2020-01-0136 Published 14 Apr 2020

LiDAR and Camera-Based Convolutional Neural Network Detection for Autonomous Driving

Ismail Hamieh, Ryan Myers, Hisham Nimri, and Taufiq Rahman National Research Council Canada

Aarron Younan, Brad Sato, Abdul El-Kadri, Selwan Nissan, and Kemal Tepe University of Windsor

Citation: Hamieh, I., Myers, R., Nimri, H., Rahman, T. et al., "LiDAR and Camera-Based Convolutional Neural Network Detection for Autonomous Driving," SAE Technical Paper 2020-01-0136, 2020, doi:10.4271/2020-01-0136.

Abstract

Autonomous vehicles are currently a subject of great interest and there is heavy research on creating and improving algorithms for detecting objects in their vicinity. A ROS-based deep learning approach has been developed to detect objects using point cloud data. With encoded raw light detection and ranging (LiDAR) and camera data, several basic statistics such as elevation and density are generated. The system leverages a simple and fast convolutional neural network (CNN) solution for object identification and localization classification and generation of a bounding box to detect vehicles, pedestrians and cyclists was developed. The system is implemented on an Nvidia Jetson TX2 embedded computing platform, the classification and location of the objects are determined by the neural network. Coordinates and other properties of the object are published on to various ROS topics which are then serviced by visualization and data handling routines. Performance of the system is scrutinized with regards to hardware capability, software reliability, and real-time performance. The final product is a mobile-platform capable of identifying pedestrians, cars, trucks and cyclists.

Introduction

The human body is littered with sensors detecting visible cues, noises, odors and physical sensations. Billions of neurons process sensory information and dispatch discoveries to higher levels in our brain. It is the visual sense that gathers most information compared to all other senses [1]. In automotive, drastic advancements in sensor, perception, and on-board computation technologies in recent years resulted in a mature ecosystem where development of autonomous driving (AD) as a mass consumer product/service has never been a more realistic goal. AD can be characterized as the task of developing a deep understanding of the surrounding environment (i.e., perception) in terms of ranging and classification of its static and dynamic elements to travel from point A to point B while safely responding to the constraints imposed by these elements (i.e., path planning). Static elements of roadway environments include spatially fixed objects such as lane markings, traffic signs, landmarks, road edges etc., while the dynamic elements are characterized by moving objects such as pedestrians, motorists, cyclists, etc. [2]. For moving objects, camera or LiDAR sensors can either be used. At a high frame-rate, cameras can work and provide dense information over a long range under good illumination and fair weather. However, being passive sensors, they are strongly affected by the level of illumination. Clearly, the process depends on the amplitude and frequency of the light waves, influencing the overall result, while a reliable system should be invariant with respect to changes in illumination [3]. LiDARs sense the environment by using their own emitted pulses of laser light and therefore they are only marginally affected by the external lighting conditions. Furthermore, they provide accurate distance measurements. However, they have a limited range, typically between 10 and 100 m, and provide sparse data [4].

Based on the benefits and drawbacks of these two sensors, the combination of both can positively contribute to the perception of the sensed data. The effective alignment (either spatially, geometrically or temporally) of multiple heterogeneous sensor streams, and utilization of the diversity offered by multi-modal sensing is referred to as sensor data fusion [5]. Sensor data fusion is not only relevant to the AV domain [6], but also applicable in different fields such as surveillance [7], smart guiding glasses [8], and hand gesture recognition [9]. Overcoming heterogeneity of different sensors through effective utilization of redundancy across the sensors is the key to fusing different sensor streams.

LiDAR and camera fusion is often tackled through utilization of the common dimension of depth in the two modalities. In comparison, fusing LiDAR with camera image is nontrivial as there is no common dimension of depth, as there is no way to capture depth in a monocular camera. To overcome those challenges, we are working toward fusing LiDAR data with camera image and utilizing techniques that goes beyond a simple geometric calibration. We are working on developing a robust fusion algorithm that can enable a robot to make decisions under uncertainty.

This paper reports precursory work completed at the Automotive & Manufacturing and Innovation Hub of National

Research Council Canada (NRC) at London, Ontario in collaboration with University of Windsor at Windsor, Ontario to help Canadian enterprises build market-ready mobility products.

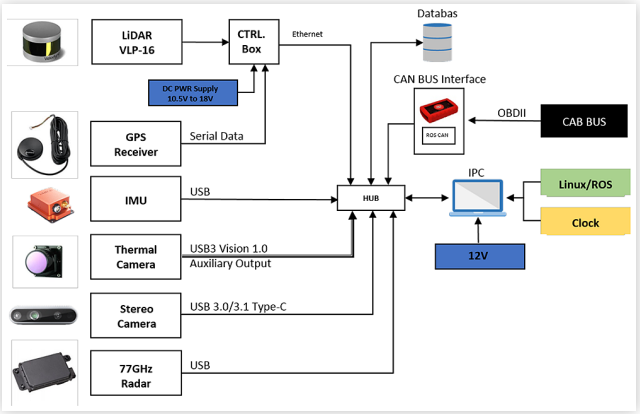
System Integration

DAQ Platform

In an effort to simulate an autonomous driving platform, a 2018 Ford Edge has been retrofitted with a Data Acquisition Suite in order to capture normal Canadian operating conditions and environments. For map creation, we used a 3D LiDAR, an IMU, a laptop running Robot Operating System (ROS), and a Global Positioning System (GPS) receiver. However, the system also includes other sensors that are utilized for other activities such as testing and evaluation of active safety and AV features. Figure 1 showcases the vehicle and sensor suite used within this study. Figure 2 presents an overview of the system used to capture data of the surrounding environment. We utilized the LiDAR, IMU, and the GPS to generate a map but we also used other sensors for different applications. The architecture has been described and is shown in [2].

FIGURE 1 Data Acquisition Hardware Stack build on top of 2018 Ford Edge

FIGURE 2 DAQ Block Diagram



Processing Platform

The resources available when operating in a battery-powered environment are, by nature, limited which makes it a challenge to effectively deploy neural computing tasks such as real-time inference in a real-world scenario. Power consumption on GPU-enabled mobile industrial computers is on the order of 280W-1kW [10, 11]. We set out to architect a portable object detection system using an Intel RealSense Camera and a Nvidia Jetson TX2 board with inter-process communication effected through ROS. Basic computation is carried out on the Nvidia's ARM integrated SoC while CUDA is used to parallelize the matrix operations inherent to neural computing as well as accelerate the image processing computations on the integrated Pascal GPU.

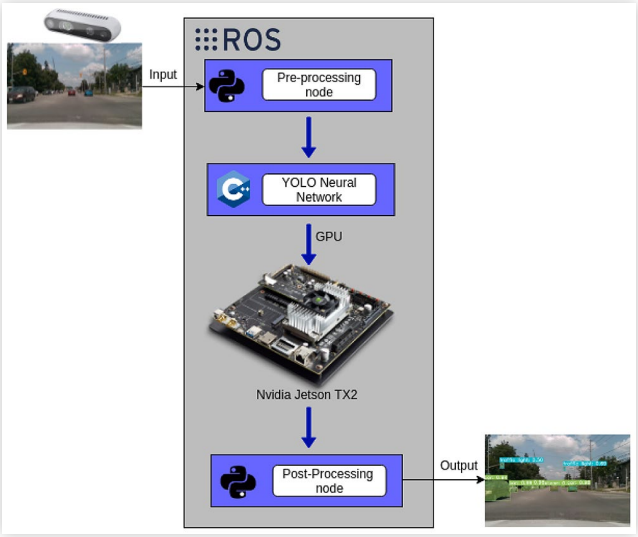
Mobile-based industrial computers exist however they exhibit a large gap in power consumption and pricepoint for GPU-enabled systems relative to the Jetson platform (i.e. 280W Max. draw on a Neuosys Nuvo-7164GC [10] vs. 90W Max. draw for the Jetson TX2 [12]).

2D Subsystem

Sensor The Intel RealSense Depth Camera D415 Series uses stereo vision to calculate depth. The D415 is a USB-powered depth camera and consists of a pair of depth sensors, RGB sensor, and infrared projector. It was ideal for our project. Refer to Figure 2 for full DAQ architecture. Since ROS is language-neutral we were able to seamlessly integrate the YOLOv3 Neural Network with our Python nodes.

ROS Processing Chain The ROS processing chain is illustrated in Figure 3. A preprocessing node ingests data from the Intel D415 and reformats it in such a way that it can

FIGURE 3 Portable Object Detection System



be passed to the object detection stage. Once the inferences are made, they are passed to a final stage which handles routing the prediction data to different services (i.e. visualization, data storage for SLAM applications, etc.).

Detection Backend Unlike traditional state-of-the-art object detection algorithms which re-purpose classifiers to perform detection, the YOLO (You Only Look Once) object detection system utilizes regression in order to spatially separate bounding boxes and associated class probabilities. This allows a single Neural Network to localize and classify objects in one evaluation by splitting an image into an SxS grid. The result is an object detection algorithm that is 1000x faster than R-CNN and 100x faster than Fast R-CNN [13]. As we were operating in an embedded environment, the choice of detection backend had to take into account the relatively limited resources available to us compared to a traditional desktop environment. As such, we utilized the object detection framework YOLOv3 with the TinyDarknet framework. The model was trained on single GPU leveraging the Berkeley Deep Drive (BDD) dataset. The parameters of the training process (i.e. learning rate, epochs, batch sizes, etc.) were chosen taking into account the processing limitations imposed by the embedded environment.

Training/Tesing Results The small model size of the TinyDarknet helps fit the entire model onto the on-chip cache which provides the mobile CPU with the ability to detect lane markings at *28 frames per second* (processing time of 36ms) on the Jetson TX2 platform. The post-training evaluation scores are expressed in terms of a metric called *Mean Average Precision* (mAP) which is the integral of the precision and recall plotted against each other. Precision and Recall are (inverse) indicators of the network's tendency to throw False Positives and Negatives respectively. The post-training evaluation classification mAP was 0.28 across the classes.

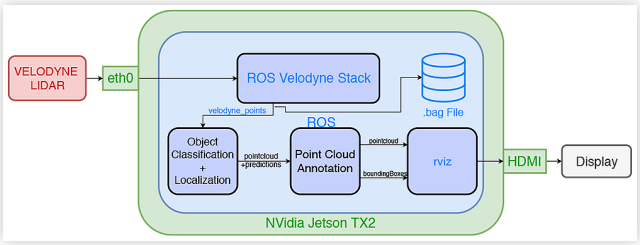
3D Susbystem

Sensor The Velodyne Puck is a 16 channel LiDAR with a 100m range and a rotation rate of up to 5-20Hz [14]. It is connected to the Jetson TX2 via an Ethernet connection and the data stream is parsed by the ROS Velodyne Stack [15].

ROS Processing Chain A total of four ROS packages, including the Velodyne Stack were employed in the processing chain from input-to-display (see Figure 4), two of which were custom developed for the 3D subystem. The processing stages are as follows:

- *Input Stage*: The ROS Velodyne Stack is a driver used as an interface between the UDP datagrams sent from the LiDAR and frames of usable point cloud data (i.e. point cloud *messages*). This is the acquisition stage of the chain.
- *Object Classification+Localization Package*: This stage acts as an interface between the incoming point cloud *messages* and the object detection network (see section 3.4) and its purpose is localization and identification of objects in the point cloud frame. This package sends a *message* with the location (in metres) and class details of all of the objects which were inferred to be in the point cloud.[1]
- *Annotation Stage*: This stage consists of a routine which listens for messages from the inference stage and generates the messages describing the shape and location of the bounding boxes for visualization. It is responsible for ensuring that the correct set of bounding boxes are paired with the correct point cloud frame.[1]
- *Visualization Stage*: This stage was simply *rviz*, a ROS Point Cloud visualization environment, waiting for the synchronized bounding box and point cloud messages from the Annotation Stage.

FIGURE 4 LiDAR Detection Platform Overview



Detection Backend The object detection backend employed for the LiDAR data was the *Sparsely Embedded Convolutional Detection* (SECOND) model, a VoxelNet-based Convolutional Neural Network, consisting of five phases; Voxelization, Voxelwise Feature Extraction, Spatial Downsampling, Region of Interest Proposal and finally Classification+Localization [16, 17]. SECOND aims to exploit the sparse nature of 3D point clouds by implementation of a spatially sparse submanifold convolution in the Spatial Downsampling stage of the VoxelNet architecture which reduces inference time by a factor of approximately 4.5 compared to a vanilla VoxelNet implementation. This was deemed advantageous due to the embedded hardware (i.e. computationally restricted) environment on which the project was implemented. A number of modifications to the underlying CUDA code for Pytorch and Spconv (sparse convolution implementation) frameworks were required in order to successfully compile and run the network on the Pascal/Jetson TX2 architecture.

Training/Testing Results The model was trained for 315,000 steps on the LiDAR portion of the KITTI dataset, a collection of point clouds generated by a 64 channel LiDAR in an urban driving scenario [18]. Four classes of objects were trained and evaluated- Pedestrians, Cars, Cyclists, and Vans. For evaluation on the KITTI dataset, the mAP is measured at various levels of bounding box overlap; 25%-50% overlap for small objects like Cyclists and Pedestrians and 50%-70% overlap for Cars and Vans.

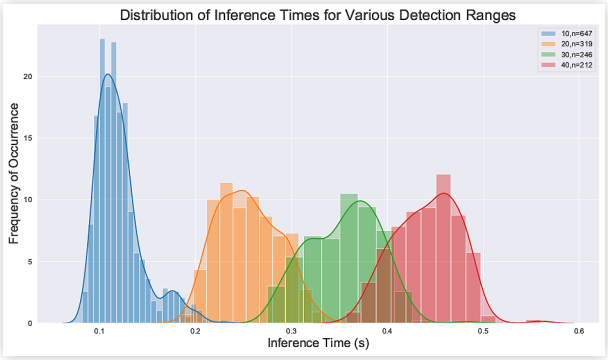
Additional, the trained model was deployed to the Jetson and the system was tested live with the Velodyne Puck in West Windsor, Ontario, Canada. Although an initial lag of approximately 3 seconds was exhibited, once started, the inference times, dependent on the spatial range over which inference is performed as there is more data to be ingested/processed, were a mean of *120-441ms* for a range of 10-40m respectively with variances of 0.6-1.1ms (see histogram of inference times in Figure 5) which corresponds to the mid-to-lower end of the

[1] Developed in-house

TABLE 1 mAP Scores for Target Objects at various levels of bounding box overlap and difficulty levels as laid out by criteria in the KITTI Vision Benchmarking Suite [18].

		0.25	0.5	0.7
Car	**Easy**	-	0.9084	0.8892
	Medium	-	0.898	0.7807
	Hard	-	0.8896	0.7633
Pedestrian	**Easy**	74.89	57.8	-
	Medium	0.7249	0.5049	-
	Hard	0.6581	0.4429	-
Cyclist	**Easy**	0.846	0.7791	-
	Medium	0.6405	0.5848	-
	Hard	0.6301	0.53	-
Van	**Easy**	-	0.5304	0.4847
	Medium	-	0.4186	0.3802
	Hard	-	0.3602	0.3203

FIGURE 5 Histogram illustrating Frequency of Occurrence of Inference Times vs. Inference Distance



of the LiDAR's rotation period. The overarching aim of the project leans less on the classifier performance as opposed to its integration and function in real-time situations on an embedded system, as such, qualitatively the authors judged the classification performance throughout the live test to be sufficient to proceed with the future fusion work, even with the factor of 4 reduction in point density from that of the training and evaluation dataset. Ground truth labels for the dataset collected from the live testing would need to be established in order to prove out hard classification performance metrics for our live tests.

Future Work

With the completion of the prototypes of each individual component, we will be working on two major topics; information fusion and cross-training.

In regards to information fusion, while both systems presented offer advantages and disadvantages when used in isolation, additional robustness can be achieved by fusion of both the LiDAR and camera based classification and detection platforms. There are many frameworks which have been proposed in literature such as hypothesis generation and validation schemes in [19], detection level fusion of a variety of sensors as presented in [20], or multilevel fusion as presented in [21]. While we have not selected a specific direction for fusion, we will be looking to ensure that the final solution is techno-economical and practical for vehicles of the future.

The other component of future work relates to cross-training. The underlying concept is to use frame transformations between the two sensors, mapping them to a global frame in which object classifications, positions, boxes, etc. can be transformed between the two sensors. If one system is well trained and able to detect objects and positions, the information can be serve as training data for a second unrelated sensor that now has temporally and spatially tagged data outlining the object and position. For instance, suppose Sensor A is a 2D RGB area scan camera connected to a localization model that has been proven to perform well. Sensor B, an infrared sensor whose FOV is identical/calibrated to that of Sensor A, can use the inference data (i.e. class and bounding box coordinates) provided by Sensor A as its ground truth database to train a network capable of detecting objects from infrared images without tedious annotation.

Clearly, the major advantage of the proposed system is the ability to cross-train additional sensor-based detection systems without having a native dataset for that particular sensor. Furthermore, dataset can easily be captured and annotated with minimal requirement for manual data labelling, cleaning and processing. While this system only showcases two major sensor components, we plan to expand this to other sensing technologies.

Conclusion

We successfully integrated a 2D object localization subsystem based on the YOLOv3 Neural Network using the TinyDarknet framework and a 3D object localization subsystem based on the SECOND 3D object detection system on to a low-cost GPU computing platform, the Nvidia Jetson TX2, with each independent system capable of inferring objects at rates comparable to the speed at which the data is being ingested. The integration of the detection backends onto an ARM-based embedded ROS environment called for the development of interface packages and modification of existing libraries in order carry out the inference operations in the reduced resource environment compared to that of the traditional desktop and cloud-based GPU computing platforms on which these detection frameworks were originally developed. The integrated subsystems were trained and evaluated on publicly available datasets and additionally tested live during driving tests on Canadian roadways for inference speed as well as qualitatively for accuracy in order to gauge the feasibility of the next step in integration, sensor fusion. We plan to improve detection results by combining both systems and using more powerful matching strategies to classification data in our training. Our objective is to continue looking for ways to bring different sources and structures of data together to make stronger models of the visual world.

References

1. Seiderman, A., Marcus, S.E., and Hapgood, D., *20/20 Is Not Enough: The New World of Vision* (Alfred A. Knopf, 1989).
2. Hamieh, I., Myers, R., and Rahman, T., "Construction of Autonomous Driving Maps Employing Lidar Odometry," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 1-4, May 2019.
3. Alvarez, J.M., Lopez, A., and Baldrich, R., "Illuminant-Invariant Model-Based Road Segmentation," in *2008 IEEE Intelligent Vehicles Symposium*, 1175-1180, IEEE, 2008.
4. Caltagirone, L., Bellone, M., Svensson, L., and Wahde, M., "Lidar-Camera Fusion for Road Detection Using Fully Convolutional Neural Networks," *Robotics and Autonomous Systems* 111:125-131, 2019.
5. Luo, R.C., Yih, C.-C., and Su, K.L., "Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions," *IEEE Sensors Journal* 2(2):107-119, 2002.
6. Choi, B.-S. and Lee, J.-J., "Sensor Network Based Localization Algorithm Using Fusion Sensor-Agent for Indoor Service Robot," *IEEE Transactions on Consumer Electronics* 56(3):1457-1465, 2010.
7. Dan, B.-K., Kim, Y.-S., Jung, J.-Y., Ko, S.-J. et al., "Robust People Counting System Based on Sensor Fusion," *IEEE Transactions on Consumer Electronics* 58(3):1013-1021, 2012.
8. Bai, J., Lian, S., Liu, Z., Wang, K. et al., "Smart Guiding Glasses for Visually Impaired People in Indoor Environment," *IEEE Transactions on Consumer Electronics* 63(3):258-266, 2017.
9. Erden, F. and Çetin, A.E., "Hand Gesture Based Remote Control System Using Infrared Sensors and a Camera," *IEEE Transactions on Consumer Electronics* 60(4):675-680, 2014.
10. Neousys, "Nuvo-7164gc Series Specifications," https://www.neousys-tech.com/Resource/Product_Document/Nuvo-7164GC/nuvo-7164gc-7166gc-intel-8th-9th-gen-nvidia-tesla-t4-p4-gpu-computing-datasheet.pdf, 2019.
11. Neousys, "Nuvo-8208gc Series," https://www.neousys-tech.com/en/product/application/edge-ai-gpu-computing/nuvo-8208gc-intel-8th-gen-dual-nvidia-rtx-2080ti-gpu-computing-platform, 2019.
12. MeanWell, "Gst90a 90w ac-dc High Reliability Adapter," https://www.meanwell.com/webapp/product/search.aspx?prod=GST90A, 2019.
13. Redmon, J. and Farhadi, A., "Yolov3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
14. Velodyne, "Velodyne Puck Datasheet," https://velodynelidar.com/products/puck/, 2019.
15. ROS, "Ros Support for Velodyne 3d Lidars," https://github.com/ros-drivers/velodyne, 2019.
16. Yan, Y., Mao, Y., and Li, B., "Second: Sparsely Embedded Convolutional Detection," *Sensors* 18:3337, Oct. 2018.
17. Yan, Y., "Second for Kitti/Nuscenes Object Detection," https://github.com/traveller59/second.pytorch, 2016-2018.
18. Geiger, A., Lenz, P., and Urtasun, R., "Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
19. Hwang, J.P., Cho, S.E., Ryu, K.J., Park, S. et al., "Multi-Classifier Based Lidar and Camera Fusion," in *2007 IEEE Intelligent Transportation Systems Conference*, 467-472, IEEE, 2007.
20. Chavez-Garcia, R.O. and Aycard, O., "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking," *IEEE Transactions on Intelligent Transportation Systems* 17(2):525-534, 2015.
21. Cho, H., Seo, Y.-W., Kumar, B.V., and Rajkumar, R.R., "A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 1836-1843, IEEE, 2014.

Contact Information

Ismail Hamieh
ismail.hamieh@nrc-cnrc.gc.ca

Ryan Myers
ryan.myers@nrc-cnrc.gc.ca

Hisam Nimri
mohd.nimri@nrc-cnrc.gc.ca

Taufiq Rahman, Ph.D.
taufiq.rahman@nrc-cnrc.gc.ca

Aarron Younan
younana@uwindsor.ca

Selwan Nissan
nissan2@uwindsor.ca

Abdulrahman El-Kadri
elkadr5@uwindsor.ca

Bradley Sato
satob@uwindsor.ca

Kemal Tepe, Ph.D.
ktepe@uwindsor.ca

Acknowledgments

We would like to thank NRC Strategic Advisor for Automotive and Surface Transportation Research Centre Mr. John Wood for his critical review and support in our project and our colleague Mr. Daniel Cheema for his great technical support in assembling the DAQ system and the sensors on the test vehicle. In addition, the contribution of Transport Canada, Government of Canada for supplying the research vehicle is gratefully acknowledged.

Definitions, Acronyms, Abbreviations

AV - Autonomous Vehicle

BDD - Berkeley Deep Drive

CNN - Convolutional neural network

DNN - Deep neural network
DAQ - Data Acquisition System
GPS - Global Positioning System
IMU - Inertial Measurement Unit
LiDAR - Light Detection and Ranging
ROS - Robot Operating System
SECOND - Sparsely Embedded Convolutional Detection
SLAM - Simultaneous Localization and Mapping
UDP - User Datagram Protocol
YOLO - You Only Look Once

ISSN 0148-7191