Implementation 2D Lidar and Camera for detection object and distance based on RoS

Leave this box blank

Please submit online <http://joiv.org/index.php/joiv/login> in DOC file  
Editor will not receive submission by email   
**Please be sure to check for spelling and grammar before submitting your paper.**

Agus Mulyanto1, Rohmat Indra Borman1, Purwono Prasetyawan1, A Sumarudin2

*1 Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Jl. ZA. Pagar Alam No.9 -11 Kota Bandar, Lampung, 35142, Indonesia   
 E-mail:* [*agus.mulyanto@teknokrat.ac.id*](mailto:agus.mulyanto@teknokrat.ac.id)*,* [*rohmat\_indra@teknokrat.ac.id*](mailto:rohmat_indra@teknokrat.ac.id)*,* [*purwono.prasetyawan@teknokrat.ac.id*](mailto:purwono.prasetyawan@teknokrat.ac.id)

2Informatics Department, Politeknik Negeri Indramayu, Jl. Lohbener Lama No 08, Indramayu, 45252, Indonesia   
E-mail: [shumaru@polindra.ac.id](mailto:shumaru@polindra.ac.id)

Abstract— the advanced driver assistance systems (ADAS) are one of the issues to protecting people from vehicle collision. Collision warning system is a very important part of ADAS to protect people from the dangers of accidents caused by fatigue, drowsiness and other human errors. Multi-sensors has been widely used in ADAS for environment perception such as cameras, radar, and light detection and ranging (LiDAR). We propose the relative orientation and translation between the two sensors are things that must be considered in performing fusion. we discuss the real-time collision warning system using 2D LiDAR and Camera sensors for environment perception and estimate the distance (depth) and angle of obstacles. In this paper, we propose a fusion of two sensors that is camera and 2D LiDAR to get the distance and angle of an obstacle in front of the vehicle that implemented on Nvidia Jetson Nano using Robot Operating System (ROS). Hence, a calibration process between the camera and 2D LiDAR is required which will be presented in session III. After that, the integration and testing will be carried out using static and dynamic scenarios in the relevant environment. For fusion, we use the implementation of the conversion from degree to coordinate. Based on the experiment, we result obtained an average of 0.197 meters

Keywords— lidar, Camera, object detection, distance detection, RoS.

1. Introduction

In the past few years, the advanced driver assistance systems (ADAS) are one of the issues to protecting people from vehicle collision. Collision warning system is a very important part of ADAS to protect people from the dangers of accidents caused by fatigue, drowsiness and other human errors. Multi-sensors has been widely used in ADAS for environment perception such as cameras, radar, and light detection and ranging (LiDAR)[1][2][3][4]. Cameras are widely used for object recognition, while radar and LiDAR are commonly used for distance measurement. LiDAR and radar have the advantage of long-distance measurements of objects in various conditions such as cloudy, rain, day and night[5]. However, this capability is not sufficient to perform environmental analysis. LiDAR or radar are often combined with cameras to perform environmental perception recognition. In driver assistance systems, cameras are widely used for traffic sign recognition[6], real-time detection and tracking of pedestrians[7], vehicle detection[8], or forward collision and overtaking detection[5].

The combination of 3D LiDAR and cameras is the popular ways to build intelligent transportation systems or self-driving cars or ADAS. However, the expensive hardware cost of 3D LiDAR is both a drawback and a barrier. To reduce this cost, LiDAR 2D is a solution in developing a low-cost driver assistance system. However, 2D LiDAR has one disadvantage which is that it only scans single horizontal line. The intrinsic and extrinsic calibration from both sensors are required to produce data accurately[1]. In addition, the relative orientation and translation between the two sensors are things that must be considered in performing fusion

Herein, we discuss the real-time collision warning system using 2D LiDAR and Camera sensors for environment perception and estimate the distance (depth) and angle of obstacles. In this paper, we propose a fusion of two sensors that is camera and 2D LiDAR to get the distance and angle of an obstacle in front of the vehicle that implemented on Nvidia Jetson Nano using Robot Operating System (ROS). Hence, a calibration process between the camera and 2D LiDAR is required which will be presented in session III. After that, the integration and testing will be carried out using static and dynamic scenarios in the relevant environment.

1. Related Work

Advanced driver-assist systems (ADAS) are a system that provides information from the car surrounding environment to assist drivers avoid accidents by providing alerts[7]. When ADAS detects an object around the car that has the potential to cause an accident, ADAS will give a warning to the driver by a sound and light indicator will light up or activate automatic braking. Various sensors are used by ADAS to map and recognize objects around cars such as LiDAR, Radar, Cameras, Ultrasonic Sensors, and Global Navigation Satellite System (GNSS)[9]. Sensor fusion has been done to produce a more accurate and robust detection. The combination of lidar and camera have been introduced for object or vehicle detection[10][11], object distance estimation[1], object tracking[2], collision avoidance system[12], and Autonomous Navigation[13] on different levels of data fusion. Cameras generally have a higher resolution than LiDAR but cameras have limited field of view and accurately estimate object distances. The Fusion technique is used as a correspondence between the points detected by the LiDAR and the points detected by the camera. The first step to integrating LiDAR and camera is to perform extrinsic calibration between sensors[1]. This implies that the geometric parameters of a sensor such as position and orientation must be determined taking into account the other sensors [14]. The key point of heterogeneous sensor fusion is identifying the features of each sensor and determining the geometric parameters from different angles and positions between sensors[15].

In recent years, deep learning algorithms are being applied to ADAS tasks. Convolutional Neural Networks (CNN) is a deep learning algorithm that  has been widely used for object detection systems in ADAS because its performance has outperformed traditional methods such as Viola Jones Detectors[16], Histogram of Oriented Gradients (HOG)[17], and Deformable Part-based Model (DPM)[18] in speed and accuracy [7]. Broadly speaking, deep learning for object detection can be grouped into two approach : “CNN based two-stage detection” and “ CNN based one-stage detection" [19]. R-CNN[20], SPP-Net[21], Fast R-CNN[22], Faster R-CNN[23], Feature Pyramid Networks (FPN)[24], and Mask R-CNN[25] are CNN based two-stage detection approach. Furthermore, CNN based one-stage detection there YOLO[26][27][28][29], SSD[30], and Retina-Net[31].The two-stage detector model has a better accuracy rate, but is slower than the one-stage detector[32]. Two-stage detector generate regions of interests use a Region Proposal Network (RPN) to classify objects and bounding box regression. On the other hand, the one-stage detector performs direct detection over a single pass through the neural network and bypasses the region proposal stage to learn the class probabilities and bounding box coordinates. SSD has advantages in terms of accuracy and detection speed for small objects compared to other models in their group.

ADAS development using deep learning algorithms and multi sensors requires parallel processing and high-speed processing to improve performance. This requires a processing machine equipped with a GPU and a supporting middleware framework. The Robot Operating System (ROS) could be a adaptable system for creating robot software[33]. ROS allows the plan of modular systems development and parallelized systems. ROS applications contain a collection of programs called nodes where each node will interact with each other through message passing. Two communication models are accessible in ROS: a subscriber-publisher model and a client-server [34]. The subscriber-publisher model uses one-way communication based on the concept of topics, and the client-server model uses two-way communication with the concept of service. ROS is a widely used platform for robotics implementations such as on the ADAS[35][13]. Another advantage of using ROS is that there are many active communities which has already created modules and drivers[35].

1. SYSTEM OVERVIEW

This research focuses on object detection, object distance measurement and object position for Advanced Driver Assistance System (ADAS). For this reason, ADAS will perform input data fusion from the camera sensor and 2D LIDAR sensor. The proposed ADAS architecture is shown in figure 1.

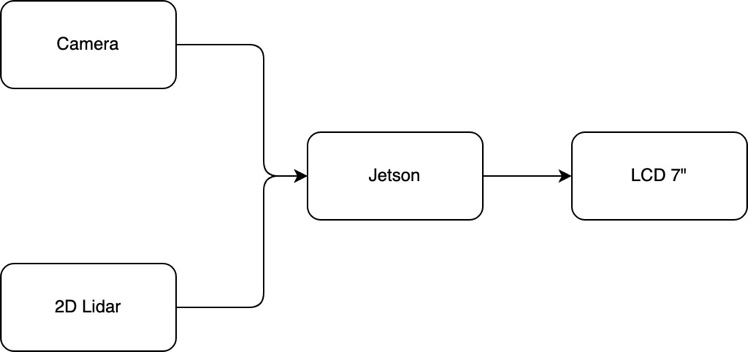


Figure 1. Architecture System ADAS

The ADAS system was developed using two sensors input such as the USB camera sensor (LOGITECH Webcam C170) and the 2D LIDAR sensor (SLIMTEC RP-LiDAR A1). Camera sensors are used for object detection purposes, and 2D LiDAR sensors will be used to measure object distances and object positions based on angles. ADAS using a single board computer NVIDIA Jetson Nano as the main processing unit and using the Robotic Operating System (ROS) as a platform. NVIDIA Jetson Nano uses a Quad-core ARM A57 @ 1.43 GHz CPU, 4GB Memory and 128-core GPU Maxwell. For the interface ADAS using 7 INCH HDMI IPS LCD.

Based on the architecture in Figure 1, processing data input from the camera sensor and LiDAR sensor, an ROS architecture is created by applying the subscriber-publisher model system whose architecture is depicted as in Figure 2. In this architecture, there are 2 nodes that will be able to work as a publisher or subscriber, namely the Object detection node and range /distance detection node. These two nodes will be connected to the ROS Master which will act as a broker.

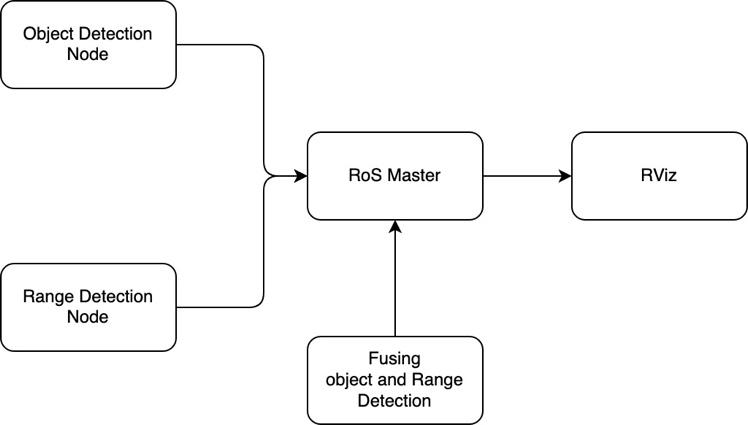


Figure 2. Architecture ROS

In Figure 2, the camera sensor will be connected to an object detection node whose function is to detect objects using SSD-Mobilnet version 2 [36] with a threshold of 0.5 which will produce an output in the form of object information in the form of a bounding box containing information on the coordinates of the x and y axes. The output of the object detection result by the Object detection node will be published to ROS Master using the topic "/jetson/result\_detection" which contains image data (width and height) in the form of pixel dots. Futhermore, the output from the RP-LiDAR sensor which has a reading radius of 360 degrees will be connected to the "Range Detection Node" which will be send the LiDAR reading data to the ROS Master using the topic "/scan\_msg".

* 1. Object Detection System

As mentioned earlier, one of the important parts of ADAS developed in this work is the object detection system. For the object detection system, the SSD-MobileNet V2 model is used to be precise the SSD300 [36], which uses an image input with a resolution of 300x300 and a dataset [37] which supports use on the NVIDIA Jetson Nano. The object class used from dataset [37] is seven classes as shown in Table 1.

Table1. Seven Class of Objects in [36] used for ADAS

|  |  |  |
| --- | --- | --- |
| **No** | **Class of Object** | **Image** |
| 1 | person |  |
| 2 | bicycle |  |
| 3 | car |  |
| 4 | motorcycle |  |
| 5 | bus |  |
| 6 | train |  |
| 7 | truck |  |

Meanwhile, the Mobilnet SSD architecture uses a convolution predictor for its detection using a single shoot multibook detector (SSD) with an input size of 680 x 480 pixels with a webcam using Filed of View (FoV) 60 and a resolution of 60 fps. Figure 3 shows the SSD300 architecture used for the object detection system.

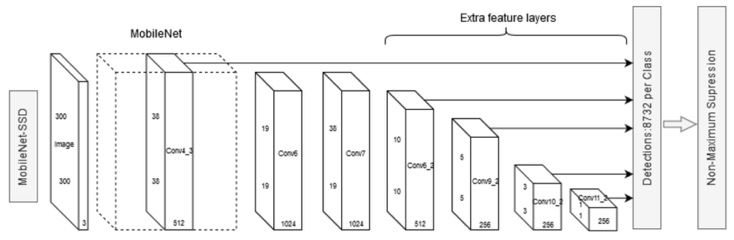


Figure 3. Mobilnet SSD300 Architecture[38]

* 1. Measure Object Distance and Angel

In implementing this system by detecting 3 labels, namely car, motorcycle and people.

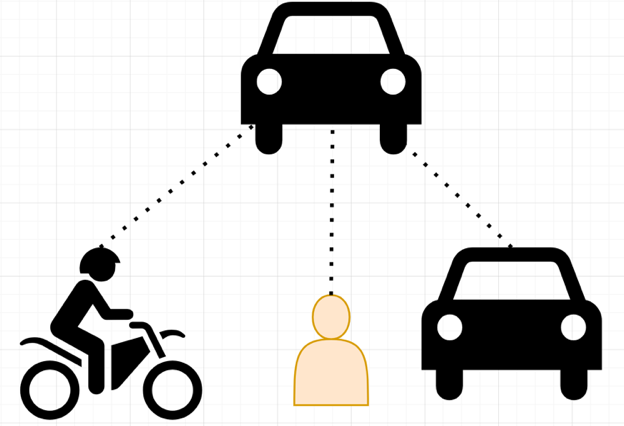


Figure 4. Object Detection

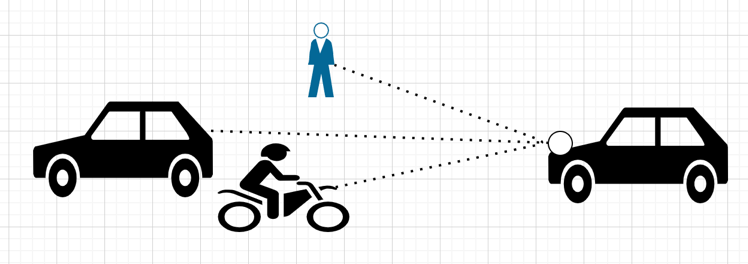
Installation of the system on the front dashboard of the car.

Figure 5. Implementation System

The lidar and camera are installed parallel to the lidar radius used 0-180o. by using the cover for detection at 180-360o degrees.

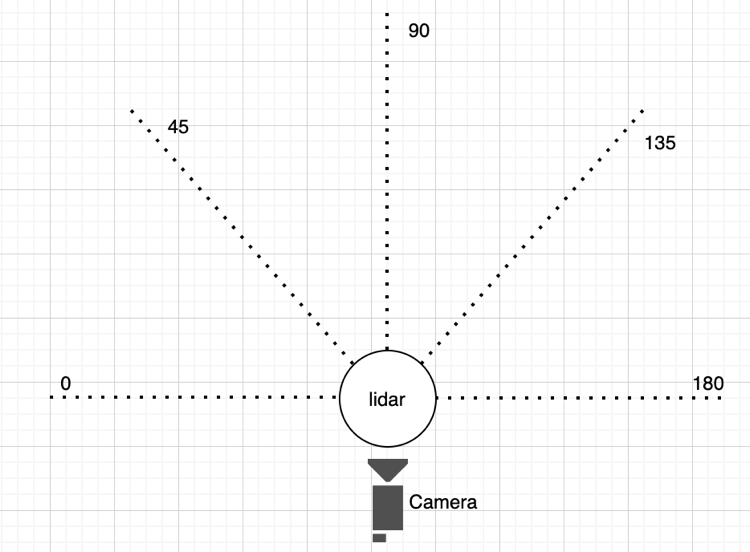


Figure 6. Lidar and camera installation

The following is the detection algorithm:

|  |
| --- |
| Algorithm 1: Object Detection |
| Object detection estimation with mobilnet  Input: Camara FoV 60o with 640x480  Output: Display object detection, width, height and center coordinate detection  Method:   1. Initial node RoS with ObjectDetection 2. Initial object detection parameter image, width and height 3. Detection clas id with mobilnet SSD 4. Detection coordinate left, right, bottom, width, height, area and center x and y coordinate 5. Publish detection with parameter image, width and height |

|  |
| --- |
| Algorithm 2: Range Detection |
| Range Detection with 2D Lidar with RPLidar  Input: RPLidar  Output: range detection  Method:   1. Initial node RoS with RangeDetection 2. Detection Angle 3. Convert Degree to Radian 4. Publish Range detection with parameter range and degree |

|  |
| --- |
| Algorithm 3: Fusion Object and Range Detection |
| Range Detection with 2D Lidar with RPLidar  Input: node ObjectDetection and node RangeDetection  Output: Display object and range detection  Method:   1. Subscribe Object detection and Range Detection 2. Initial range Radian to pixel 3. Detection object with center coordinate 4. Convert center coordinate with range to save data range 5. Publish range detection and range |

1. Experiment and RESULT

For this experiment, Figure 7 shows the installation of the ADAS prototype mounted on the front of the car. ADAS uses the ROS Library for reading distance data from the RPLiDAR measurement results obtained from [39].



1. (b)

Figure 7. Installation of the lidar sensor and camera on the ADAS prototype

The data obtained from the detection results of the object detection nodes and the range detection nodes are as follows:

t = 1602756357940128087 | data = (inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, 8.597999572753906, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, 8.57699966430664, inf, 3.565999984741211, 3.0840001106262207, 3.0339999198913574, 2.8399999141693115, 2.696000099182129, 2.677999973297119, 2.681999921798706, 2.885999917984009, inf, 8.833999633789062, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf)

Data deteksi object terdiri dari center cordinat, estimasi jarak dan label deteksi.

data deteksi

t = 1602756362772608041 | (x,y) = (236.691894531,272.439361572) | jarak = 0.641584333167 | label = motorcycle

t = 1602757073928807020 | (x,y) = (431.908874512,215.612228394) | jarak = 2.18477000313 | label = truck

t = 1602756751280932903 | (x,y) = (478.734436035,191.203140259) | jarak = 3.9323105664 | label = bus

t = 1602757064671120882 | (x,y) = (377.174682617,215.417694092) | jarak = 0.528737693536 | label = car

t = 1602757063625045061 | (x,y) = (316.203216553,210.66519165) | jarak = 1.91034458866 | label = person

table 3. pengujian estimasi jarak dan Jarak actual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| scenario | object | Actual distance [m] | Estimated distanced [m] | Error [m] |
| 1 | Car | 5.2 | 5.3 | 0.1 |
| Motor cycle | 5.2 | 5.8 | 0.6 |
| People | 5.2 | 5.3 | 0.1 |
| 2 | Car | 3 | 3.12 | 0.12 |
| Motor Cycle | 3 | 3.13 | 0.13 |
| People | 3 | 3.13 | 0.13 |
| Average |  |  |  | 0.197 |

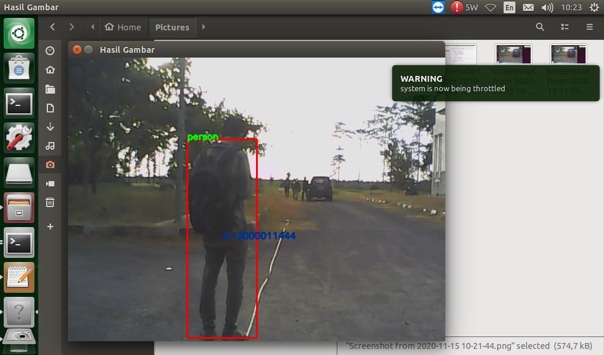
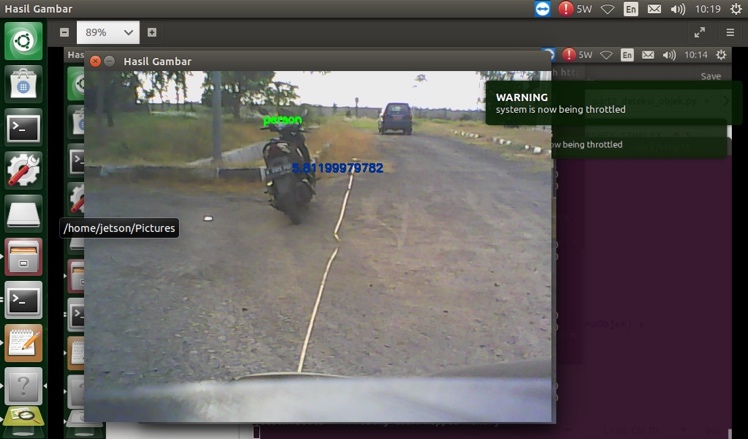
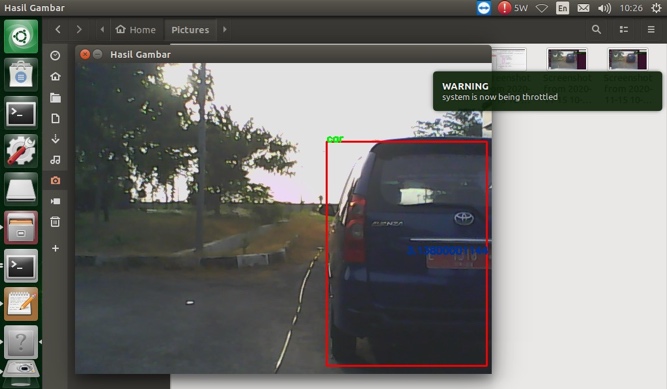


Figure 8. System Testing

1. Conclusions

The ADAS system that has been developed using NVIDIA Jetson Nano with the ROS platform and input from 2 sensors such as 2D LiDAR sensor which has a 360o range with a maximum distance of 12 meters and a Logitech USB Camera sensor has shown a good performance where the object detection system is able to work with 40 fps performance so it is suitable for real-time systems. Meanwhile, the fusion performance that is applied with the conversion from degree to coordinate has also shown satisfactory performance where from the experiment to detect car, motorcycle and people objects, the error rate is obtained with an average of 0.197 meters. this is very realistic considering the distance is still below 1 meter.

Acknowledgment

Thank you to The Directorate General of Higher Education, Ministry of National Education and Culture for the financial support related to Penelitian Kerjasama Antar Perguruan Tinggi (PKPT) through LPPM Teknokrat Indonesia University and LLDIKTI Region 2 with the contract number 939/SP2H/LT/JAMAK/LL2/2020.

References

[1] G. A. Kumar, J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon, “LiDAR and camera fusion approach for object distance estimation in self-driving vehicles,” *Symmetry (Basel).*, vol. 12, no. 2, 2020, doi: 10.3390/sym12020324.

[2] A. Rangesh and M. M. Trivedi, “No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles Using Cameras and LiDARs,” *IEEE Trans. Intell. Veh.*, vol. 4, no. 4, pp. 588–599, 2019, doi: 10.1109/TIV.2019.2938110.

[3] R. H. Rasshofer and K. Gresser, “Automotive radar and lidar systems for next generation driver assistance functions,” *Adv. Radio Sci.*, vol. 3, pp. 205–209, 2005, doi: 10.5194/ars-3-205-2005.

[4] Q. Li, L. Chen, M. Li, S. L. Shaw, and A. Nüchter, “A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, 2014, doi: 10.1109/TVT.2013.2281199.

[5] H. Y. Lin, J. M. Dai, L. T. Wu, and L. Q. Chen, “A vision-based driver assistance system with forward collision and overtaking detection,” *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–19, 2020, doi: 10.3390/s20185139.

[6] M. M. William *et al.*, “Traffic Signs Detection and Recognition System using Deep Learning,” *Proc. - 2019 IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2019*, pp. 160–166, 2019, doi: 10.1109/ICICIS46948.2019.9014763.

[7] A. Mulyanto, R. I. Borman, P. Prasetyawan, W. Jatmiko, and P. Mursanto, “Real-Time Human Detection and Tracking Using Two Sequential Frames for Advanced Driver Assistance System,” 2019, doi: 10.1109/ICICoS48119.2019.8982396.

[8] C. W. Lai, H. Y. Lin, and W. L. Tai, “Vision based ADAS for forward vehicle detection using convolutional neural networks and motion tracking,” *VEHITS 2019 - Proc. 5th Int. Conf. Veh. Technol. Intell. Transp. Syst.*, pp. 297–304, 2019, doi: 10.5220/0007626902970304.

[9] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, “Review of advanced driver assistance systems (ADAS),” *AIP Conf. Proc.*, vol. 1906, no. November, 2017, doi: 10.1063/1.5012394.

[10] X. Zhao Pengpeng Sun Zhigang Xu Haigen Min Hongkai Yu, “Fusion of 3D LIDAR and Camera Data for Object Detection in Autonomous Vehicle Applications,” 2020, [Online]. Available: https://scholarworks.utrgv.edu/cs\_fac/22.

[11] F. Zhang, D. Clarke, and A. Knoll, “Vehicle detection based on LiDAR and camera fusion,” *2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014*, pp. 1620–1625, 2014, doi: 10.1109/ITSC.2014.6957925.

[12] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, “LiDAR and camera detection fusion in a real-time industrial multi-sensor collision avoidance system,” *Electron.*, vol. 7, no. 6, 2018, doi: 10.3390/electronics7060084.

[13] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, “ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera,” *2019 1st Int. Symp. Instrumentation, Control. Artif. Intell. Robot. ICA-SYMP 2019*, pp. 151–154, 2019, doi: 10.1109/ICA-SYMP.2019.8645984.

[14] L. Zhou and Z. Deng, “A new algorithm for the extrinsic calibration of a 2D LIDAR and a camera,” *Meas. Sci. Technol.*, vol. 25, no. 6, 2014, doi: 10.1088/0957-0233/25/6/065107.

[15] W. Dong and V. Isler, “A novel method for the extrinsic calibration of a 2D laser rangefinder and a camera,” *IEEE Sens. J.*, vol. 18, no. 10, pp. 4200–4211, 2018, doi: 10.1109/JSEN.2018.2819082.

[16] M. J. J. Paul Viola, “Robust Real-Time Face Detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004, doi: 10.1112/jlms/s2-30.3.419.

[17] R. Rai, S. Shukla, and B. Singh, “Histograms of Oriented Gradients for Human Detection,” in *IEEE Computer Vision and Pattern Recognition, 2005*, 2005, vol. 1, no. CVPR 2005, pp. 886–893, doi: 10.1109/TII.2019.2950094.

[18] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Cascade object detection with deformable part models,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,San Francisco*, 2010, pp. 2241–2248, doi: 10.1109/CVPR.2010.5539906.

[19] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” pp. 1–39, May 2019, [Online]. Available: http://arxiv.org/abs/1905.05055.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, vol. 37, no. 9, pp. 1904–1916, doi: 10.1007/978-3-319-10578-9\_23.

[22] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[24] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[25] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[27] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[28] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv Prepr. arXiv1804.02767*, 2018.

[29] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[30] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *European Conference on Computer Vision – ECCV 2016*, 2016, vol. 9905, pp. 21–37, doi: 10.1007/978-3-319-46448-0.

[31] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2999–3007, 2017, doi: 10.1109/ICCV.2017.324.

[32] P. Soviany and R. T. Ionescu, “Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,” *Proc. - 2018 20th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2018*, pp. 209–214, 2018, doi: 10.1109/SYNASC.2018.00041.

[33] “ROS.org | About ROS.” https://www.ros.org/about-ros/ (accessed Oct. 27, 2020).

[34] M. Amy *et al.*, “Towards Adaptive Fault Tolerance on ROS for Advanced Driver Assistance Systems To cite this version : HAL Id : hal-01707514 Towards Adaptive Fault Tolerance on ROS for Advanced Driver Assistance Systems,” 2018.

[35] J. Lussereau *et al.*, “In-tegration of ADAS algorithm in a Vehicle Prototype,” *IEEE Int. Work. Adv. Robot. its Soc. Impacts ARSO*, 2015.

[36] “SSD-Mobilenet-v2.” https://nvidia.box.com/shared/static/jcdewxep8vamzm71zajcovza938lygre.gz.

[37] “Open Images Dataset V6.” https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=detection&c=%2Fm%2F0k4j (accessed Nov. 20, 2020).

[38] “-.” https://raw.githubusercontent.com/dusty-nv/jetson-inference/dev/docs/images/pytorch-ssd-mobilenet.jpg.

[39] “GitHub - robopeak/rplidar\_ros.” https://github.com/robopeak/rplidar\_ros (accessed Nov. 20, 2020).