



Article Smart Driver Behavior Recognition and 360-Degree Surround-View Camera for Electric Buses

Mehmet Uğraş Cuma ^{1,*}, Çağrı Dükünlü ¹, and Emrah Yirik ²

- Electrical & Electronics Engineering, Çukurova University, Adana 01130, Türkiye; cagridknl@gmail.com
 Ottomotiva Mühandiclik va Tasarım A.S. Bilisim Vadisi Muallim Mah. Daniz Cad. No:143/6
- ² Ottomotive Mühendislik ve Tasarım A.Ş., Bilişim Vadisi, Muallim Mah. Deniz Cad. No:143/6,
- Kocaeli 41400, Türkiye; emrah.yirik@ottomotive.tech
- * Correspondence: mcuma@cu.edu.tr

Abstract: The automotive industry's focus on driver-oriented issues underscores the critical importance of driver safety. This paper presents the development of advanced driver assistance system (ADAS) algorithms specifically tailored for an electric bus (e-bus) to enhance safety. The proposed approach incorporates two key components: a 360-degree surround-view system and driver behavior recognition utilizing the You Only Look Once V5 (YOLO_V5) method. The adoption of YOLO_V5 in ADASs enables rapid response by processing multiple class probabilities and region proposals within an image instantaneously. Additionally, ADAS implementation includes an image processing-based surround-view system utilizing OpenCV. In order to evaluate the performance of the proposed algorithms regarding a smart e-bus, comprehensive experimental studies were conducted. The driver behavior recognition system underwent rigorous testing using various images captured by an onboard camera. Similarly, the surround-view system's performance was verified in diverse driving scenarios, including regular driving, parking, and parking in near-to-line situations. The results demonstrate the viability and effectiveness of the proposed system, validating its potential to significantly improve driver safety in electric buses. This paper provides a comprehensive overview of the work accomplished by emphasizing the specific contributions of the 360-degree surround-view system, driver behavior recognition using YOLO_V5, and the experimental validation conducted for an e-bus.

Keywords: advanced driver assistance system; driver behavior recognition; surround view; you only look once

1. Introduction

Nowadays, one of the emerging concepts in automotive technologies is smart vehicle systems that are equipped with a variety of sensors to perceive their surroundings and driver behavior recognition systems. The main motivation is to reduce the rate of accidents caused by driver inattention/misbehavior and blind spots by equipping vehicles with smart features. Thus, the recognition of driver activity and the detection of blind spots have become key issues for safer vehicles. As a matter of fact, the smart vehicle industry is expected to improve gradually in the following decades.

The outside mirrors used in vehicles to change lanes have visibility restrictions. Blind spot detection studies have been carried out to prevent these limitations from causing fatal traffic accidents as a result of reckless lane departures. The related system is one of the critical functionalities of conventionally named advanced driver assistance systems (ADAS). Several studies have been carried out regarding ADASs, and the authors stated that the automotive industry market intention is increasingly moving toward it as it increases the driver's control of the vehicles by providing advanced visualization in real time. Studies within the scope of ADASs, which involve lane departure monitoring [1], adaptive cruise control [2,3], speed limit monitoring [4], forward collision warning [5], emergency brake assistance [6], and surround-view monitoring [7], have been accelerating in the literature.



Citation: Cuma, M.U.; Dükünlü, Ç.; Yirik, E. Smart Driver Behavior Recognition and 360-Degree Surround-View Camera for Electric Buses. *Electronics* 2023, *12*, 2979. https://doi.org/10.3390/ electronics12132979

Academic Editor: Felipe Jiménez

Received: 2 May 2023 Revised: 15 June 2023 Accepted: 16 June 2023 Published: 6 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Studies highlight that the lane departure warning and forward collision warning systems provide 9.3–33.3% and 23–50% crash prevention rates, respectively [8].

One of the real-time systems that has been adapted to the ADAS is the bird's-eye surround view (SV), which integrates several cameras to produce a virtual bird's-eye view in real-time [9,10]. SV represents a sensor fusion problem, in which many sensors, in this case, cameras, are merged and processed together to create a complete image of the location surrounding a vehicle and to assist a machine or human driver in making quick choices. Ananthanarayanan [11] states that processing many sensors in real time can be a difficult computing task, especially for high-bandwidth applications incorporating video or radar inputs. The other problem is defined as image stitching in [12]. Several studies, including (i) pairing the corresponding detected key points and the (ii) interpolation of image elements captured by vehicle mirror cameras and stereo pairs of cameras at the rear of the vehicle, were performed in order to solve the stitching problem in [13–16].

The other way to enhance road safety for a vehicle driver is to adopt driver behavior recognition systems. Intelligent (i.e., highly automated) vehicles are classified into categories by considering their range of autonomy [17]. Some vehicles have no driving automation, while some vehicles allow drivers to take over vehicle control under emergencies to take control of a vehicle considering the driver's behavior/activity. Several studies have been performed regarding real-time driver behavior/activity in the literature. While previous studies have focused on drivers' attention/distraction [18], driver intention [19], driving styles [20], and driver fatigue detection [21,22], the studies based on artificial intelligence have been accelerated.

A convolution-based approach for driver activity recognition tasks using body posture data is proposed in [23]. Additionally, the items and position information, which are used as inputs for activity recognition graph convolution-based network models, have been extracted. In order to categorize images, three-dimensional ResNets were used in [24]. Some researchers have used deep convolution neural networks in order to define common driving activities [25]. The raw images have been segmented by a Gaussian mixture model to extract the driver's body as training data. Segmentation is frequently performed according to the head pose angle, gaze direction, and hand/body joints [25,26]. A virtual reality driving simulator environment, which is employed with force sensors in the driver's seat to collect sensor inputs from many participants, was utilized in [27]. In order to categorize the test signals, a basic difference-based comparison technique with training data was employed. An activation map, including the entire image, one body position, and body-object interactions, based on a deep fusion network that merges layers, was developed to recognize driver activity in [28]. Zhao, et al. [29] developed a cascading multiple attention-based deep learning model, where the entire network is composed of subnetworks. At first, one of the subnetworks takes the original image and creates an activation map. The most active area on the activation map is then cut out of the input image and used as input in another subnetwork. A K-nearest neighbor classifier is used to combine and classify the outputs of each subnetwork.

To summarize, these techniques generally aim to improve accuracy by using convolution-based approaches. In addition to high accuracy, the frame-per-second factor has great importance. YOLO (You Only Look Once) is an algorithm for object detection using a convolution neural network. YOLO excels at fast object detection capabilities (high frame-per-second factor) in comparison with the aforementioned methods. The performance benchmarking of the methods is summarized in Table 1 in terms of mean average precision (mAP) and frames per second (fps).

This study contains two different solutions for ADASs: (i) a fast driver behavior recognition machine learning model, which was trained using a new, qualified dataset, and (ii) an original manual surround-view method. The main superior aspect of the proposed system is the fast recognition of the driver's behavior in comparison with existing methods owing to the fast prediction capability of YOLO_V5. The proposed surround-view system does not require calibration for different vehicles. Additionally, it excels at fixed

pixel positions that make for fast and cheap surround-view capture for the system. The improved ADAS, which was equipped with methods that include the aforementioned advantages, was experimentally tested using an e-bus.

Table 1. Benchmarking of YOLO with other object detection algorithms.

Detection Frameworks	Train	mAP	fps
Fastest DPM [30]	VOC 2007	30.4	15
R-CNN Minus R [31]	VOC 2007	53.5	6
Fast R-CNN [32]	VOC 2007 + 2012	70.0	0.5
Faster R-CNN VGG-16 [33]	VOC 2007 + 2012	73.2	7
Faster R-CNN ZF [33]	VOC 2007 + 2012	62.1	18
YOLO VGG-16 [34]	VOC 2007 + 2012	66.4	21
YOLO [35]	VOC 2007 + 2012	63.4	45

The arrangement of the current work is described in the following manner: the proposed methods and experimental setup are described in Section 2. The details of the developed methods are presented in the subtitles of Section 2. The performance results are presented in Section 3. Finally, Section 4 analyses the results and the contributions of this paper.

2. Proposed Method and Experimental Setup

The main purpose of the study was to increase the driver's awareness of the environment, develop a warning system to prevent fatal accidents and integrate a composite feature for the ADAS. The equipment utilized to achieve the objectives consisted of five automotive-grade cameras equipped with wide-angle fish-eye lenses, a 10.4-inch widescreen LCD monitor, and a high-end computer optimized for image processing. These components were integrated into an electric bus platform to evaluate the effectiveness and user-friendliness of the proposed systems. As depicted in Figure 1, four fish-eye lens cameras were strategically positioned at the most convenient locations on the bus to cover all blind spots. Two cameras were situated on the sides, while the other two were positioned at the front and rear of the bus. The angles of the cameras were determined as being 185°—horizontal and 142°—vertical in order to provide proper placement and ease of wiring inside the vehicle. The fifth camera, as seen in Figure 1, was installed inside the interior of the bus. The detailed camera distances and degrees are summarized in Table 2.



Figure 1. Camera locations on the bus model (vector illustration).

Camera Horizontal Distances					
Cabin Width		2512 mm			
Cabin Height		8953 mm			
Distance Of Rear Camera To Front Camera		8953 mm			
Distance Of Rear Camera To Side Camera		x: 1189 mm y: 2719 mm			
Distance Of Side Camera To Side Camera		2512 mm			
Distance Of Side Camera To Front Camera		x: 1269 mm y: 6233 mm			
Camera Vertical Distances					
Front Camera Ground Distance	:	3290 mm			
Side Camera Ground Distance		1428 mm			
Rear Camera Ground Distance		855 mm			
Outer Camera Degrees					
Camera Lateral Angle	:	185°			
Camera Vertical Angle	:	142°			
Inner Camera Degrees					
Camera Lateral Angle		180°			
Camera Vertical Angle	:	135°			

Table 2. Camera distances and degrees of vehicle.

2.1. 360-Degree Surround-View System

In order to provide an effective surround view and detect blind spots, several steps were followed for (i) image acquisition, (ii) undistortion, (iii) calibration, (iv) warping, (v) stitching, and (vi) output image processing, respectively. The steps initiate with the placement of the cameras to cover a 360-degree view from the bus, as illustrated in Figure 2. The second step is to transmit the images captured from cameras to the computer via the serial port. Following image acquisition, several image processing steps were performed to define depth in the image, calibrate the image, mask the selected areas, warp as order, stitch the masked images, and perform manipulations through pixels to produce a surround view. When the two cameras, which have a common place for vision, are different in the vertical plane but aligned in the horizontal plane (like the human eye), the point of a visible object falls on the same horizontal axis in both cameras. Thus, information about depth can be extracted from the proximity and distance of the objects [36]. Searching where the same point coincides in two images has a huge computational load, so searching only on the horizontal axis, which is called stereo calibration, by matching the horizontal axes reduces the processing load. For this, the OpenCV and grab&retreive methods were used, and the calibration function, flag, and relationship definitions, as well as noise canceling, are explained by a pseudo-code in Table 3.



Figure 2. Schematic of vehicle, including distances.

5 of 14

 Table 3. OpenCV stereo calibration function pseudo-code.

ret, K1, D1, K2, D2, R, T, E, F = cv2.stereoCalibrate(objp, leftp, rightp, K1, D1, K2, D2, image_size, criteria, flag

CV_CALIB_FIX_INTRINSIC: CV_CALIB_USE_INTRINSIC_GUESS: CV_CALIB_FIX_PRINCIPAL_POINT: CV_CALIB_FIX_FOCAL_LENGTH: CV_CALIB_FIX_ASPECT_RATIO: CV_CALIB_SAME_FOCAL_LENGTH: CV_CALIB_ZERO_TANGENT_DIST: CV_CALIB_FIX_K1, ..., CV_CALIB_FIX_K6:

*R*1, *R*2, *P*1, *P*2, *Q*, *roi_left*, *roi_right* = *cv*2.stereoRectify(K1, D1, K2, D2, *image_size*, *R*, *T*, *flags=cv*2.CALIB_ZERO_DISPARITY, *alpha=*0.9)

LeftMapX, leftMapY = cv2.initUndistortRectifyMap(K1, D1, R1, P1, (width, height), cv2.CV_32FC1) left_rectified = cv2.remap(leftFrame, leftMapX, leftMapY, cv2.INTER_LINEAR, cv2.BORDER_CONSTANT) rightMapX, rightMapY = cv2.initUndistortRectifyMap(K2, D2, R2, P2, (width, height), cv2.CV_32FC1) right_rectified = cv2.remap(rightFrame, rightMapX, rightMapY, cv2.INTER_LINEAR, cv2.BORDER_CONSTANT)

The next step involves the wrapping and masking of images. Once image calibration is performed, the frames captured by the fish-eye lenses are stabilized, resulting in calibrated image frames. The calibrated image's perspective views are then selected and undergo warping, as the surround view necessitates the utilization of four distinct images and stitching. Warping is achieved by specifying four points on the frame to map to the corresponding destination points. The pseudo-codes of warping and masking are presented in Table 4.

Table 4. Pseudo-code of warping and masking.

$eq:point_start_$
%masking
cv::Mat maskR = cv::Mat::zeros(cv::Size(width, height), CV_8U);
vector <point> pts = {Point(x1,y1),Point(x2,y2),Point(x3,y3),Point(x4,y4)};</point>
fillPoly(maskR,pts,Scalar(255));
cvtColor(maskR, maskR, COLOR_GRAY2BGR);
cv::Mat resR;
bitwise_and(frame,maskR,resR);

getPerspectiveTransform calculates a perspective transform from four pairs of the corresponding points. The function calculates the 3×3 matrix of a perspective transform so that

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = map_{matrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$
(1)

$$dst(i) = (x'_i y'_i), src(i) = (x_i, y_i), i = 0, 1, 2, 3$$
⁽²⁾

The wrapping process continues to stitch images by considering the differences between the pixels. The camera frames are taken from the operating system, which is Linux/Ubuntu. A capture card is required for taking the camera frames. The serial channel gathers the frames from the video channels and provides channels that are stored in channel buffers. Using cat commands provides the desired channel data transmission in ASCII code placement. Having different ASCII characters flowing through the channels means video transmission data are arriving in the serial channels; thus, it could be accessed via OpenCV libraries. OpenCV has a VideoCapture class that gives features to use frames for open channels and retrieve frames and show them with imshow commands. These frames have default settings for PAL- or NTSC-type video capture. Both are terrestrial broadcast signals. Image quality is the same for both. "VideoCapture::read(frame*)" accesses these PAL frames using colored images from a 0-to-255 matrix using a three-dimensional color matrix.

After all the necessary preparations, the videos are recorded for the image stitching test. Since different images can have varying frame types, such as PAL, NTSC, and other formats, stitching the images requires them to be captured separately. Even different frames can have different dimensionalities, like 2D, 3D, and 4D, and color channels may differ from full RGB and limited RGB, like 0–255 or 16–235. That is why image stitching needs to be taken care of separately. If any frame is irrelevant to the other frames regarding type or range, the exception is thrown out.

Stitching images requires warping and masking. Four different frames are warped from upside to down, and important points are masked for stitching. The masks are implied to have colored frames with the bitwise and operator. The frames are rotated due to stitching places: the front stays the same, the left frame rotates 90 degrees counterclockwise, the right image rotates 90 degrees clockwise, and the back image rotates 180 degrees. All frames are resized for stitching purposes to have a size of 640,480 for the final image. Lastly, all frames are summed up. If all frame types are the same and the dimensions do not differ, this can be summed with the "+" operator.

final_frame = front_frame + left_frame + right_frame + back_frame (3)

Stitched images may not show the best surround view when first stitched. So, a function is written for pixel manipulation that changes the source points from srcPoints and dstPoints by giving arguments to increase some x and y dimensions for the written function.

2.2. 360-Degree Surround-View System Differences from Related Studies

The development of 360-degree surround-view systems for car safety is an ongoing effort, with different models utilizing varying numbers of cameras and technologies to create surround views of varying quality. While there is no industry standard for this technology, some models outperform others in terms of quality and frame rates, with specific equipment and technologies being used depending on the needs of the application. The primary goal of these systems is to improve driver safety by providing vision for blind spots.

One approach proposed in a recent article [37] uses a super-resolution convolutional neural network (SRCNN) to enhance image quality. While the use of SRCNN resulted in better image quality when compared to traditional systems, it does come with a significant disadvantage in that it is computationally intensive and requires high processing power and memory to operate effectively. Thus, using embedded systems for such models is not recommended.

Another article [38] proposed a method using a homography matrix, MSL deform, polynomial deformation, and homography merging to warp an image. While this method can automatically calibrate the system, it can be slow and heavy to process, and any changes to the calibration file require recalibration.

In this article, we propose a method that uses fixed calibration points instead of fully automatic calibration, making it more efficient and cost-effective. The proposed approach requires less computation and can be integrated into less powerful systems, making it more suitable for embedded systems in vehicles.

Overall, the goal of a 360-degree surround-view system is to improve driver safety by providing visibility to blind spots. The proposed method offers a more efficient and cost-effective approach to achieve this goal, making it a promising option for integration into future car safety systems.

2.3. Driver Behavior Recognition Based on YOLO

Figure 1 also shows the placement of the internal camera that checks the behavior of the driver. The driver behavior detection system used the YOLO method. YOLO is much faster than other algorithms because it takes the region proposal inside of an image with multiple class probabilities and processes it instantaneously. The YOLO algorithm performs several steps [27], as follows; (i) dividing the image into N grids, with each region having an equal dimension of SxS, (ii) using bounded boxes to predict objects, (iii) using a loss function, and (iv) using nonmaximal suppression to suppress all bounding boxes that have low probability scores. The YOLO method includes different versions, and YOLO_V5 was used in this study since it gives better results for real-time object detection. YOLO_V5 was chosen rather than the new version: YOLO_V8, due to several reasons. Firstly, YOLO_V5 is significantly advanced regarding improved speed and efficiency. It delivers fast inference times and reduces computational requirements when compared to YOLO_V8. Another important reason is that YOLO_V5 has an active development community and ongoing support, ensuring continuous updates and bug fixes. This active community guarantees access to the latest improvements, making YOLO_V5 a future-proof choice for object detection projects. Furthermore, YOLO_V5 maintains compatibility with previous YOLO versions, facilitating a seamless integration process for users with existing workflows and projects. A comprehensive comparison study on the versions of YOLO is presented in [39]. By considering these factors, YOLO_V5 emerged as a compelling choice for object detection.

The detection process initiates with training, which is essential for machine/deep learning models, as illustrated in Figure 3. In order to extract the training data, a simulation area was formed, and 5 h of video data was captured. Several drivers took the role of simulating their typical activities in order to create the dataset and to train and quantitatively evaluate the proposed method; a total of 3099 images were selected from this video data for use in model training. The dataset has been split into three parts for better evaluation: Train 70%, Validation 20%, and Test 10%. The dataset median image ratio is 960 \times 1080, and the class balance is around an equal ratio. The dependent classes for model training are as follows (Table 5):

Databases:	Driver Behaviors	
Instances:	3099	
Attributes:	10	
Sum of Weights:	286	
No	Attributes	Туре
1	C0_smoking	Nominal
2	C1_talking_passenger	Nominal
3	C2_radio_checking	Nominal
4	C3_reaching_behind	Nominal
5	C4_drinking	Nominal
6	C5_texting	Nominal
7	C6_talking_on_phone	Nominal

Table 5. Dataset class distribution.

The dataset images were preprocessed to decrease training time and increase performance by applying image transformations of Auto Orient, Resize (Stretch to 416 × 416), and Grayscale. Additionally, data augmentation was applied to the driver behavior detection datasets to obtain better datasets and more instances. In this regard, Rotation Between -15° and $+15^{\circ}$ (helps the model be more resilient to camera roll), Shear $\pm15^{\circ}$ Horizontal, $\pm15^{\circ}$ Vertical (adds variability to the perspective to help the model be more resilient to camera and subject pitch and yaw), saturation between -25% and +25% (adjusts the vibrancy of the colors in the images), brightness between -25% and +25% (adds variability to image brightness to help the model be more resilient to lighting and camera setting changes) were



applied. Following the data augmentation, the generated maximum version size of the images included 7437 images.

Figure 3. (a) Steps for proposed method; (b) configuration of YOLO model.

2.4. Driver Behavior Recognition Based on YOLO Differences from Related Studies

In the context of detecting human behavior, machine learning models were compared to determine which model produces better results. Various studies have been conducted on driver behavior detection, such as "Pose Estimation Based Activity Recognition [40]", which employed State Farm Distracted Driver Detection [41]. This study also utilized a distracted driver dataset to achieve better results. Because the datasets used in these studies are similar, model learning can be easily compared. While developing the YOLO_V5 model, in addition to carefully building the database, the State Farm dataset was used to increase the dataset and the performance of the model.

When comparing the two articles, it can be concluded that the YOLO_V5 model proposed in this study is slightly superior to the Pose Estimation model in terms of detection method and accuracy. OpenPose is a pretrained deep learning model for pose estimation and is supported by logistic regression, support vector machine, decision tree, and random forest classifiers. It has performed well in detection, with an accuracy ranging from 0.90 to 0.92. In contrast, the YOLO_V5 model proposed in this study is a real-time object detection algorithm that detects and classifies driver behaviors using a dataset of 3099 images. The model's performance was improved through data augmentation, resulting in a precision level of 0.60.

While both articles use different machine-learning approaches, they fall within the same recognition field. The use of a supported CNN model has led to slow model processing, and typically requires multiple passes to make a decision, as noted in another article [42]. Conversely, YOLO_V5 is a better fit for real-time applications, particularly those with limited resources, such as those running on mobile devices or embedded systems. Because this model is intended to run on an embedded system in a car, a faster and more effective model needs to be used for driver behavior detection. The article also shows that the YOLO_V5 model achieved a precision level of 0.60 using a short learning time. Moreover, more training with additional steps and data produced better results, as explained in the performance results section, due to the recall and precision metrics' relationship with the steps. YOLO_V5 models have achieved better performance in object detection tasks compared to other algorithms and are slightly more efficient and effective, which is crucial in real-time applications.

3. Performance Results

The experimental study was conducted on a platform called Google Colab, which requires n1-highmem-2 instance, 2vCPU @ 2.2 GHz, 12 GB RAM, 100 GB of free space, an idle cut-off of 90 min, and a maximum of 12 h. The test results of the proposed method



illustrate the classification performance, as shown in Figure 4. The proposed approach's mAP (mean average precision) value is 0.3264, and the precision value is around 0.6342.

Figure 4. Training: (a) 100 epochs; (b) 300 epochs; (c) precision/recall correlation.

In machine learning applications, the precision and recall values should be in inverse relationships to achieve high performance. ROC curve tests show that if a model aims to be better than a random model, it is expected that recall and precision have an inverse ratio with time. If the model is better than random, an increase in recall generally decreases precision. If the model is worse than random, precision generally increases. As illustrated in Figure 4a,b, after increasing the epoch numbers, the relationship between precision and recall is inverse, which highlights that the model obtains better results than a random prediction model with an increase in epochs. Additionally, it emphasizes that the model becomes better when using the YOLO approach and when there is enough data/time for model training as illustrated in Figure 5.



Radio

Phone

Texting

Talking

Figure 5. Positive predicted dataset samples.

In order to validate the performance of the proposed 360-degree surround-view system, a blind/referenceless image spatial quality evaluator (brisque) technique was used to calculate the differences between the original frame and modifications [43]. As illustrated in Table 6, while the brisque value of the original frame varies within the range of 18.9–28.5, the brisque value of the modified frame varies within the range of 46.6–56.1. The results highlight that the brisque values are within acceptable limits, as in the methods proposed in the literature. Moreover, the performance of the 360-degree surround-view system was tested by drivers in the case studies of routine driving, parking, and parking near-to-line tests. The performance results were illustrated in Figure 6. Another metric for calculating the performance of a binary classification model [44]. It helps assess the model's ability to correctly identify negative instances by considering the FPR. It can understand the trade-off between correctly identifying positive instances and accurately identifying negative instances.

False Positive Ratio (FPR) = False Positives (FP)/(False Positives (FP) + True Negatives Count (TN)) (4)

$$FP = (1 - Precision) \times TN$$
 (5)

With True Negatives (TN) = 1000;

At threshold 0.05: Precision: 80, Recall: 90

FPR = 0.1667 or 16.67%

At threshold 0.23: Precision: 40, Recall: 70

FPR = 0.375 or 37.50%

Table 6. Brisque scores of original and modified surround view.

Original Surround View	Modified Surround View
24.827	56.108
28.592	46.661
18.911	48.012
25.743	53.881



Figure 6. Performance results of 360-degree surround-view system: front and back view, right and left view, and 360-degree view and resized 360-degree view.

These estimated FPR values are considered to reflect relatively good results. The FPR ranges from 16.67% to 37.50%, indicating that the model has a moderate-to-low rate of false positives. This suggests that the model maintains a good balance between identifying positive instances while keeping false positives relatively low.

According to Figure 4c, precision and recall were positively correlated; as the threshold decreased, precision tended to decrease, while recall also decreased. Additionally, even though the precision values decrease with a decreasing threshold, they remain relatively high. This indicates that the model maintains a good level of accuracy in identifying positive instances. The overall trend suggests that the model can strike a balance between capturing a significant number of positive instances (high recall) while maintaining a reasonably low rate of false positives (relatively high precision). This can be advantageous in various applications where both sensitivity and accuracy are essential.

4. Conclusions

This paper is about the development of an advanced driver assistance system (ADAS). It is important to consider both a 360-degree surround-view car technology system and a driver behavior recognition system via machine learning. These two systems are interconnected, as they both contribute to improving driver safety and preventing accidents on the road. The 360-degree surround-view system enhances the driver's awareness of their surroundings, while the driver behavior recognition system uses machine learning to detect distracted or impaired driving. By including both systems in the article, readers can gain a comprehensive understanding of the current state and potential future advancements of ADAS technology.

This paper proposes (i) a 360-degree surround view to detect blind spots and minimize visibility restrictions and (ii) a YOLO-based driver behavior recognition system to detect inattention/misbehavior to determine suitable vehicle control strategies. Unlike the existing behavior recognition systems, the YOLO-based method was adopted for the ADAS because of its faster prediction capability and fps value. The proposed methods were applied to an electric bus that was equipped with the required hardware cameras. The performance of the proposed methods was tested under different scenarios: routine driving, parking, and parking near-to-line tests, providing a 360-degree surround view, and monitoring radio/phone/texting/talking driver behavior via a recognition system.

In conclusion, the research highlights the significance of integrating 360-degree surround-view car technology systems and driver behavior recognition systems in ADASs. The findings demonstrate the successful implementation and performance of these systems, emphasizing their potential to enhance driver safety and contribute to a comprehensive understanding of ADAS technology advancements.

As a result, the performance of the proposed driver behavior recognition system has been validated, with n mAP value of 0.3264 and a precision value of 0.6342. Besides, the effectiveness of the 360-degree surround view was tested with drivers. It was concluded from the test results performed by the drivers that the proposed system has a positive effect on awareness.

Author Contributions: Conceptualization, M.U.C. and Ç.D.; methodology, M.U.C. and E.Y.; software, Ç.D. and E.Y.; validation, M.U.C. and E.Y.; formal analysis, Ç.D.; investigation, Ç.D.; writing—original draft preparation, M.U.C. and Ç.D.; writing—review and editing, M.U.C., Ç.D., E.Y.; visualization, M.U.C.; supervision, M.U.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ottomotive Mühendislik ve Tasarım A.Ş.

Data Availability Statement: The data are not publicly available due to privacy restrictions.

Acknowledgments: The authors would like to acknowledge Ottomotive Mühendislik ve Tasarım A.Ş. for full financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Cicchino, J.B. Effects of lane departure warning on police-reported crash rates. J. Saf. Res. 2018, 66, 61–70. [CrossRef] [PubMed]
- Hidayatullah, M.R.; Juang, J.C. Adaptive Cruise Control with Gain Scheduling Technique under Varying Vehicle Mass. *IEEE Access* 2021, 9, 144241–144256. [CrossRef]
- Wang, Y.; Wang, Z.; Han, K.; Tiwari, P.; Work, D.B. Gaussian Process-Based Personalized Adaptive Cruise Control. *IEEE Trans. Intell. Transp. Syst.* 2022, 23, 21178–21189. [CrossRef]
- Strišković, B.; Vranješ, M.; Vranješ, D.; Popović, M. Recognition of maximal speed limit traffic signs for use in advanced ADAS algorithms. In Proceedings of the 2021 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021; pp. 21–26.
- Iranmanesh, S.M.; Mahjoub, H.N.; Kazemi, H.; Fallah, Y.P. An Adaptive Forward Collision Warning Framework Design Based on Driver Distraction. *IEEE Trans. Intell. Transp. Syst.* 2018, 19, 3925–3934. [CrossRef]
- Cicchino, J.B. Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates. *Accid. Anal. Prev.* 2017, 99, 142–152. [CrossRef] [PubMed]
- Lee, J.; Kim, M.; Lee, S.; Hwang, S. Real-Time Downward View Generation of a Vehicle Using Around View Monitor System. IEEE Trans. Intell. Transp. Syst. 2020, 21, 3447–3456. [CrossRef]
- 8. Yue, L.; Abdel-Aty, M.A.; Wu, Y.; Farid, A. The Practical Effectiveness of Advanced Driver Assistance Systems at Different Roadway Facilities: System Limitation, Adoption, and Usage. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3859–3870. [CrossRef]
- Gojak, V.; Janjatovic, J.; Vukota, N.; Milosevic, M.; Bjelica, M.Z. Informational bird's eye view system for parking assistance. In Proceedings of the 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 3–6 September 2017; pp. 103–104.

- 10. Kato, J.; Sekiyama, N. Generating Bird's Eye View Images Depending on Vehicle Positions by View Interpolation. In Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, China, 18–20 June 2008; p. 16.
- 11. Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; Sinha, S. Real-Time Video Analytics: The Killer App for Edge Computing. *Computer* **2017**, *50*, 58–67. [CrossRef]
- 12. Al-Hami, M.; Casas, R.; El-Salhi, S.; Awwad, S.; Hussein, F. Real-Time Bird's Eye Surround View System: An Embedded Perspective. *Appl. Artif. Intell.* 2021, *35*, 765–781. [CrossRef]
- 13. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vis. 2004, 60, 91–110. [CrossRef]
- 14. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In *Computer Vision—ECCV 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
- 15. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
- Pan, J.; Appia, V.; Villarreal, J.; Weaver, L.; Kwon, D.K. Rear-Stitched View Panorama: A Low-Power Embedded Implementation for Smart Rear-View Mirrors on Vehicles. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1184–1193.
- 17. *J3016_202104*; Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE International: Warrendale, PA, USA, 2018.
- 18. Pugeault, N.; Bowden, R. How Much of Driving Is Preattentive? IEEE Trans. Veh. Technol. 2015, 64, 5424–5438. [CrossRef]
- 19. Butakov, V.; Ioannou, P. Personalized Driver/Vehicle Lane Change Models for ADAS. Vehicular Technology. *IEEE Trans. Veh. Technol.* 2015, 64, 4422–4431. [CrossRef]
- 20. Martinez, C.M.; Heucke, M.; Wang, F.Y.; Gao, B.; Cao, D. Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 666–676. [CrossRef]
- Hu, J.; Xu, L.; He, X.; Meng, W. Abnormal Driving Detection Based on Normalized Driving Behavior. *IEEE Trans. Veh. Technol.* 2017, 66, 6645–6652. [CrossRef]
- Chai, R.; Naik, G.R.; Nguyen, T.N.; Ling, S.H.; Tran, Y.; Craig, A.; Nguyen, H.T. Driver Fatigue Classification With Independent Component by Entropy Rate Bound Minimization Analysis in an EEG-Based System. *IEEE J. Biomed. Health Inform.* 2017, 21, 715–724. [CrossRef] [PubMed]
- Martin, M.; Voit, M.; Stiefelhagen, R. Dynamic Interaction Graphs for Driver Activity Recognition. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.
- Nel, F.; Ngxande, M. Driver Activity Recognition Through Deep Learning. In Proceedings of the 2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), Potchefstroom, South Africa, 27–29 January 2021; pp. 1–6.
- Xing, Y.; Lv, C.; Wang, H.; Cao, D.; Velenis, E.; Wang, F. Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach. *IEEE Trans. Veh. Technol.* 2019, 68, 5379–5390. [CrossRef]
- Xing, Y.; Lv, C.; Zhang, Z.; Wang, H.; Na, X.; Cao, D.; Velenis, E.; Wang, F.Y. Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition. *IEEE Trans. Comput. Soc. Syst.* 2018, 5, 95–108. [CrossRef]
- Halabi, O.; Fawal, S.; Almughani, E.; Al-Homsi, L. Driver activity recognition in virtual reality driving simulation. In Proceedings of the 2017 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 4–6 April 2017; pp. 111–115.
- Behera, A.; Wharton, Z.; Keidel, A.; Debnath, B. Deep CNN, Body Pose, and Body-Object Interaction Features for Drivers' Activity Monitoring. *IEEE Trans. Intell. Transp. Syst.* 2022, 23, 2874–2881. [CrossRef]
- Zhao, L.; Yang, F.; Bu, L.; Han, S.; Zhang, G.; Luo, Y. Driver behavior detection via adaptive spatial attention mechanism. *Adv. Eng. Inform.* 2021, 48, 101280. [CrossRef]
- Yan, J.; Lei, Z.; Wen, L.; Li, S.Z. The Fastest Deformable Part Model for Object Detection. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2497–2504.
- 31. Lenc, K.; Vedaldi, A. R-cnn minus r. arXiv 2015, arXiv:1506.06981.
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 2015, 28. [CrossRef] [PubMed]
- 34. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Bandyopadhyay, H. YOLO: Real-Time Object Detection Explained. 2022. Available online: https://www.v7labs.com/blog/yoloobject-detection#two-stagedetectors (accessed on 20 April 2023).
- Szeliski, R. Stereo Vision: Introduction and Overview. In Computer Vision: Algorithms and Applications; Springer: Berlin/Heidelberg, Germany, 2010.
- 37. Zhu, X.; Du, X.; Zhang, T.; Wei, Y. 360-Degree Surround View System Based on Super Resolution Convolutional Neural Network. *J. Phys. Conf. Ser.* **2020**, *1621*, 012041.
- Hong, S.; Lee, J.; Lee, D.; Kim, M. An improved 360-degree surround view system using multiple fish-eye cameras. Sensors 2015, 15, 31614–31634.

- 39. Terven, J.; Cordova-Esparza, D.-M. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *arXiv* 2023, arXiv:2304.00501.
- Çetinkaya, M.; Acarman, T. Driver Activity Recognition Using Deep Learning and Human Pose Estimation. In Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Kocaeli, Turkey, 25–27 August 2021.
- Smith, J. State Farm Distracted Driver Detection. Kaggle. 2016. Available online: https://www.kaggle.com/c/state-farmdistracted-driver-detection (accessed on 20 April 2023).
- 42. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. arXiv 2018, arXiv:1804.02767.
- Mittal, A.; Moorthy, A.K.; Bovik, A.C. Blind/Referenceless Image Spatial Quality Evaluator. In Proceedings of the 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, USA, 6–9 November 2011; pp. 723–727.
- 44. Triki, N.; Karray, M.; Ksantini, M. A Real-Time Traffic Sign Recognition Method Using a New Attention-Based Deep Convolutional Neural Network for Smart Vehicles. *Appl. Sci.* **2023**, *13*, 4793. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.