        img = torch.from_numpy(img).to(self.model.device)
```

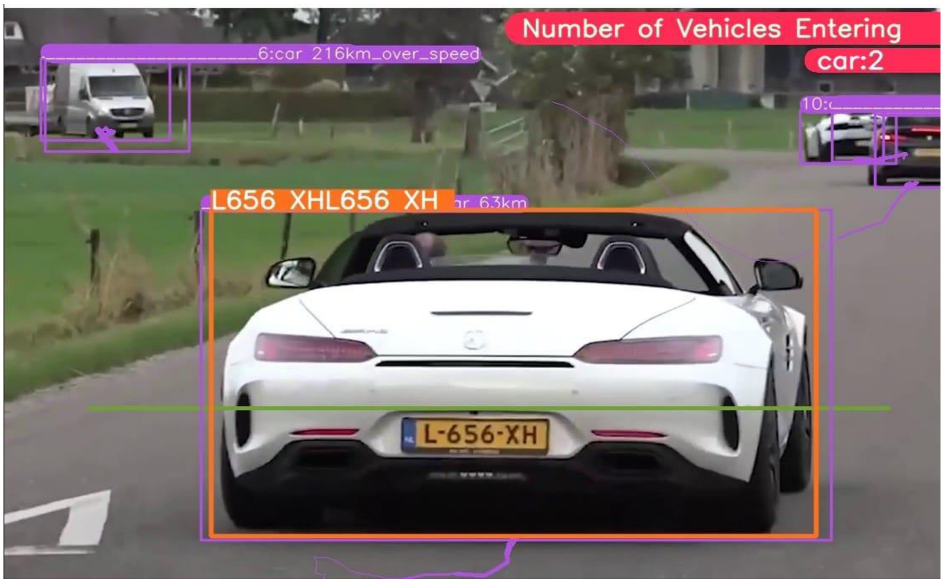
```
    img = img.half() if self.model.fp16 else img.float()  # uint8 to fp16/32
    img /= 255  # 0 - 255 to 0.0 - 1.0
    return img

  def postprocess(self, preds, img, orig_img):
    preds = ops.non_max_suppression(preds,
                                    self.args.conf,
                                    self.args.iou,
                                    agnostic=self.args.agnostic_nms,
                                    max_det=self.args.max_det)
      for i, pred in enumerate(preds):
      shape = orig_img[i].shape if self.webcam else orig_img.shape
      pred[:, :4] = ops.scale_boxes(img.shape[2:], pred[:, :4], shape).round()
    return preds
    def write_results(self, idx, preds, batch):
    p, im, im0 = batch
    log_string = ""
    if len(im.shape) == 3:
      im = im[None]  # expand for batch dim
    self.seen += 1
    im0 = im0.copy()
    if self.webcam:  # batch_size >= 1
      log_string += f'{idx}: '
      frame = self.dataset.count
```

```
else:
    frame = getattr(self.dataset, 'frame', 0)
self.data_path = p
# save_path = str(self.save_dir / p.name)  # im.jpg
self.txt_path = str(self.save_dir / 'labels' / p.stem) + ('' if self.dataset.mode == 'image'
else f'_{frame}')
log_string += '%gx%g ' % im.shape[2:]  # print string
self.annotator = self.get_annotator(im0)
det = preds[idx]
self.all_outputs.append(det)
if len(det) == 0:
    return log_string
for c in det[:, 5].unique():
    n = (det[:, 5] == c).sum()  # detections per class
    log_string += f"{n} {self.model.names[int(c)]}{'s' * (n > 1)}, "
# write
gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
for *xyxy, conf, cls in reversed(det):
    if self.args.save_txt:  # Write to file
        xywh = (ops.xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()  #
        normalized xywh
        line = (cls, *xywh, conf) if self.args.save_conf else (cls, *xywh)  # label format
        with open(f'{self.txt_path}.txt', 'a') as f:
            f.write(('%g ' * len(line)).rstrip() % line + '\n')
if self.args.save or self.args.save_crop or self.args.show:  # Add bbox to image
```

```
            c = int(cls)  # integer class
            label = None if self.args.hide_labels else (
                self.model.names[c] if self.args.hide_conf else f'{self.model.names[c]} {conf:.2f}')
            text_ocr = ocr_image(im0,xyxy)
            label = text_ocr
            self.annotator.box_label(xyxy, label, color=colors(c, True))
        if self.args.save_crop:
            imc = im0.copy()
            save_one_box(xyxy,
                         imc,
                         file=self.save_dir/'crops'/self.model.model.names[c] / f'{self.data_path.stem}.jpg',
                         BGR=True)
    return log_string
@hydra.main(version_base=None,config_path=str(DEFAULT_CONFIG.parent), config_name=DEFAULT_CONFIG.name)
def predict(cfg):
    cfg.model = cfg.model or "yolov8n.pt"
    cfg.imgsz = check_imgsz(cfg.imgsz, min_dim=2)  # check image size
    cfg.source = cfg.source if cfg.source is not None else ROOT / "assets"
    predictor = DetectionPredictor(cfg)
    predictor()
if _name_ == "_main_":
    predict()
```

SNAPSHOTS



CONCLUSION

In conclusion, the integration of vehicle type and license plate recognition using YOLOv8 is a game-changer in the world of traffic monitoring. With its advanced object detection and tracking capabilities, YOLOv8 makes counting and tracking vehicles a breeze. Overall, YOLOv8's advanced object detection and tracking capabilities make it a valuable tool for traffic monitoring and management. The ability to accurately and efficiently track vehicle speed and detect number plates has significant implications for law enforcement, traffic management, and public safety.

FUTURE ENHANCEMENTS

A future enhancement for vehicle speed tracking and number plate detection using YOLOv8 could be the integration of real-time traffic data analysis. By analyzing traffic data in real-time, YOLOv8 could provide insights into traffic patterns, congestion, and accidents, which could be used to optimize traffic flow and improve public safety.

Additionally, YOLOv8 could be integrated with other technologies such as GPS and traffic cameras to provide a more comprehensive view of traffic conditions. This could be particularly useful for law enforcement and emergency services, as it would allow them to respond more quickly and effectively to accidents and other incidents.

REFERENCES:

[1] SC-YOLO: A Object Detection Model for Small Traffic Signs, Yanli Shi, Xiangdong Li, Miaomian Chen,2023 IEEE Base.

[2] Automatic Social-Distance Monitoring System using Deep-Learning Algorithms, T.Shanthi,R.Anand, K. Hareesh, M.S. Jagan, V. Baskar, and Thanneru Bhanu Prakash,2021 IEEE Conference.

[3] CRAS-YOLO: A Novel Multi - Category Vessel Detection and Classification Model Based on YOLOv5s Algorithm, Wenxiao Zhao, Muhammad Syafrudin And Norma Latif Fitriyani, 2023 IEEE Conference.

[4] License Plate Recognition System Based on Improved YOLOv5 and GRU, HENGLIANG SHI &Dongnan Zhao, 2023 IEEE Conference.

[5] Traffic Signs Detection and Recognition Using A Combination of YOLO and CNN, Yasir Ali AlKhafaji; Nidhal K. El Abbadi , 2023 Cited Conference Paper.

6] Methodology for Weapon Detection in Social Media Profiles using an Adaptation of YOLO-V5 and Natural Language Processing Techniques, Alberto Baez-Velazquez; Aldo Hernandez-Suarez; Gabriel SanchezPerezz; Karina Toscano-Medina; Jose Portillo-Portillo, 2023 Cited Conference Paper.

[7] Object Detection through Modified YOLO Neural Network (#8403262, Research Article Open Access – Article ID 8403262.

[8] Real-Time Detection of Vehicle Speed Based on Video Image, (Genyuan Chung, Yubin Guo, Xiaochun Cheng, Dongliang Wang, Jiandong Zhao, 2020 12th International Conference on Measuring Technology and Measuring Automation (ICMTMA), 30 March 2020, DOI: https://doi.org/10.1109/ICMTMA50254.2020.00076, IEEE ACCESS).

[9] Data-driven vehicle speed detection from synthetic driving simulator images,(A.Hernandez Martinez, J.Lorenzo Diaz, I.Garcia Daza, D.Fernandez Llorca, 2021 IEEE International

Intelligent Transportation Conference, DOI: https://doi.org/10.1109/ITSC48978.2021.9564888).

[10] Fast-Yolo-Rec: Incorporating Yolo-Base Detection and Recurrent-Base Prediction Networks for Fast Vehicle Detection in Consecutive Images (Nafiseh Zarei, Paymem Moallem, Mohammedreza Shams, 14th November 2022, DOI : https://doi.org/10.1109/ACCESS.2022.3221942, IEEE ACCESS Volume 10).

AUTHOR 1

Mr.B.Arunmozhi M.E., is an Assistant Professor in the Department of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. He has completed his M.E, CSE under Anna University Affiliation College in the year 2011. He has done his B.E, CSE under Anna University Affiliation College in the year 2007. Mr.B.Arunmozhi has 11 years of teaching experience and has 12 publications in International Journals and Conferences. His area of interests includes Network Security, Computer Networks, Data Science and Machine Learning. He is an active member of CSI and IEANG .He has organized various International Conferences, workshops and Seminars in the area of Computer Networks, Cloud Computing &Machine Learning respectively.

AUTHOR 2

Ms. G.Subanandhini B.E., Student of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops, Seminars in Python, Machine Learning. I got placed in Reputed Companies like Sight Spectrum.

AUTHOR 3

Ms.E.Monisha B.E., Student of Computer Science and Engineering at St.Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops and Seminars in the area of Python and Machine Learning. I got placed in Reputed Companies like Q Spider.