integration and testing using static and dynamic scenarios in the relevant environment.

II. Materials and Method

A. Materials

Advanced driver-assist systems (ADAS) are a system that provides information from the car surrounding environment to assist drivers in avoiding accidents by providing alerts [7]. When ADAS detects an object around the car that can cause an accident, ADAS will warn the driver by a sound, and a light indicator will light up or activate automatic braking. ADAS use various sensors to map and recognize objects around cars, such as LiDAR, Radar, Cameras, Ultrasonic Sensors, and the 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 an 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 a 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 each sensor's features 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. The methods include Viola Jones Detectors [16], Histogram of Oriented Gradients (HOG)[17], and Deformable Part-based Model (DPM) [18] in speed and accuracy [7]. Deep learning for object detection can be grouped into two approaches, i.e., “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 on a two-stage detection approach. Furthermore, CNN based one-stage detection there YOLO [26]–[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 detectors generate regions of interest using 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 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 an 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 [13], [35]. Another advantage of using ROS is that many active communities have already created modules and drivers[35].

B. Method

This paper focuses on discussing the real-time collision warning system using 2D LiDAR and Camera sensors for environment perception and estimate the distance (depth) and angle of obstacles for the Advanced Driver Assistant System (ADAS). In this work, a sensor fusion approach between the camera and 2D LiDAR sensor is proposed to estimate the distance and angle of obstacles in front of the vehicle implemented on the Nvidia Jetson Nano using the Robot Operating System (ROS). Figure 1 shows an overview of the ADAS architecture, and figure 2 shows an overview of the ROS architecture for conducting sensor fusion.

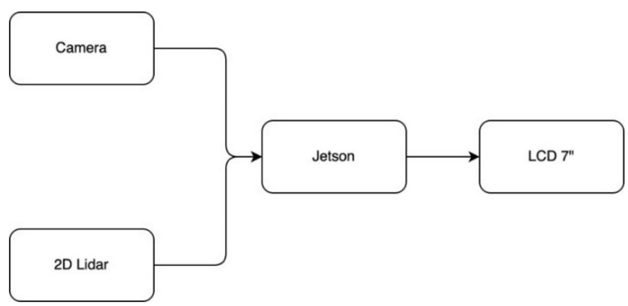


Fig **1**. Architecture System ADAS

In this context, the ADAS was equipped with a LOGITECH Webcam C170 camera for video streaming and SLIMTEC RP-LiDAR A1 model 2D LiDAR sensor to estimate the distance and angle of obstacles. A single board computer NVIDIA Jetson Nano with a Quad-core ARM A57 @ 1.43 GHz CPU, 4GB Memory, and 128-core GPU Maxwell has been implemented as the main processor of ADAS systems equipped with 7" HDMI IPS LCD, which is installed in the car dashboard. For software, ROS Melodic was installed on the NVDIA Jetson Nano as the main platform to implement the collision warning system for ADAS.

Furthermore, ROS architecture is created by applying the subscriber-publisher model system, whose architecture is depicted as in figure 2 to fuse the data from both sensors. In this architecture, there are two nodes that works as a publisher or a subscriber, i.e., the “Object Detection Node” and “Range /distance detection node.” These two nodes are connected to the ROS Master that works as a broker

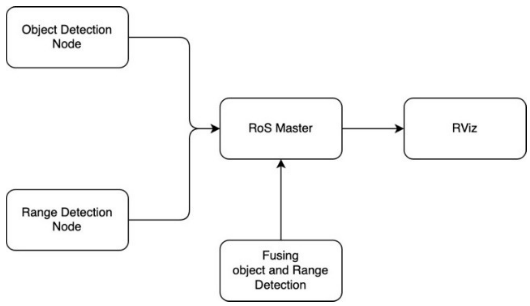


Fig. 2 ROS Architecture for ADAS

The main part of the collision warning system in this work is the object detection system and a system to measure the distance and angle of obstacles

A. Object Detection System

In this work, an essential part of the collision warning system is the object detection system. This part serves to identify the obstacle in front of the vehicle, such as cars, motorcycles, pedestrians, and other objects. For this reason, the camera sensor plays a vital role as a supplier of the environmental images for the object recognition process. "Object Detection Node" is a set of ROS packages that contains tools for object detection system using SSD MobileNet-V2 from [36]. The SSD MobileNet-V2 architecture uses a convolution predictor SSD300 with an input size of 680x480 pixels from a LOGITECH webcam with a field of view (FoV) of 60^0. A dataset from [37] has been applied to create this model, which supports the NVIDIA Jetson Nano. Figure 3 shows the SSD300 architecture used for the object detection system.

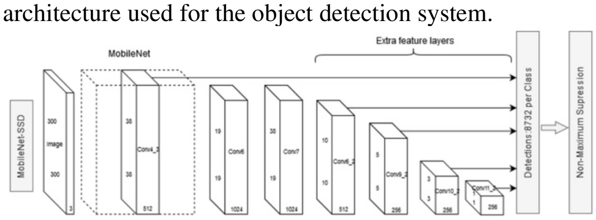


Fig. 3 Mobilnet-V2 SSD300 Architecture[38]

The output of the "*Object Detection Node*" is subject to be published in ROS Master using the topic "/jetson/result_detection" which contains image data (width and height) and bounding box coordinates of the detected object.

The following is the object detection algorithm:

Algorithm 1: Object Detection

Object detection estimation with Mobilenet
Input: Camera FoV 60º with 640x480
Output: Display object detection, width, height and centre coordinate detection
Method:

1. Initial node ROS with ObjectDetection
2. Initial object detection parameter image, width and height
3. Detection class 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

B. Measurement of Distance and Angle of Obstacles

Measuring the distance and angle of obstacles is carried out by fusing the camera sensor and RP-LiDAR 2D sensor. The fusion process is executed in "*Fusing Object and Range Detection Node*" with input from "*Object Detection Node*", "*Range Detection Node*", and "*ROS Master*". "*Range Detection Node*" will read the distance and angle data of obstacles around the vehicle from the RP-LiDAR sensor with a reading radius of 360 degrees using rplidar library. Furthermore, the data is sent to *ROS Master* using topic "/ scan_msg" and then the data will be fused on "*Fusing Object and Range Detection Node*".

In order to obtain the distance and angle obstacle, data fusion from RP-LiDAR and camera sensors was performed. The result of detection of distance and obstacle angle will be used to provide warning in ADAS. Beforehand, the calibration process has been carried out using a mechanism as shown in Figure 4 where the Camera sensor and the Lidar sensor are placed in the same position as in figure 6. In the calibration process, the focal length or field of view (fov) from camera and RP-LiDAR is set at an angle of 60 degrees. FoV will be used as a reference in the warning system at ADAS. This means that if the middle point of the bounding box from object detected by SSD-MobileNet within the FoV area, the warning system in ADAS will work (warning will be visualized on the 7" LCD using the RVIz package). The formula used for this purpose is as follows:

angle = (60.0/640.0)* detected.Center (1)
angle_detection = 60.0 + angle (2)
Distance = scan[int(angle_detection)] (3)

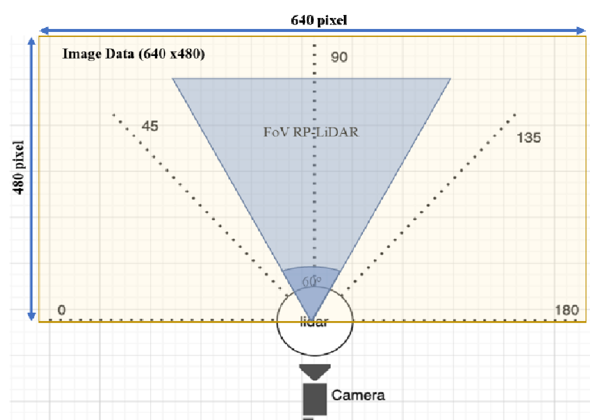


Fig. 4 RP-LiDAR and Camera Calibration Mechanism

The following is the distance detection algorithm:

Algorithm 2: Distance Detection

Distance Detection with 2D Lidar with RPLidar
Input: RPLiDAR
Output: distance detection
Method:

1. Initial node RoS with RangeDetection
2. Detection Angle
3. Convert Angle to Radian
4. Publish Range detection with parameter range and degree

The following is the fusion object and angle detection algorithm:

Algorithm 3: Fusion Object and Angle Detection

Range Detection with 2D Lidar with RPLiDAR
Input: node ObjectDetection and node RangeDetection
Output: Display object and angle detection
Method:

1. Subscribe Object detection and Range Detection
2. Initial range Radian to pixel
3. Detection object with midpoint coordinate
4. Convert center coordinate with range to save data range
5. Publish range detection and range

III. Results and Discussion

In this section, we describe the detailed experimental setup and performance evaluation process of the data fusion method within the context of distance and angle sensing for a warning system in ADAS. The experiment scenario used in this work is shown in Figure 5. The positional displacement between the sensors is explained in detail in Figure 6.

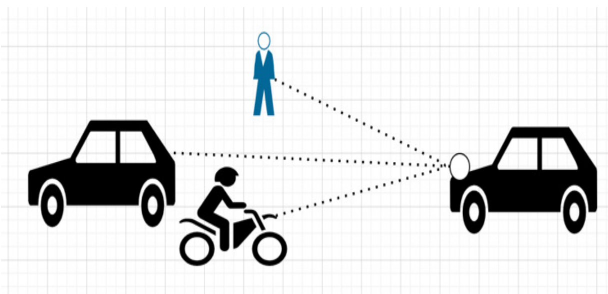
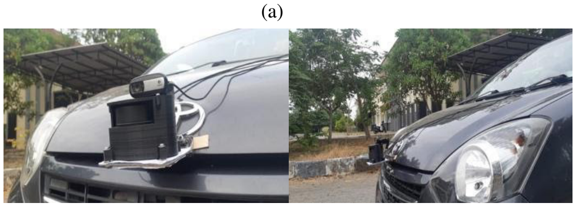


Fig 5. ADAS Experimental Scenarios

(a)



(b)

Fig. 6 (a) NVIDIA Jetson Nano to implement ROS and LCD display 7" (b) Installation of RP-LiDAR and Camera

ADAS prototype is mounted on the front of the car to get good detection results. In this work, we use 2D lidar, which can scan objects with a radius of 0 - 360. The RP-LiDAR and the camera are mounted parallel, as shown in figure 6 to facilitate the fusion process's calibration.

The experiment for ADAS performance testing was performed for three objects such as pedestrians, motorcycles and cars with a scenario, as shown in Figure 5. Based on the experimental results, the output data from the RP-LiDAR sensor is in the form of a 360 array with an infinity value if the object is above 12 meters, and the data will have a numeric value if an object is detected under that distance. Furthermore, the data will be sent by the "*range detection node*" to the *ROS master* with the message topic *"/scan_msg"*. The following is an example of data from RP-LiDAR detection results on t_n :

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)

Meanwhile, the data from the obstacle detection result from "*Object Detection Node*" contains information about the time, the midpoint coordinate of the object bounding box, and the object label. The following is an example of data from the object detection results:

t = 1602756362772608041 | (x,y) = (236.691894531,272.439361572) | distance = 0.641584333167 | lable = motorcycle

t = 1602757073928807020 | (x,y) = (431.908874512,215.612228394) | distance = 2.18477000313 | lable = truck

t = 1602756751280932903 | (x,y) = (478.734436035,191.203140259) | distance = 3.9323105664 | lable = bus

t = 1602757064671120882 | (x,y) = (377.174682617,215.417694092) | distance = 0.528737693536 | lable = car

t = 1602757063625045061 | (x,y) = (316.203216553,210.66519165) | distance = 1.91034458866 | lable = person

As a result, the object detection data from the "object detection node" would be fused with the "range detection node" data—the results of the fusion output, as shown in figure 7.

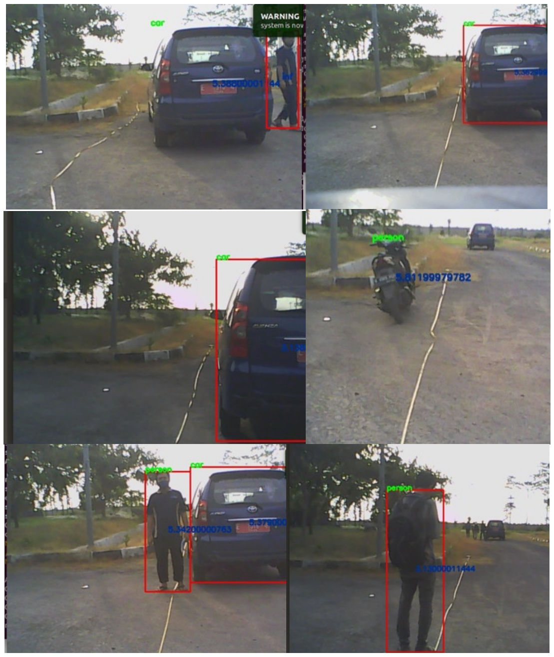


Fig. 7 The fusion results between object detection - distance and angle of objects

In Figure 7, it can be seen that the distance data (blue text) is the result of fusion. Table 1 describes the test results in more detail from two scenarios with an actual distance of 5.2 meters and 3 meters. In the table, it can be seen that the calculation of the distance (fusion) with actual conditions has an average error rate of 0.197 meters.

TABLE I.
TESTING THE ESTIMATED DISTANCE AND THE ACTUAL DISTANCE.

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

IV. CONCLUSION

The ADAS system has been developed using NVIDIA Jetson Nano with the ROS platform and input from two sensors. The first sensor is a 2D LiDAR sensor with a 360^o range with a maximum distance of 12 meters. The second sensor is a Logitech USB Camera sensor has shown a good performance where the object detection system can work with 40 fps performance, so it is suitable for real-time systems. Meanwhile, the fusion performance applied with the conversion from degree to coordinate has also shown satisfactory performance from the experiment to detect a 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. However, to improve its accuracy for the upcoming study, the researchers recommend using the computational geometric transformation and projection approach.

V. 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 (Research Collaboration between Higher Education) through LPPM Universitas Teknokrat 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," 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01707514.
[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.