Creating a future-ready workforce

College Name

Park college of engineering and technology

Student Name :Sruthikka.S
Student ID :au712221104022

PROJECT SHOWCASE

Project Title

vehicle speed estimation system

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion

Abstract

- The need to accurately estimate the speed of road vehicles is becoming increasingly important for at least two main reasons. First, the number of speed cameras installed worldwide has been growing in recent years, as the introduction and enforcement of appropriate speed limits are considered one of the most effective means to increase the road safety. Second, traffic monitoring and forecasting in road networks plays a fundamental role to enhance traffic, emissions and energy consumption in smart cities, being the speed of the vehicles one of the most relevant parameters of the traffic state.
- Among the technologies available for the accurate detection of vehicle speed, the use of vision-based systems brings great challenges to be solved, but also great potential advantages, such as the drastic reduction of costs due to the absence of expensive range sensors, and the possibility of identifying vehicles accurately.
- This paper provides a review of vision-based vehicle speed estimation. The terminology and the application domains are described and a complete taxonomy of a large selection of works that categorizes all stages involved is proposed. An overview of performance evaluation metrics and available datasets is provided. Finally, current limitations and future directions are discussed.

Problem Statement

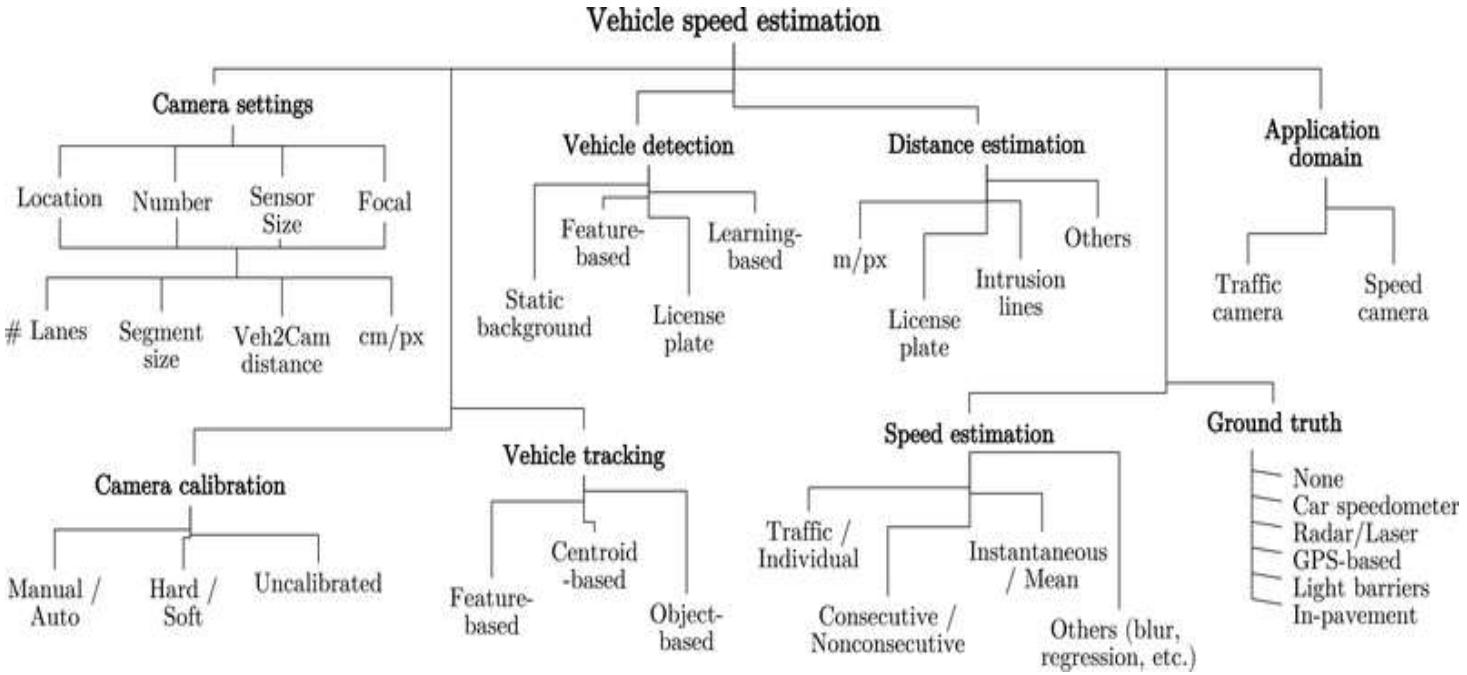
- **BACKGROUND**

Background In recent years many researchers have worked on video cameras which are considered as a sensor device for capturing and recognizing moving vehicle. Video based systems can capture a large variety of information which is less expensive to install and maintain cameras. It is not easy for a single moving camera to quickly capture information.

- **METHODOLOGY**

Methodology The speed of the vehicle in each frame is calculated using the position of the vehicle in each frame, so the next step is to find the spots Bounding, and the centre of gravity. Bubble centroid distance is important to understand the moving vehicle in consecutive frames and therefore is known as the frame rate for motion capture, the speed calculation becomes possible. This information must be recorded in a continuous array cell in the same size as the camera image captured because the distance travelled by the centroid is needed. Specific coordinate will be used to determine the vehicle estimated speed, tracking, and distance travelled by the vehicle.

Project Overview



Proposed Solution

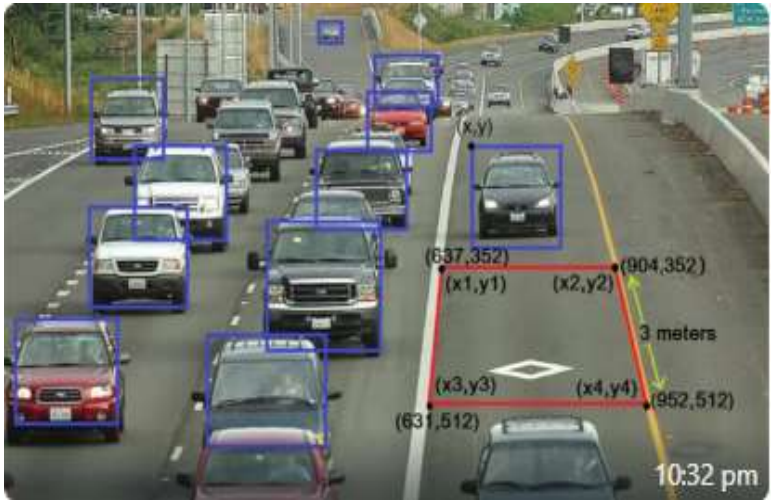
- (1)***Adaptive***
  The protocol should be adaptive to different traffic density. Specifically, it needs to handle network disconnections in sparse traffic condition.
  (2)***Infrastructure-Free***
  The protocol should not rely on any road side infrastructure or cellular station. Only V2V networks should be used.
  (3)***On-Demand***
  The protocol should not wait a period of time to warm up before it can function well. It must work on demand.
  (4)***Real-Time***
  The traffic information needs to be collected and estimated with low latency.

SOURCE CODE:

```
import cv2

# Load video
cap = cv2.VideoCapture('video.mp4')

# Create background subtractor
fgbg = cv2.createBackgroundSubtractorMOG2()

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Apply background subtraction
    fgmask = fgbg.apply(frame)

    # Apply morphology operations
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    fgmask = cv2.morphologyEx(fgmask, cv2.MORPH_OPEN, kernel)

    # Find contours
    contours, _ = cv2.findContours(fgmask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        # Filter out small contours
        if cv2.contourArea(contour) < 500:
            continue

        # Get bounding box
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow('Frame', frame)
    if cv2.waitKey(30) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Results

- Various legal issues arise from such cameras and the laws involved in how cameras can be placed and what evidence is necessary to prosecute a driver varies considerably in different legal systems.
- In some areas the cameras themselves have been ruled illegal. Other issues surround the actual type approval of cameras.



Source :

Conclusion

We finally have a solution to the speeding problem. • We would expect there to be a great number of these every were and that day would be highly visible and identifiable to make a drivers slow down. • We use traffic lights and other traffic manager to reduce the speed. One among them is speed cameras.

Source :

Thank You!