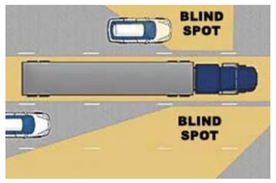
Advanced Driver Assistance Systems Radar Blindspot Monitoring

ELECTRICAL ENGINEERING
UNIVERSITY of WASHINGTON

Matthew Lee, Jerrold Erickson, Yi-Ting Tsai

Sponsor: Daimler Trucks North America | Industry Mentor: Alexander Puschnig | Faculty Mentor: Alexander Mamishev, PhD

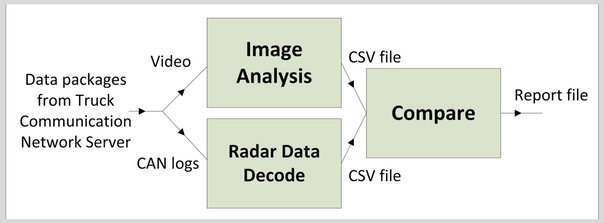


Problem Statement:

Daimler Trucks North America (DTNA) uses a radar system to monitor the right-side blindspot on their trucks in order to give warnings to the driver about objects present. Currently, the system can indicate to the driver that an object is in the blindspot when nothing is actually present. Camera videos of this blindspot are recorded and saved, but the only way to detect these false warnings is to manually search through the footage and find them at random. This process is very time-consuming, but is necessary if the radar system is to be improved. Therefore, the work in this project created a program that uses image analysis techniques to automatically find false warnings.

System Overview

A false positive in the radar system is defined as sending a warning to the driver when there is nothing concerning in the blindspot. One known example of this is that the radar system can detect the truck's trailer and send warnings about it, even though the truck will not hit its own trailer. To find out such instances, we proposed to develop our own obstacle detector using deep learning. By comparing the results of our image analysis program and the radar system data, our system generates reports indicating when and where these instances of false positives occurred for Daimler's engineers to analyze and improve their radar system.



System high-level block diagram

Tools & Implementation

Software
- Python
- TensorFlow
- MATLAB

Hardware Tools
- NVIDIA TITAN X (model training)
- UW EE Linux Machine (system environment)
- Camera (data collection)

Radar Data Decoding

- Daimler uses multiple Controller Area Network (CAN) busses for vehicle systems to communicate with each other
  - For every minute of driving, ~70 MB of CAN logs are recorded
- Python is used to read the CAN bus logs and decode information about the radar blindspot system
- A report containing information about the radar system at every timestamp of the blindspot video is written to a CSV file
  - High-level radar information includes whether or not a warning is currently being delivered to the driver
  - Low-level radar information includes the coordinates of all objects currently detected by the radar system

Image Analysis

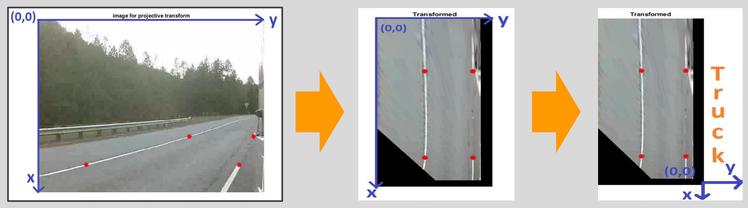
1. **Preprocessing**
   a. Get image frames from input video
   b. Resize and filter image frames for analysis
2. **Object Detection**
   a. Use a Single Shot MultiBox Detector (SSD) in TensorFlow to detect vehicles and trailers with bounding boxes
      - SSD model trained with manually labelled images from datasets expanded by image augmentation

Image augmentation - Get new images by changing the sizes and HSV values of the source image

3. **Object Coordinates Extraction**
   a. Select a position point from the detected bounding boxes
   b. Apply projective transformation on the selected point to get top-view coordinates with OpenCV in Python
   c. Change the top-view coordinates into Daimler's truck coordinates

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Projecting coordinates into another plane using pre-calculated Homography Matrix H



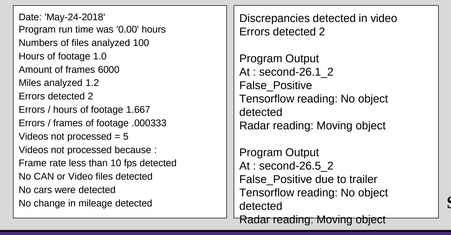
Extracting real-world coordinates from image coordinates using projective transformation

4. **Output Detection Results to a CSV file**
   a. Write timestamps and detection information (detected object class and coordinates) into a CSV file

Example object detection results with real-world object coordinates displayed

Image Analysis & Radar Comparison

At all instances when a warning was sent to the driver, the video feed is analyzed using image analysis. This analysis indicates what was found in the truck's blindspot. If nothing is present, or if only the truck's trailer is present, the instance is flagged by the program as a false positive. The program then generates a report of all of these flagged instances.


Date: 'May-24-2018'
Program run time was '0.00' hours
Numbers of files analyzed 100
Hours of footage 1.0
Amount of frames 6000
Miles analyzed 1.2
Errors detected 2
Errors / hours of footage 1.667
Errors / frames of footage .000333
Videos not processed = 5
Videos not processed because :
Frame rate less than 10 fps detected
No CAN or Video files detected
No cars were detected
No change in mileage detected

Discrepancies detected in video
Errors detected 2

Program Output
At : second-26.1_2
False_Positive
Tensorflow reading: No object detected
Radar reading: Moving object

Program Output
At : second-26.5_2
False_Positive due to trailer
Tensorflow reading: No object detected
Radar reading: Moving object

Sample results

Challenges

- Developing a robust program to run on thousands of CAN logs and videos
- The videos are very low resolution and framerate due to compression
- Image analysis is not accurate for videos recorded at night
- The camera field of view doesn't cover the entire area monitored by the radar system
- The videos would sometimes be named incorrectly, causing our program to analyze a different perspective on the vehicle

Results

The program filtered out all videos recorded at night, cases where the radar system detected objects outside of the camera field of view, and other cases which would impact the validity of our results. After applying these filters, the program successfully and consistently identifies radar false-positives, without flagging legitimate warnings as false positives.

False positives - the radar system is giving blindspot warnings to the driver here

Independent testing of our image analysis program found it to have an accuracy of 99.2% when analyzing 535 test photos.