DeepTrailerAssist: Deep Learning based trailer detection, tracking and articulation angle estimation on automotive rear-view camera

Ashok Dahal[1], Jakir Hossen[1], Sumanth Chennupati[1], Ganesh Sistu[2],
Kazumi Malhan[1], Muhammad Amasha[1] and Senthil Yogamani[2]
[1]Valeo R&D Troy, USA [2]Valeo Vision Systems, Ireland
firstname.lastname@valeo.com

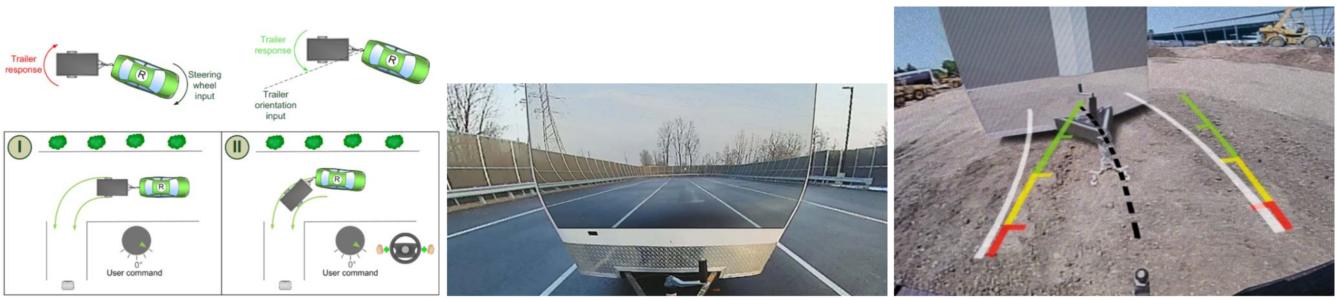

Figure 1: Trailer Assist End User Functions. From left to right: Trailer Maneuver Assist, Trailer See Through [26] and Trailer Hitch Guidance Overlays [25].

Abstract

Trailers are commonly used for transport of goods and recreational materials. Even for experienced drivers, manoeuvres with trailers, especially reversing can be complex and stressful. Thus driver assistance systems are very useful in these scenarios. They are typically achieved by a single rear-view fisheye camera perception algorithms. There is no public dataset for this problem and hence there is very little academic literature on this topic. This motivated us to present all the trailer assist use cases in detail and propose a deep learning based solution for trailer perception problems. Using our proprietary dataset comprising of 11 different trailer types, we achieve a reasonable detection accuracy using a lightweight real-time network running at 30 fps on a low power embedded system. The dataset will be released as a companion to our recently published dataset [24] to encourage further research in this area.

1. Introduction

Advanced Driver Assistance Systems (ADAS) have become a common feature in most of the modern vehicles. Commonly available ADAS features include lane keep assist, cross-traffic alert, front collision warning and traffic sign recognition [8]. Recent textbook by Rezaei and Klette [15] provide an excellent overview of computer vision algorithms used in ADAS systems. The progress in this area is accelerated by the pursuit of fully Autonomous Driving (AD) which has significantly impacted the automotive industry [16]. Complexity of the system and computational power has drastically increased over the last few years as well and the current generation embedded systems can deploy computationally intensive deep learning algorithms using efficient design techniques [18, 2]. Deep learning algorithms are also becoming successful beyond object detection [17] for applications like visual SLAM [13], depth estimation [10], soiling detection [22] and motion estimation [19].

Relatively, trailer assist algorithms are less explored in academic literature due to lack of datasets. A trailer is a wheeled vehicle which is unpowered and towed by a regular vehicle. It is commonly used in rural areas for transportation of animals (e.g: horse trailer) and agricultural produce. It is also used in urban areas for recreational purpose for towing caravans or boats. Complex manoeuvring with trailers can be quite challenging even for experienced drivers and trailer sway accidents are quite common. In this paper, we focus on trailer assist use cases and its associated visual perception algorithms.

Saxe and Cebon [5] use a template matching algorithm to estimate articulation angle of trailer and unscented Kalman filter for tracking. Caup *et al*. [3] convert the image to polar co-ordinates to better estimate the articulation angle and use edge-based operators to detect the trailer. Xu *et al*. [23] propose hitch angle estimation using a novel vehicle model and Kalman tracker for trailer backup assist algorithm. Ljungqvist *et al*. [12] present a detailed and rigorous path planning control framework for vehicles with a trailer. Atoum *et al*. [1] used a CNN model to detect trailer coupler. In comparison, our system additionally detects trailer and articulated angle as well. Classical computer vision approaches using edge detection and shape detection were commonly used in previous generation systems, however they do not generalize well to different types of trailers.

The rest of the paper is structured as follows. Section 2.1 provides an overview of the trailer assist system including use cases and high level vision modules. Section 3 discusses the proposed trailer perception algorithms using CNN and LSTM. Section 4 discusses the results and technical challenges. Finally, Section 5 summarizes the paper and provides potential future directions.

2. Trailer Assist System

This section describes the various trailer assist use cases, high level vision algorithms needed and an overview of the platform.

2.1. Trailer end user functions

Trailer Maneuver Assist helps the driver when reversing and maneuvering with a trailer. It guides the trailer while reversing in the direction the driver presets or wishes to maneuver. It is hard for an experienced driver to maneuver a trailer as its response to steering input varies drastically as shown in Figure 1. This system helps the driver by automatically driving to reach the desired reversing angle of trailer.

Trailer Hitch Guidance Overlays: In many systems, instead of automated maneuvering, guidance overlays are provided on the dashboard display as shown in Figure 1. Capture from the rear view camera can help align a car/truck with a trailer hitch. It helps line up center of car/truck with center of trailer. The overlays take steering wheel angle of the vehicle as an input and creates an optimal trajectory of tow ball position. These overlays thus can help a driver to maneuver accordingly to get as close as possible to the trailer hitch.

Jackknifing prevention: Jackknifing is a situation where trailer and vehicle fold together at hitch like a jackknife. This happens during backing up of trailers when the articulation/hitch angle increases beyond a safe angle. Continuing a backward motion beyond it can worsen the situation and can possibly led to contact of trailer with

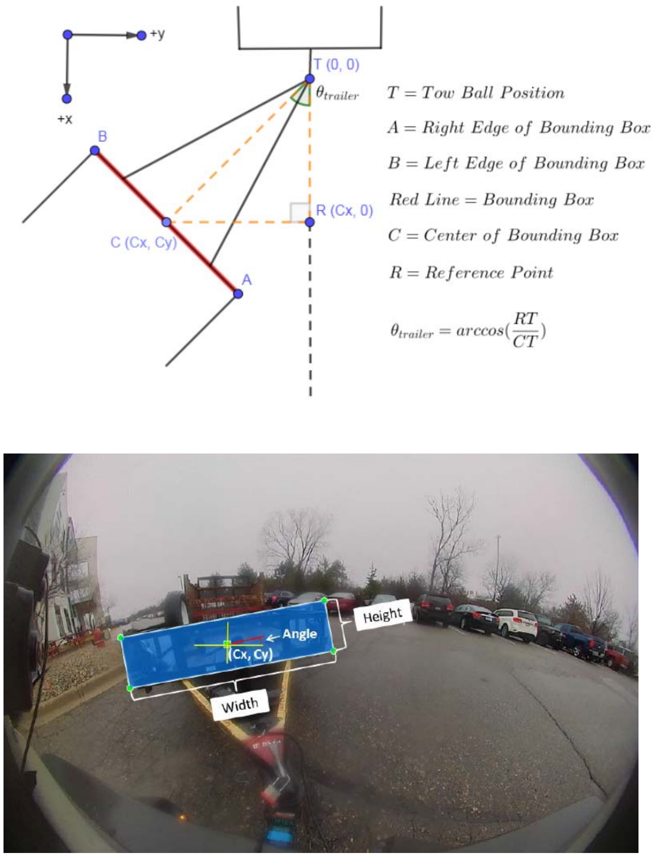

Figure 2: (top) Trailer Angle Calculation based on predicted trailer bounding box. (bottom) Trailer Angle Estimation - Predicted bounding box will mark the front portion of the trailer we are tracking with the following information computed from the bounding box: 1) Bounding Box Width 2) Bounding Box Height 3) Bounding Box Center X coordinate 4) Bounding Box Center Y coordinate and 5) Bounding Box Angle.

vehicle. This can also happen when the vehicle and the trailer are going at high speeds. In order to prevent this behaviour, the articulation/hitch angle has to be monitored actively.

Trailer See Through is constructed using frames captured from cameras located at the rear of both the vehicle and the trailer. These frames are stitched into a single homogeneous image shown in Figure 1. This extends the rear view range behind the vehicle by making the trailer totally invisible. This helps a driver to maneuver with ease in parking lots, drive into merging traffic, make turns and *etc*. It requires an algorithm to find the relative position of the vehicle camera and the trailer camera.

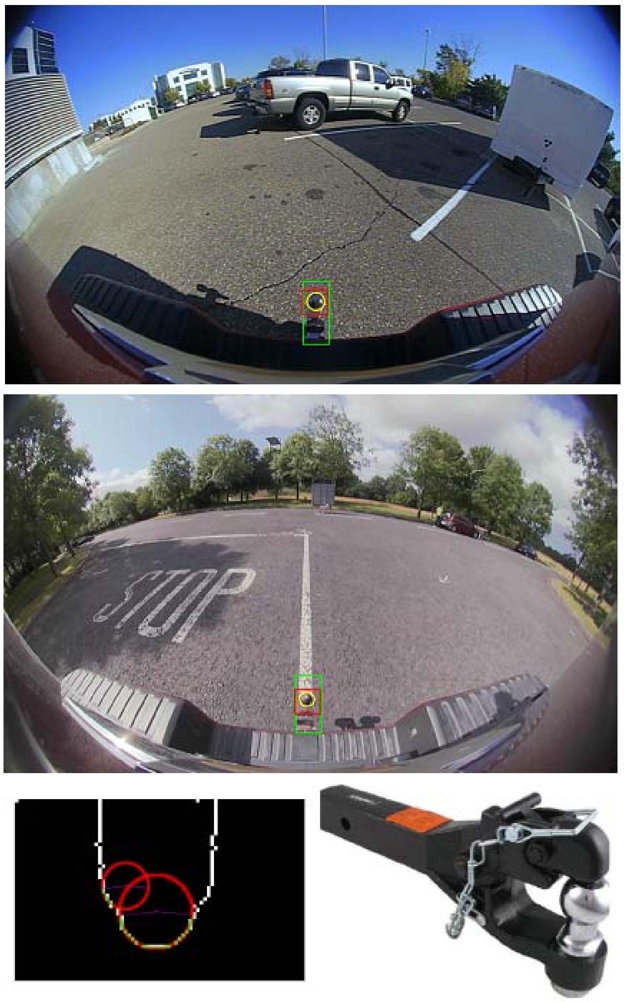

Figure 3: Tow Ball Localization. Top two images illustrate tow ball detection with two bounding boxes and a circle. Red box is used for the tow ball and yellow circle within the box provides better localization of the tow ball. Green box is used for detecting the tow ball bar. Bottom left image illustrates the locking mechanism of tow ball with the trailer's part shown in bottom right image.

2.2. High Level Vision Components

In Section 2.1, we discussed the important trailer assistance systems. These systems require the knowledge of trailer angle w.r.t vehicle, tow ball location and trailer hitch location. In this section, we discuss how we obtain this information using computer vision algorithms.

Trailer Angle Estimation: Figure 2 illustrates the trailer angle definition geometrically in top view and in image view. The trailer angle is defined as the yaw angle w.r.t to the central axis of the vehicle. The center point of bottom side/edge of bounding box (C_x, C_y) is projected

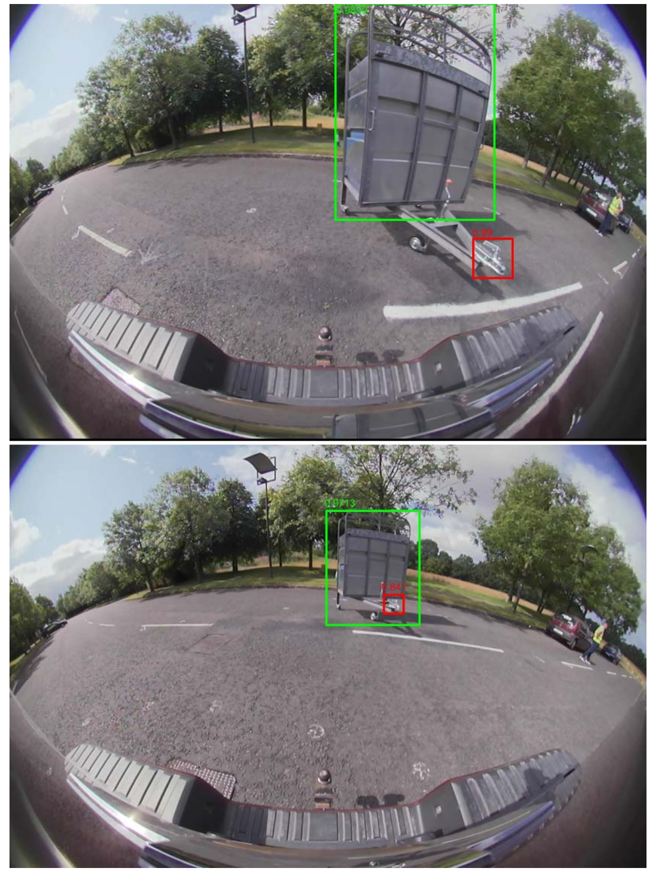

Figure 4: Hitch Couple Localization. Green bounding box is used for detecting the trailer and red bounding box is used for detecting the hitch couple.

from image plane into vehicle coordinates, a.k.a. world coordinates (C_x,C_y,C_z). In vehicle coordinates, the known tow ball position (derived from vehicle mechanical data) will be used, along with the bounding box center, to calculate the trailer angle. This trailer angle can then be used to prevent trailer swing leading to jackknife. This can also help the trailer backup assist system as well as view switching based on the trailer system.

Tow Ball Localization: Figure 3 illustrates Tow Ball Localization (TBL) in the image. The goal is to identify and locate the two ball position using the rear facing camera, so that they can be connected. Commonly used tow balls are couplers with Pintle hitches as shown in bottom right sub-figure of Figure 3. There is no standardized appearance for these tow balls and it becomes challenging to detect various types. This module will be useful for hitching a trailer to a vehicle and also for trailer manoeuvres and overlays.

Hitch Couple Localization: Figure 4 illustrates hitch cou-

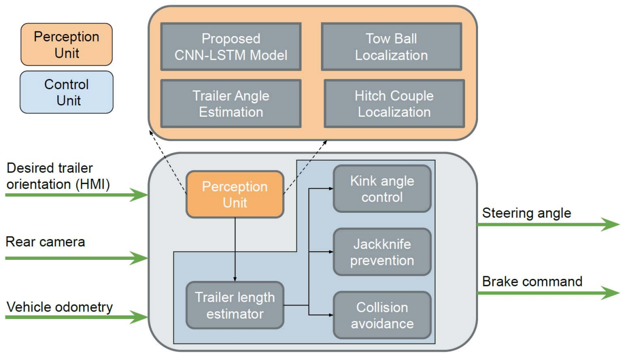

Figure 5: Functional diagram of Trailer Assist

ple localization (HCL) in the image. The goal is to detect the hitch couple and calculate the position of the coupler center without a known reference target on the trailer. The appearance of the hitch will vary significantly depending on its distance to the camera. Thus two different models for near-field and far-field are typically employed. It is challenging to detect it in far-field because of its small size.

2.3. Platform Overview

Camera Sensor: Our car setup comprises of a commercially deployed automotive grade fisheye camera sensor. It can either be a standard rear-view camera or part of a surround-view camera suite comprising of four fisheye cameras around the car. The cameras are 1 megapixel resolution having a wide horizontal field of view (FOV) of 190°. These cameras are designed to provide optimal near-field sensing upto 10 metres and slightly reduced perception upto 25 metres. The images are captured at a frame rate of 30 fps. The camera has a HDR sensor with a rolling shutter and a dynamic range of 120 dB. It has features including black level correction, auto-exposure control, auto-gain control, lens shading (optical vignetting) compensation, gamma correction and automatic white balance for color correction.

SOCs: The trailer vision algorithms may either run on a standalone microprocessor dedicated for the trailer assist system or be part of a larger SOC which is shared for other systems like parking assist or highway driving. The typical automotive SOC vendors include Texas Instruments TDAx, Renesas V3H and Nvidia Xavier platforms. All of the SOC vendors provide accelerators specialized for deep learning which will be useful for deployment of our proposed algorithm. As our work is targeted for industrial deployment, there are computational bounds available for the design of the algorithms due to cost, power consumption and heat dissipation.

Software Architecture: Images captured from cameras are usually pre-processed before sending them to computer vision algorithms. These pre-processing includes distortion correction, contrast enhancement and de-noising *etc*. Computer vision algorithms generally perception algorithms detect objects, understand scene and feed the information to high level application layer to plan maneuvering of the vehicle.

Trailer assist system shown in Figure 5 takes rear camera feed, vehicle odometry and user desired trailer orientation as input and outputs steering angle (for jackknifing prevention, overlays *etc*.) and emergency braking command (for collision avoidance). Trailer assist system comprises of two major functional blocks perception unit and control unit. Perception unit contains CNN-LSTM model that detects a trailer and helps estimate trailer. It also contains tow ball localization and hitch couple localization algorithms. This data is fed to control unit. Control unit estimates trailer length and facilitates kink angle control, jackknife prevention, collision avoidance *etc*. Finally, brake and steering angle commands are sent to vehicle control and planning unit.

In this work, we are focused on developing a standalone trailer assist system. However in many cases, there are other visual perception algorithms like semantic segmentation, depth estimation and motion estimation already available for automated driving as shown in Figure 7. In this case, the proposed CNN model can be integrated in a multi-task framework by leveraging the larger encoder available in the system [20, 4]. Depth and Motion estimation will also greatly help in achieving better accuracy of trailer assist algorithms.

3. Proposed CNN+LSTM model

A standard approach to trailer detection would be to use a handcrafted features followed by a binary classifier. To smoothen the predictions over the time a Kalman filter can be used. Analogous to this approach we used CNN followed by an LSTM network shown in Figure 6 to perform detection and tracking simultaneously.

Spatio-temporal Model: The proposed architecture consists of two sub-models: A CNN model for deep feature extraction and detection of trailers at multi-scale over a single image and LSTM model for interpreting the features across time steps. The CNN model is only capable of handling a single image, transforming image pixels into an deep representation. These features across multiple images allow the LSTM to build up an internal state and update its weights. As the trailer will have a consistent temporal structure in the sequence of input images, the LSTM can help to fill gaps if accurate detection is not possible over a single image due to occlusions, motion blur, shadows and

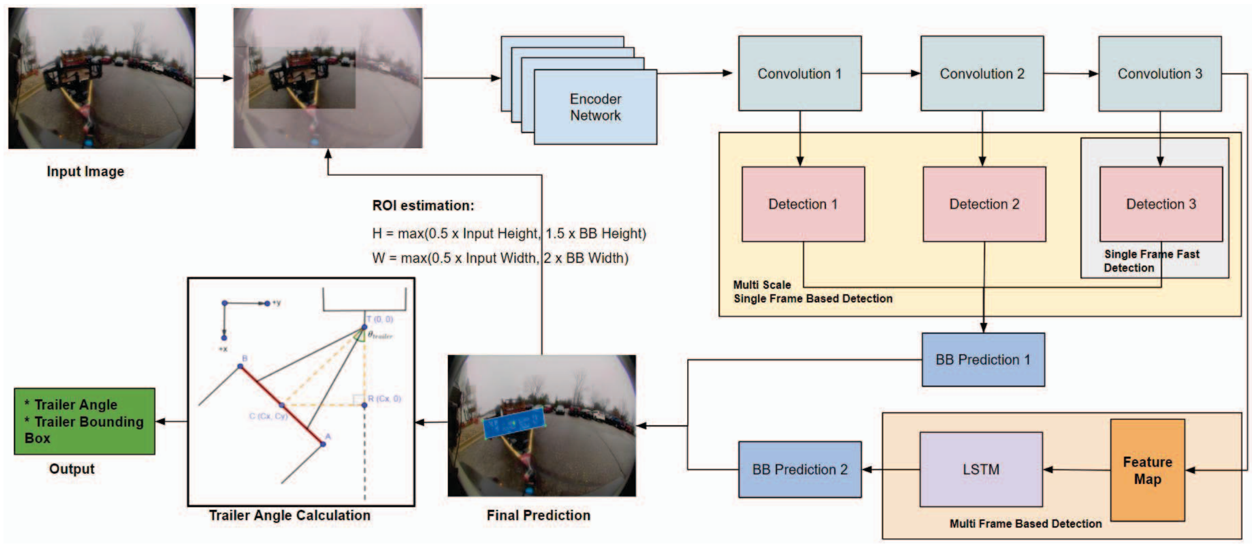

Figure 6: Proposed CNN architecture consisting of a single encoder and multi-scale detection decoder with LSTM. The final predicted bonding box is used to limit the image region for processing for the next frame. The final output of the network is the Trailer Angle and Bounding Box to localize the detected trailer.

Table 1: Details of proposed Deep Learning Architecture

Layer	**Input**	**Output**	**Details**
ROI Select	1280 x 800	H,W	based on Equation 1
Encoder	H X W x 3	H/8 X W/8 x 128	ResNet-10
Convolution 1	H/8 X W/8 x 128	H/8 X W/8 x 256	Conv, PReLU and BatchNorm
Convolution 2	H/8 X W/8 x 128	H/16 X W/16 x 512	MaxPool, Conv, PReLU and BatchNorm
Convolution 3	H/16 X W/16 x 512	H/32 X W/32 x 1024	MaxPool, Conv, PReLU and BatchNorm
Detection 1	H/8 X W/8 x 256	G_h X G_w X 5	YOLO V2 style decoder
Detection 2	H/16 X W/16 x 512	G_h X G_w X 5	YOLO V2 style decoder
Detection 3	H/32 X W/32 x 1024	G_h X G_w X 5	YOLO V2 style decoder
Feature Map	H/32 X W/32 x 1024	H/64 X W/64 x 64	Reduce depth prior to LSTM
LSTM	H/64 X W/64 x 64	1X5	Temporally Smoothen detection

severe lighting conditions.

Multi-scale detection algorithms [11] have proven to be more efficient than single scale detection counterparts. Trailer physical dimensions and shapes varies based on the manufacturer and purpose of use, hence our CNN model performs trailer detection at three scales. At each bottle-neck layer we perform bounding box detection similar to YOLO [14].

Prior Knowledge based Region of Interest (ROI): Trailers are always behind the vehicle and have restricted movement in the image. Once we detect the trailer in the first few frames after turning on the system, it is safe to assume that the variations in trailer positions are quite minimal. Taking this prior knowledge into consideration, we define a region of interest to narrow down the search window during the inference over the time. A complete image is passed through the network only during the first few initial frames when the ego vehicle start moving from ideal state. Once the trailer got localized we process over the specified ROI only. By various trails and observations we have found the following ROI criteria yields optimal performance and meets run-time constraints of our system.

$$ROI = max(0.5 \times In_H, 1.5 \times BB_H) \times max(0.5 \times In_W, 2.0 \times BB_W) \quad (1)$$

where In_H, In_W (Input height and width) are number of pixels in X and Y directions in image plane. BB_H, BB_W are BBox height and width of the detected trailer.

CNN Module: Convolutional Module consists of an encoder module followed by 3 convolution layers and a detection module connected to these layers. The encoder module is a Resnet-10 [7] architecture with each bottle neck layer consisting of a Convolution, Padding, Convolution with stride, Skip Connection, Linear Activation, PReLU [6], Batch Normalization [9], Concatenation and Addition layers. A detection module is similar to YOLO [14]. We regress for bounding box width, height and bottom center co-ordinates in the image. A class agnostic object confidence score is inferred at each grid and at each scale. The grid size varies across each scale to keep total grid count same across the different scales even though the input to the detection module changes in resolution. This brings us two advantages, one it keeps the run-time constant across the scales and two it reduces the false positives as the sampling frequency at initial layers is less than further layers.

CNN Architecture Details: Table 1 shows architecture details of the proposed CNN module. Initial frames are processed at full resolution, once consecutive frames show high confidence on bounding box predictions, ROISelect is enabled and cropped ROI region is resized to have a resolution as multiples of 32 for computational ease. All the detection outputs are passed through a non maximum suppression algorithm to choose one final detection bounding box. Unlike a traditional NMS algorithm in YOLO, which produces per grid optimal detections, our implementation takes multiple inputs from different scales and produces a unique bounding box proposal per image, as probability of having multiple trailers with maximum area of occupancy is almost zero. Feature Map layer is used to reduce the feature dimension prior to LSTM module.

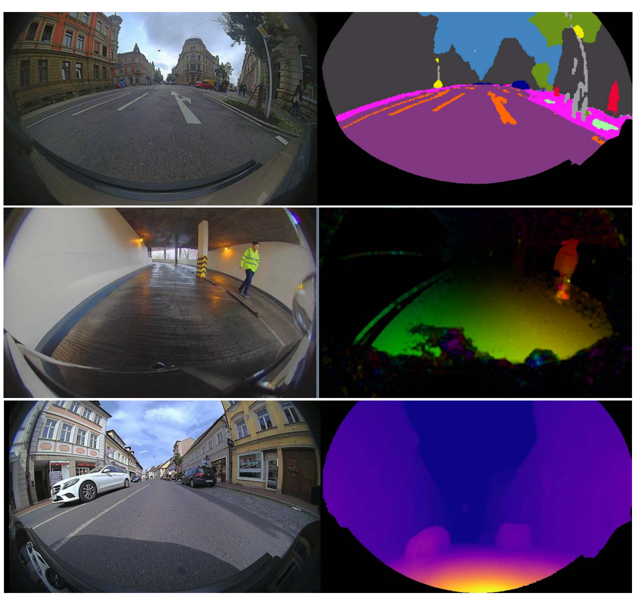

Figure 7: Typical perception algorithms on rear-view camera. From top to bottom: Semantic Segmentation, Motion Estimation and Depth Estimation

LSTM Module: The LSTM module consists of a zero padding layer to convert the output to a fixed length vector as input ROI dimensions changes during the run time. A feature vector length is fixed to 16640. This is followed by an LSTM layer with 5 output units. These 5 units regress for bouning box dimensions and co-ordinates similar to detection decoder modules. Now the final predictions from detection module and LSTM are passed through a NMS algorithm to produce a single robustly detected bounding box over the trailer.

Integration of Hitch Couple and Tow Ball Localization: The first step in trailer usage is to attach the trailer to the vehicle. To enable this, hitch couple and tow ball localization algorithms are required. To accomplish this, we use the same trailer detection algorithm to locate the trailer. Once a minimum distance criteria is met while backing to the parked trailer we switch the operating to model to localize the hitch couple and tow ball. This is achieved by replacing detection decoder in our CNN-LSTM model with another one which detects hitch couple and tow ball. For practical reasons, we reuse the same encoder block of the CNN-LSTM model and tune the detector for the new object size.

4. Results and Discussion

There are three independent datasets for the three tasks namely Trailer Angle estimation, HCL and TBL. Trailer angle estimation is the main task as it runs all the time whenever the trailer is connected and thus it has a larger dataset. It comprises of 1400 images extracted using 11 different types of trailers using sampling strategy discussed in [21]. The scenes contain different environmental conditions including daylight, shadows & rain. The driving surface had both asphalt roads and grass. Training/Validation/Test split is of the ratio 60/15/25. HCL and TBL have their own datasets of 500 images respectively having the same ratio of

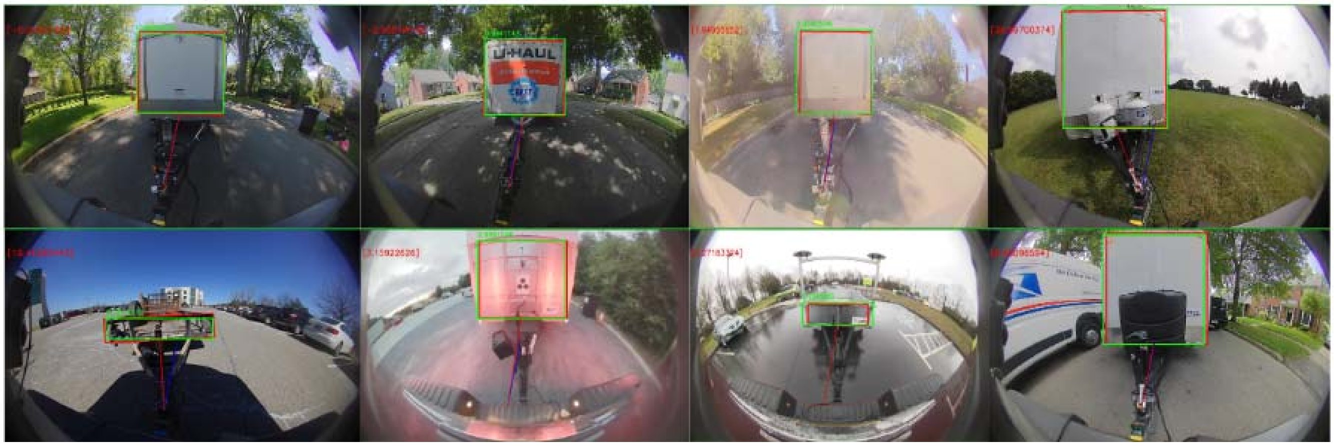

Figure 8: Qualitative results of Trailer detection and articulation angle estimation. Red box is ground truth and green box is estimated. Red line denotes vehicle axis and blue line denotes trailer articulation.

dataset split. Ground truth for detection of trailer, hitch coupler and tow ball were done manually. Trailer angle ground truth was generated using a rotary encoder.

For trailer detection, True positive rate obtained is 0.86 for an IoU threshold of 0.7 and True Negative rate is 0.98. For trailer angle estimation, 87% of the time the estimation was accurate within a tolerance of 1°. Figure 8 illustrates accurate detection for various trailers and environmental conditions. For HCL, True positive rate obtained is 0.72 for an IoU threshold of 0.7 and True Negative rate is 0.98. For TBL, True positive rate obtained is 0.82 for an IoU threshold of 0.7 and True Negative rate is 0.98. For HCL, True positive rate obtained is 0.72 for an IoU threshold of 0.7 and True Negative rate is 0.98. HCL is more challenging especially in far field because of the small size of the object.

Technical Challenges: We briefly list the practical challenges involved in deploying this system based on our experience.

- Trailer appearance is not standardized and it is difficult to include all possible types to get a robust system.
- The achievable angle accuracy output is limited by the camera resolution because fisheye camera has high angular deviation per pixel.
- Small size of hitch coupler causes mis-detections and false positives.
- Reflection of brake light from trailer body can cause misclassification.

5. Conclusion

In this paper, we provided a high level overview of a trailer assist system and the main visual perception modules. We created a dataset for deep learning trailer detection and articulation angle estimation tasks. We proposed an efficient CNN and LSTM model to detect and track the trailer and its angle and obtained a high accuracy. Finally, we discussed the results and current challenges. We also plan to release the dataset to encourage more research in this area.

ACKNOWLEDGMENT

The authors would like to thank their employer for the opportunity to work on fundamental research. We would also like to thank our colleague Gerald Koudijs and Enrique Romay Castineira for providing a few images used in the paper.

References

[1] Y. Atoum, J. Roth, M. Bliss, W. Zhang, and X. Liu. Monocular video-based trailer coupler detection using multiplexer convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5477–5485, 2017. 2

[2] A. Briot, P. Viswanath, and S. Yogamani. Analysis of efficient cnn design techniques for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 663–672, 2018. 1

[3] L. Caup, J. Salmen, I. Muharemovic, and S. Houben. Video-based trailer detection and articulation estimation. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184. IEEE, 2013. 2

[4] S. Chennupati, G. Sistu, S. Yogamani, and S. A Rawashdeh. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 4

[5] C. de Saxe and D. Cebon. Measurement of articulation angle by image template matching. *Proceedings of the Institution*

of Mechanical Engineers, Part D: Journal of Automobile Engineering, page 0954407019833819, 2019. 1

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 6

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[8] J. Horgan, C. Hughes, J. McDonald, and S. Yogamani. Vision-based driver assistance systems: Survey, taxonomy and advances. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2032–2039. IEEE, 2015. 1

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 6

[10] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech. Monocular fisheye camera depth estimation using sparse lidar supervision. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2853–2858. IEEE, 2018. 1

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 5

[12] O. Ljungqvist, N. Evestedt, D. Axehill, M. Cirillo, and H. Pettersson. A path planning and path-following control framework for a general 2-trailer with a car-like tractor. *arXiv preprint arXiv:1904.01651*, 2019. 2

[13] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, and S. Yogamani. Visual slam for automated driving: Exploring the applications of deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 247–257, 2018. 1

[14] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 5, 6

[15] M. Rezaei and R. Klette. *Computer vision for driver assistance*. Springer-Cham Switzerland, 2017. 1

[16] Y. Ro and Y. Ha. A factor analysis of consumer expectations for autonomous cars. *Journal of Computer Information Systems*, 59(1):52–60, 2019. 1

[17] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017. 1

[18] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand. Rtseg: Real-time semantic segmentation comparative study. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1603–1607. IEEE, 2018. 1

[19] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab. Modnet: Motion and appearance based moving object detection network for autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864. IEEE, 2018. 1

[20] G. Sistu, I. Leang, S. Chennupati, S. Milz, S. Yogamani, and S. Rawashdeh. NeurAll: Towards a unified model for visual perception in automated driving. *arXiv preprint arXiv:1902.03589*, 2019. 4

[21] M. Uricar, D. Hurych, P. Krizek, and S. Yogamani. Challenges in designing datasets and validation for autonomous driving. *arXiv preprint arXiv:1901.09270*, 2019. 6

[22] M. Uricar, P. Krizek, G. Sistu, and S. Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. *arXiv preprint arXiv:1905.01492*, 2019. 1

[23] L. Xu, E. Tseng, T. Pilutti, and S. Schondorf. Yaw rate based trailer hitch angle estimation for trailer backup assist. Technical report, SAE Technical Paper, 2017. 2

[24] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uřičář, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Perez. WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving. *CoRR*, abs/1905.01489, 2019. To appear in ICCV 2019. 1

[25] Ford's Trailer Hitch Guidance Overlays. https://www.digitaltrends.com/cars/2018-ford-f-150-specs-release-date-price/. Accessed: 2019-09-21. 1

[26] Valeo's Trailer See through system. https://www.valeo.com/en/world-premiere-at-ces-2019-of-valeo-xtravue-trailer-the-invisible-trailer-system/. Accessed: 2019-09-21. 1