2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Multi-Sensor Object Detection System for Real-Time Inferencing in ADAS

Sai Rithvick Mandumula
Mobility System
Kettering University
Flint, USA

Jungme Park
ECE
Kettering University
Flint, USA
jpark@kettering.edu

Ritwik Prasad Asolkar
Mobility System
Kettering University
Flint, USA

Karthik Somashekar
Mobility System
Kettering University
Flint, USA

Abstract— **Advanced Driver Assistance Systems (ADAS) are designed to assist drivers in various driving scenarios, and the object detection system is a critical component of ADAS. This paper aims to develop and evaluate an object detection system using two cameras placed on the vehicle's front and rear sides for real-time inferencing in ADAS. The real-world data set is collected under different weather and lighting conditions to evaluate the object detection system. The object detection system is further optimized using the TensorRT engine to deploy the system on the in-vehicle computing unit, NVIDIA Jetson AGX Xavier. The object detection system achieved 18 fps to process two cameras simultaneously on the in-vehicle computing unit, NVIDIA Jetson AGX Xavier. The experimental findings of this study will be useful for researchers, engineers, and manufacturers in the field of ADAS and autonomous vehicles to improve road safety and reduce accidents.**

Keywords—ADAS, object detection, TensorRT engine, deep neural networks, camera calibration, embedded devices

I. INTRODUCTION

Road vehicles are the primary mode of daily transport. The increasing use of transport increases accidents due to human inattention. With the rapid advancements in automotive technology, Advanced Driving Assistance Systems (ADAS) have become an integral part of modern vehicles. ADAS can reduce the number of car accidents and prevent deaths by improving road safety. ADAS date back to the early 20th century, with the first driver assistance systems including basic features such as windshield wipers and headlights. ADAS continued to evolve by introducing features such as Anti-lock Braking Systems (ABS), lane departure warning systems, and collision avoidance systems in the 1990s and 2000s. Today, ADAS technology advances, introducing self-driving capabilities and a growing emphasis on improving road safety.

One crucial component of ADAS is environment perception, which involves collecting and interpreting data about the environment surrounding the vehicle. Cameras are essential sensors for environment perception, as they can capture visual information about the road, traffic, and obstacles. Recently, camera-based Deep Neural Networks (DNNs) have achieved state-of-the-art performances in computer vision [1-3]. In addition, onboard computing units such as the NVIDIA Jetson series [4] (Nano, NX, AGX Xavier, etc.) have been introduced. Based on the advancement in DNN modules and computing power, evaluating the state-of-the-art object detection systems' performance with multiple cameras on the in-vehicle computing unit is necessary. In this paper, two camera sensors are placed on the testing vehicle's front and rear sides, developed the object detection system that can process images from the two cameras simultaneously, and deployed it on the in-vehicle computing unit for real-time inferencing.

The paper is organized in the following chapters. Chapter 2 reviews the state-of-the-art technologies in ADAS and object detection technologies. Chapter 3 explains the importance of the camera sensor, its limitations, the necessity for calibration, and the procedure to perform the calibration. Chapter 4 discusses the approaches to developing and evaluating the object detection system and how to deploy it on the in-vehicle computing unit. Chapter 5 concludes the main findings, highlights the study's contributions, and discusses the future research scope.

II. LITERATURE REVIEW

Recently, DNNs have been used extensively for object detection in ADAS. Research studies have shown that the state-of-the-art DNN algorithms can significantly improve object detection accuracy in various driving conditions, including low light and adverse weather. Zou et al. [1] explained how object detection is one of computer vision's most fundamental and challenging problems. Over 400+ papers have been reviewed by the authors [1], spanning over a quarter-century (from the 1990s to 2019). Several topics are discussed, such as detectors in history, detection datasets, metrics, speed-up techniques, fundamentals, and current state-of-the-art detection systems. They reviewed key technologies in detection methods, including Viola-Jones detectors, Histogram of Oriented Gradients, discriminatively trained part-based models, faster-region-based convolution neural network, You Only Look Once (YOLO), and Single Shot Detection (SSD).

Wang et al. [2] developed the most advanced object detection system, YOLO v7, in 2022. This system

outperforms all currently available object detectors in speed and accuracy. The authors examined different YOLO versions and related object detectors and claimed that YOLO v7 performed better than any other YOLO versions and other Convolution Neural Networks (CNN)-based object detectors. The proposed YOLOv7 was trained from scratch on the Microsoft COCO dataset to classify 80 classes. They focused on module-level re-parametrization by selecting modules using gradient flow propagation paths. Since Model re-parameterization techniques merge multiple computational modules into one at the inference stage, they could reduce about 40% of DNN parameters, which reduced the inferencing time. It is said to have a range of 5 frames per second (FPS) to 160 FPS in inferencing time and an accuracy of 56.8% average precision (AP). They addressed the problems existing in conventional architectures and focused on typical generic methods, modifications, and tricks to improve performance further.

Sensors are a critical component of ADAS, and their calibration and placement are essential for ensuring accurate detection and response to driving conditions. Proper sensor calibration and placement are crucial for accurately detecting objects and potential hazards. Various studies [5-7] have focused on developing calibration algorithms and techniques for different sensors used in ADAS. Camera calibration is closely related to the sensor framework, which describes capturing and converting light into digital signals by the sensor [5]. Fetic et al. [6] explained the calibration procedure of a charge-coupled device (CCD) digital camera to extract precise three-dimensional information from images. The calibration procedure determined which light is associated with each pixel on the resulting image. Dey et al. [7] proposed a novel framework called **VESPA** (**V**ehicle **S**ensor **P**lacement and orientation for **A**utonomy) for optimizing heterogeneous sensor placement and orientation for autonomous vehicles. They studied the synthesis of a heterogeneous sensor configuration for accomplishing autonomous vehicle goals. The authors claimed the VESPA framework could achieve the best setup for heterogenous sensors installed across two current actual vehicles, the Chevrolet Blazer and Chevrolet Camaro.

ADAS applications [8-11] encompass many features, including lane departure warning, collision avoidance, parking assistance, and adaptive cruise control. Park et al. [9] carried out a rear cross-traffic detection system for ADAS applications using radars and a camera. The proposed methodology uses a region of interest and CNN to classify the static and dynamic objects at the vehicle's rear. A Blind Spot Detection (BSD) system [10-11] is an essential functionality in ADAS. BSD is a safety feature commonly found in modern vehicles that helps drivers identify potential hazards in areas outside their field of view. Blind spots are areas around a vehicle that are not visible to the driver through the mirrors or windows, and they can pose a significant danger when changing lanes or merging with traffic. Ciberlin et al. [12] explained how modern vehicles are equipped with different ADAS systems and the importance of object detection and tracking using front-view cameras. Two-object detection methods, the Viola-Jones algorithm, and YOLO v3, are evaluated in accuracy and performance. They evaluated nine object detection modules for the Viola-Jones algorithm, and four object detection modules for YOLOv3 on precision, recall, and frames per second (fps) as inference time.

State-of-the-art literature review on ADAS-related topics showed that these systems can help reduce driver fatigue, improve fuel efficiency, reduce accidents, and increase road safety. However, challenges still need to be addressed, such as the cost of the technology, the need for reliable and accurate sensors, and the development of more advanced algorithms for object detection and sensor fusion.

III. Camera Sensor Calibration

Cameras are one of the most prevalent types of ADAS sensors used in today's automobiles, and they come in various forms and sizes depending on their role in a system. Recently vehicles have been equipped with 360-degree cameras that display an overhead picture of the vehicle's immediate surroundings by utilizing many tiny cameras positioned at the front, rear, and sides. Camera sensor provides rich visual data, are relatively low-cost, and have a wide field of view. On the other hand, they are limited by range, can be affected by lighting conditions, have limited depth perception, and are less weather-resistant than other sensors.

Camera calibration is an important process to ensure the accuracy and reliability of the visual data captured by a camera in a vehicle. The image presented in Fig. 1. has a distortion, and the actual location and size of the object are different from the real one. In addition, the image bends as it moves toward the end. So, calibration is required to produce an undistorted actual image. Cameras have two main types of distortion: radial distortion and tangential distortion. Radial distortion occurs when light rays entering the camera lens do not pass through the lens' optical center. This causes straight lines in the real world to appear curved in the image. On the other hand, tangential distortion occurs when the camera's lens is not perfectly parallel to the image sensor, as shown in Fig. 2. This causes the image to appear tilted or skewed.

Fig. 1. Example of a distorted image.

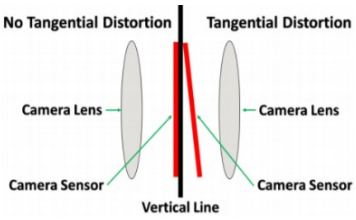

Fig. 2. Tangential distortion because the camera's lens is not parallel to the image sensor.

Both radial and tangential distortion can be corrected through camera calibration, which involves estimating the distortion parameters. Calibration techniques typically include capturing multiple images of a known calibration target and using the image data to estimate the distortion parameters. Camera calibration involves estimating the relationship between the 3D world and the 2D image captured by the camera. This relationship is defined using several coordinate systems and parameters.

The extrinsic parameters shown in Fig. 3 include the rotation matrix, R, and translation vector, t. The extrinsic matrix is denoted by [R, t], where R is the 3x3 rotation matrix, and t is the 3x1 translation vector. The rotation matrix, R, brings the corresponding axes of the two frames into alignment, as shown in Fig. 4(a). The translation vector, t, between the relative positions of the origins of the two reference frames is found, as mentioned in Fig. 4(b). The extrinsic matrix describes the position and orientation of the camera relative to the object being photographed. On the other hand, the intrinsic camera parameters define the internal characteristics of the camera, such as the focal length, principal point, and distortion coefficients as shown in Fig. 3. The intrinsic parameter matrix, K, is defined as below:

$$K = \begin{bmatrix} 1/\Delta x & 0 & u_0 \\ 0 & 1/\Delta y & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{f}{\Delta x} & 0 & u_0 & 0 \\ 0 & \frac{f}{\Delta y} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1)$$

Where f is the camera focal length, Δx and Δy are the physical x and y pixel lengths in the image coordinate, respectively. The notation,u_0 is the x pixel coordinate of the intersection point between axis Zc in camera frame coordinates and the image plane. Similarly, v_0 is the y pixel coordinate of the intersection point between axis Zc and the image plane. The intrinsic and extrinsic camera parameters are estimated from a set of calibration images during calibration. This allows us to convert between the different coordinate systems and correct for distortion and other errors in the images. These parameters are used to determine an accurate mapping between 3D world coordinates and their corresponding 2D image coordinates.

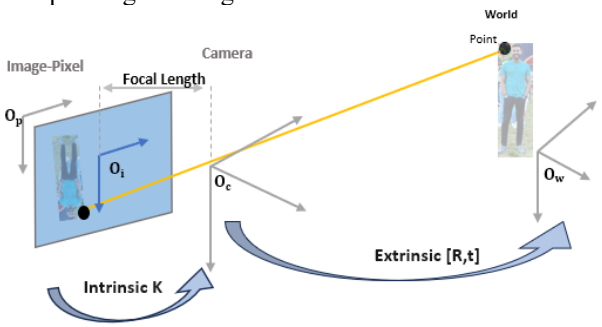

Fig. 3. Coordinate systems of a camera and image.

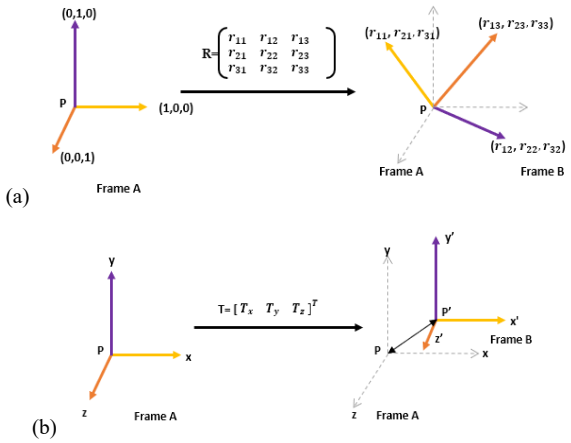

Fig. 4. Extrinsic parameters: (a) rotational matrix, R, b) translation vector, t.

The camera calibration procedure can be implemented using MATLAB's Computer Vision System Toolbox [13]. A checkered board with a known size is used for the camera calibration, as shown in Fig. 5 (a). It is required to acquire a set of calibration images as shown in Fig. 5 (b). The camera's intrinsic and extrinsic parameters are estimated by detecting and re-projecting all 63 points (7 rows x 9 columns) in the checker-board image. Fig. 6 shows the corrected image using the intrinsic and extrinsic parameters found from camera calibration, where Fig. 6 (a) is the distorted image and Fig. 6 (b) is the corrected image using the extrinsic and intrinsic camera parameters. It is important to note that the accuracy of the camera calibration depends on the quality and quantity of the calibration images, the calibration pattern used, and the calibration algorithm. Therefore, capturing multiple images of the calibration pattern from different viewpoints and distances is recommended to ensure the robustness and accuracy of the calibration.

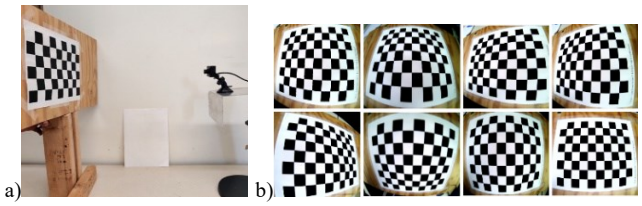

Fig. 5. Calibration procedure: a) A checkered board and the camera sensor, b) Captured images for calibration.

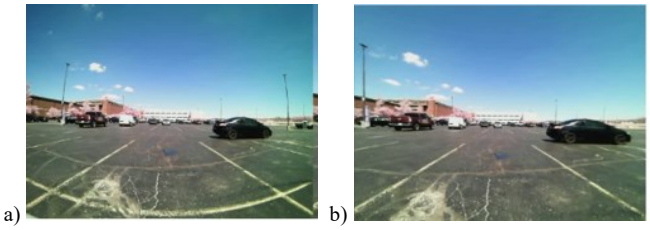

Fig. 6. Corrected images using intrinsic and extrinsic camera parameters: a) a distorted parking lot image, b) a corrected parking lot image.

IV. Developing the Object Detection System for Two Cameras

To develop a robust and reliable object detection system that can monitor the vehicle's front and rear sides simultaneously, two Spinel USB cameras [14] are selected and mounted in the testing vehicle as shown in Fig. 7. The Spinel USB camera is selected because it has high resolution (5Mpixels), compact size, and low price. The specifications of the camera are presented in TABLE I.

Deep learning-based object detection techniques use end-to-end learning, which transforms raw input images into hierarchical feature representations. Since YOLO was proposed by J. Redom et al. [15] in 2016, it has become one of the most popular deep-learning architectures for object detection. YOLO takes an image and predicts the object locations with bounding box coordinates and class probabilities. The main features of YOLO are high speed, high accuracy, and learning ability. The YOLO family of models has continued to evolve and the current version of YOLOv7 [2] is released in July, 2022. Two versions of YOLO architectures, YOLOv4 [16] and YOLOv7, are considered as object detection system in this research because these two versions were achieved great improvement in object detection. YOLOv4 uses a more advanced backbone network CSPDarknet53, with Spatial pyramid pooling for better feature extraction, the addition of a Path Aggregation Network (PANet) for handling different object scales, and the implementation of several optimization techniques to improve accuracy. On the other hand, YOLOv7 expanded upon YOLOv4 by incorporating a larger and more powerful backbone network to capture richer spatial and semantic information. YOLOv7 also leverages a combination of anchor-free and anchor-based approaches for better localization and detection accuracy. YOLOv7 has a much smaller model size than YOLOv4, which means it is faster to run the model for real-time inferencing and requires less memory.

Fig. 7. Hardware set-up: a) Camera Sensor, b) cameras mounted in front and rear side of the testing vehicle.

TABLE I. Specification of the Spinel Camera

Parameter	Description
Resolution	5 Megapixels
Compliance	USB
Streaming	MJPEG & YUV2
WDR (Wide Dynamic Range)	120 DB (decibels)
Minimum Illumination	0.2 Lux
Operating Temperature	~4° F ~167° F
Dimensions	38 mm x 38 mm

It is necessary to evaluate the two selected DNN models for ADAS with two cameras in front and rear of the testing vehicle. To compare the performances of the two selected DNN models in terms of accuracy and processing time, a total of 14,015 images are collected with various light conditions. Lighting conditions can significantly impact objects' visibility and appearance, affecting the system's accuracy. The collected image samples are presented in Fig. 8, including image samples in different weather conditions and traffic densities. Ground truth annotation for two object classes "car" or "pedestrian" has been done using the MATLAB Image Labeler app [17] as shown in Fig. 9. Total 128,341 objects are labeled as 'car or 'pedestrian' using the labeling app.

The detection model is evaluated by comparing the prediction outputs with the ground truth annotations. If the predicted bounding box from the DNN model overlaps with the ground truth bounding box above 50% or more, it is considered a correct detection, and it is considered as True Positive (TP). If the predicted bounding box does not overlap with any ground truth bounding box, it is considered a False Positive (FP) detection. If ground truth bounding boxes are not detected, False Negative (FN) is incremented. Three different metrics are used to measure the accuracy of the object detection model: Precision, Recall, and F1-score. Precision measures how many positive predictions are correct, and Recall measures how many positive cases the classifier correctly predicted over all the positive cases in the testing data. F1-Score is a measure combining both Precision and Recall. F1-Score is generally described as the harmonic mean of the two. The mathematical representations of the evaluation metrics are defined in (2)-(4):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, \tag{3}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}}. \tag{4}$$

Fig. 8. Sample collected images with various lighting and traffic conditions.

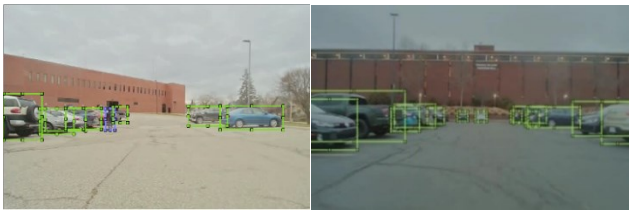

Fig. 9. Example of labeled images for ground truth.

A total of 14,015 collected images with the size 640 by 480 are used to evaluate the two DNN models. TABLE II summarizes the performances of two models on the collected images. The precision of YOLOv7 is 0.89, which is 4.5% higher than the YOLOv4 precision, 0.85. The recall of YOLOv7 is 0.92, which is 2% higher than the recall of YOLOv4, 0.90. Similarly, the F1 score of YOLOv7 is 0.90, which is 2% higher than the F1 score of YOLOv4, which is 0.88. The three metrics, Precision, Recall, and F1-score, are plotted in Fig. 10. In Fig. 10, the recall is slightly higher than the precision for both DNN models. Based on the performance evaluation in Fig. 10 and TABLE II, YOLOv7 has shown better accuracy and sensitivity in object detection for both car and pedestrian classes. Fig. 11 presents example detection results by YOLOv4 and YOLOv7 models. The detection results in Fig. 12 show that YOLOv7 can detect tiny objects even in low illumination conditions.

To deploy the model on the in-vehicle computing unit for real-time inferencing, the NVIDIA® Jetson AGX Xavier™ Developer Kit [4] is selected. The NVIDIA® Jetson AGX has a powerful computing power of up to 32 TOPs (Tera operations per second) with a 512-core Volta GPU, as shown in Fig. 13 (a). For real-time inferencing, object detection processing time is critical. The selected two DNN models were deployed on the Jetson AGX and measured the processing time. Fig. 14. displays the inferencing time for two DNN models. For one camera processing, the inferencing times for YOLOv4 and YOLOv7 are 8 fps and 15 fps, respectively. For two cameras' simultaneous processing, the processing time becomes slower than one camera processing, and the inferencing times for YOLOv4 and YOLOv7 are 5 fps and 9 fps, respectively. It is necessary to improve the inferencing time further.

TABLE II. DETECTION RESULTS OF YOLOV4 AND YOLOV7

Models	TP	FP	FN	Precision	Recall	F1-score
YOLOv4	116,653	28,859	11,688	0.85	0.90	0.88
YOLOv7	120,707	23,077	7,634	0.89	0.92	0.90

Fig. 11. Evaluation of the DNN models using three metrics.

Fig. 12. Example Detection results of YOLOv4 and YOLOv7.

Fig. 13. Deployment of the object detection system on the in-vehicle computing unit: a) the NVIDIA® Jetson AGX Xavier™, b) In-vehicle computing unit in the testing vehicle.

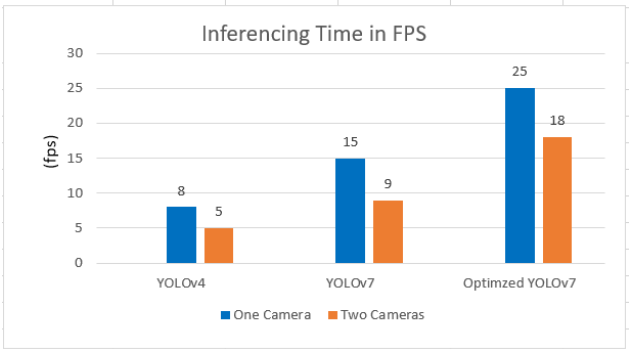

Fig. 14. Average inferencing time for the DNN models.

Since the low response time and high accuracy in ADAS are critical factors when the object detection system is deployed, the selected YOLOv7 model is optimized further to reduce the inferencing time using the TensorRT engine [18]. First, the YOLOv7 model is converted into the ONNX [19] model, then the ONNX model is converted into the TensorRT engine. The detailed optimization procedures can be found in [20]. Finally, the optimized YOLOv7 model is deployed on the in-vehicle computing unit (Jetson AGX Xavier) for real-time inferencing. The measured inferencing time for one camera is 25 fps and for two cameras is 18 fps, as presented in Fig. 14. Fig. 15 presents the demo of simultaneous real-time inferencing using two cameras mounted on the testing vehicle's front and rear sides. The demo in Fig. 15 shows that the deployed object detection system can process two camera images simultaneously for real-time inferencing.

V. CONCLUSION

This paper is dedicated to developing and evaluating an object detection system for ADAS using two camera sensors placed on the testing vehicle's front and rear sides. Two DNN models are evaluated using a collected real-world dataset in various scenarios, including daytime and nighttime driving, weather conditions, and traffic densities. The selected model is further optimized using the TensorRT engine. The optimized object detection system has achieved running 25 fps for the single-camera processing and 18 fps for two cameras. The findings of this study will be helpful for researchers, engineers, and manufacturers in the field of ADAS and autonomous vehicles. The proposed object detection system using camera sensors helps in proximity detection around the car, improving road safety and reducing accidents.

For future research, research on optimization of the DNN models and the inferencing time with more cameras for a 360-degree surrounding monitoring system. In addition, research on the sensor fusion system is vital to improve the performance of the obstacle detection system by fusing different sensor information.

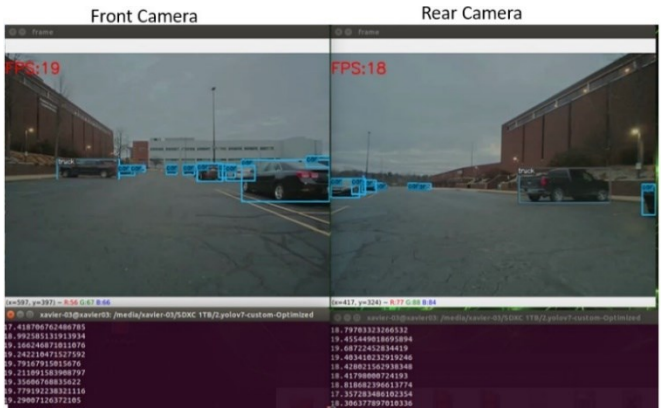

Fig. 15. Real-time inferencing using the optimized YOLOv7 model on the NVIDIA Jetson AGX Xavier.

ACKNOWLEDGMENT

Thanks to the scholarship support for this research by the Robert Bosch Centennial Professorship at Kettering University.

REFERENCES

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

[2] C. -Y. Wang, A. Bochkovskiy, H. -Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors" (2022).

[3] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object detection with deep learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[4] NVIDIA Jetson Xavier. Available online: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/

[5] P. Gabriel, B. Octavian, I. Daniela and I. Daniel, "In situ geometric calibration of a hybrid system composed by Uav/Gnss/Imu/Aerial Camera and Lidar." Scientific Papers-Series E-Land Reclamation Earth Observation and 9: 241-249. 2020, 2285-6064.

[6] A. Fetić, D. Jurić, and D. Osmanković, "The procedure of a camera calibration using camera calibration toolbox for MATLAB," 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1752-1757.

[7] J. Dey, W. Taylor, and S. Pasricha, "VESPA: A framework for optimizing heterogeneous sensor placement and orientation for autonomous vehicles," in IEEE Consumer Electronics Magazine, vol. 10, no. 2, pp. 16-26, 1 March 2021, doi: 10.1109/MCE.2020.3002489.

[8] Z. Zhong, S. Liu, M. Matthew, and A. Dubey, "Camera Radar Fusion for Increased Reliability in ADAS Applications," in *Proc. IS&T Int'l. Symp. on Electronic Imaging: Autonomous Vehicles and Machines*, 2018, pp 258-1 - 258-4. https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-258

[9] J. Park, W. Yu, "A sensor fused rear cross-traffic detection system using transfer learning." sensors (Basel). 2021 Sep 9;21(18):6055. doi: 10.3390/s21186055. PMID: 34577263; PMCID: PMC8470253.

[10] National Transportation Safety Board (NTSB), "The use of forward collision avoidance systems to prevent and mitigate rear-end crashes," 2015.

[11] I. Baftiu, A. Pajaziti and K. C. Cheok, "Multi-mode surround view for ADAS vehicles," 2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Tokyo, Japan, 2016, pp. 190-193, doi: 10.1109/IRIS.2016.8066089.

[12] J. Ciberlin, R. Grbic, N. Teslić, and M. Pilipović, "Object detection and object tracking in front of the vehicle using a front view camera," 2019 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2019, pp. 27-32, doi: 10.1109/ZINC.2019.8769367.

[13] Matlab app for camera calibration:
https://www.mathworks.com/help/vision/camera-calibration.html

[14] Spinel Camera: https://www.spinelelectronics.com/UC50MPD

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE CVPR, Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[16] A. M. Roy, R. Bose, and J. Bhaduri, "A fast accurate fine-grain object detection model based on YOLOv4 deep neural network," *Neural Comput & Applic* **34**, 3895–3921 (2022).

[17] Image Labler: https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html

[18] NVIDIA Deep Learning TensorRT Documentation. Available online: https://docs.nvidia.com/deeplearning/tensorrt/developer-guide/index.html.

[19] ONNX. Available online: https://onnx.ai/get-started.html.

[20] J. Park, P. Aryal, S. Mandumula, and R. Asolkar, "An Optimized DNN Model for Real-Time Inferencing on an Embedded Device," *Sensors* 23, no. 8: 3992, 2023.