İSTANBUL UNIVERSITY CERRAHPAŞA

ORIGINAL ARTICLE
Electrica 2023; 23(3): 607-618

Deep Learning-Based Driver Assistance System

Barışcan Kurtkaya, Arda Tezcan, Murat Taşkıran

Department of Electronic and Communication Engineering, Yıldız Technical University, İstanbul, Turkey

Cite this article as: B. Kurtkaya, A. Tezcan and M. Taşkıran, "Deep learning-based driver assistance system," *Electrica*, 23(3), 607-618, 2023.

ABSTRACT

Nowadays, vehicles have become an integral part of our lives due to mobility advantages. However, traffic accidents continue to occur worldwide. This study aims to develop a pure image-based solution using a combination of "deep learning" and "image processing" techniques to minimize the occurrence of traffic accidents. While the You Only Look Once (YOLO) algorithm is one of the fastest object detection algorithms, it faces slight accuracy and robustness problems. Afterward, the YOLO algorithm with Darknet-53 architecture, which is pretrained with COCO Dataset, has faced reliability issues to detect objects in "Night" images while getting high results on "Day" images. Therefore, we suspect that the COCO Dataset is inclined toward brighter images rather than low-light ones. To support this idea with scientific evidence, we analyzed the COCO Dataset. Besides, to overcome this issue, fine-tuning and classifier filter designs have been proposed. Additionally, lane detection systems were developed to improve the reliability of the feedback system. As a result, the classifier filter system achieved 99.92% accuracy in distinguishing between "Night" and "Day" images. After evaluation processes, the proposed system achieved ~0.92 IOU with YOLOV3 fine-tuned model and ~0.95 IOU with YOLOV4 fine-tuned model. Furthermore, the lane detection algorithm achieved 88.00% accuracy.

***Index Terms*—**Day and night classification, object detection, driver assistance system

I. INTRODUCTION

Transportation plays a vital role in human life, particularly with the increasing population growth. However, it also brings about several problems, such as traffic jams and accidents. Therefore, it is crucial to establish rules and regulations to regulate traffic and prevent accidents. However, statistics show that even with government regulations, these problems continue to persist in human life. According to data released by the Turkish Statistical Institute (TurkStat), the lowest number of traffic accidents recorded between 2009 and 2019 was 1.053.346 in 2009, while the highest number was 1.313.359 in 2015. Additionally, the same report mentions that 1.168.144 accidents occurred, and 174.896 of them resulted in death or injury. Moreover, a total of 5.473 people died and 283 234 people were injured in these accidents. TurkStat also revealed that 88% of these accidents were caused due to driver defects in the first place [1]. In addition, as reported by the National Highway Traffic Safety Administration (NHTSA) in the United States, a total of 33 654 fatal accidents were experienced in 2018 [2]. Additionally, the same institution conducted a study that found in 2018, about 400.000 people were injured in accidents caused by distracted drivers' negligence [3]. Besides that, as shown in another research on accidents, 70% of individuals involved in car-to-pedestrian incidents were positioned in front of the vehicle. Furthermore, among these individuals, 90% were in motion at the time of the accident [4].

Throughout recent technological developments, companies can manufacture fully autonomous, semi-autonomous, or partially autonomous vehicles. Despite that, the manufacturing costs are increasing due to slower development [5], additional sensors, and other factors. To reduce manufacturing costs while maintaining optimal performance, it is crucial to develop collision warning systems, adaptive cruise control, lane control systems [6], and other related technologies with maximum accuracy and minimal cost.

The NHTSA has classified the level of automation in vehicles into six levels [7]. The first level, which is described as level 0, represents zero automation in vehicles. This means that the driver is fully responsible for all decisions and the car is under human control. None of the car's systems,

Corresponding author:
Barışcan Kurtkaya

E-mail:
bariscankurtkaya@gmail.com

Received: September 17, 2022

Accepted: May 23, 2023

Publication Date: July 20, 2023

DOI: 10.5152/electr.2023.22152

including the brake, steering, throttle, and motive power, are dependent on any autonomous system. Levels 1 and 2 of vehicle automation are quite similar, with level 2 providing continuous assistance to both acceleration/braking and steering, while level 1 only provides assistance to one of them. As the automation level increases, the technology in cars gradually improves. However, in the United States of America, automation levels above level 2 are not yet available for consumer purchase as levels 3 to 5 have the ability to fully control the car autonomously.

This paper proposes a level 1 system for improving driver safety and comfort through various image-based operations and a feedback module. The system aims to enhance the driver's experience by providing lane tracking, detecting objects, calculating their distance, and utilizing a prioritization algorithm. As this paper does not include direct interventions in acceleration or braking, which require extensive electronic and mechanical devices, the proposed system falls into the level 1 category.

The proposed system uses four different image-based operations, including three image processing techniques and a deep learning-based object detection algorithm. To prevent overfitting of the object detection algorithm to specific object colors, data augmentation techniques such as color shifting and flipping were used [8]. These techniques not only helped to protect the convolutional neural network (CNN) model from overfitting but also increased the amount of available data, resulting in improved model performance.

Image preprocessing techniques remain valuable for improving the accuracy of driver assistance systems. Well-known preprocessing techniques, such as noise cancellation, classification, data cleaning, and transformation [9, 10, 11], can help to map the input dataset space to a more diverse one, thereby increasing the system's robustness. For instance, the Laplacian [12] and Sobel Feldman [13] filters are popular for extracting Salt and Pepper [14] and Gaussian noise [12] from images. Furthermore, it is common to use filters to classify data with a threshold. In this case, the filter is specifically designed to amplify the differences between the most important properties of the two classes. Also, lane detection is a common task in the literature and is often performed using the Hough Linear Transform algorithm [15], which detects points in the same direction by analyzing their slope and bias [16].

While driver assistance systems and autonomous vehicles are popular topics in the literature, many papers build their systems using expensive sensors, such as lidar sensors [17, 18, 19]. In contrast, the objective of this paper is to create an only image-based application that reduces the occurrence rate of traffic accidents. In addition to building a robust system, it is essential for the system to be real-time. Thus, the system was implemented using the YOLO algorithm to meet these demands. Despite having high mAP results on the COCO dataset with pretrained Darknet-53, both YOLOv3 [20] and YOLOv4 [21] algorithms performed poorly on night images from the Oxford RobotCar (ORC) dataset. To overcome this issue, Darknet-53 model was fine-tuned with ORC dataset [22] in this article.

In addition, there are several object detection algorithms available other than YOLOv3 and YOLOv4 [23], such as Fast R-CNN [24], Faster R-CNN [25], Mask R-CNN [26], RetinaNet [27], and SSD [28]. Although RetinaNet may provide higher accuracy than YOLOv3, it is worth noting that YOLOv3 is approximately 1.5 times faster than the fastest RetinaNet50-500 [20]. Additionally, both Faster R-CNN and RetinaNet have lower performance in terms of accuracy and running speed compared to YOLOv4 [20, 23].

Moreover, since both YOLOV3 and YOLOV4 algorithms utilize CNNs, the selection of an appropriate network for accurate image feature extraction is crucial. Some commonly used networks include Darknet-53 [20], Darknet-19 [20], ResNet-101 [29, 31], ResNet-152 [29], VGG-16 [30], and VGG-19 [30].

II. APPROACH

Autonomous driving is a crucial concept for the future of transportation, and it is a hot topic in the literature. As even minor issues can result in fatalities, it is essential to build a robust system. Hence, this study implemented two specialized deep learning models for different weather conditions. Additionally, a condition classifier filter was designed to select the appropriate model for the current state. With this basic filter design, the system still achieved excellent results during the day, while improving its accuracy score by more than 400% in the night images for object detection. Additionally, in order to enhance the precision of the feedback, an algorithm for controlling the location of detected objects is employed. This algorithm utilizes the Hough Transform technique to estimate the road lane and filters out any objects that are not close to the estimated cruise track. After verifying the relative track position, the algorithm estimates the distance and provides feedback, as illustrated in Fig. 1. Lastly, a final control algorithm is in place to ensure that the vehicle remains between the lanes throughout its course.

A. Preprocessing and Classifying the Day and Night

This study evaluates the competency of the pretrained YOLO algorithm backbone, trained on the COCO dataset, in the classification of cars and individuals under low light scenarios like nighttime imagery. The study aims to shed light on the limitations and deficiencies of the algorithm in this particular scenario and propose a system to overcome them. To maintain accuracy in day images while improving accuracy in night images, a classification system was designed to select the appropriate deep learning model. Initially, the most basic classification method of thresholding the average of the image pixels was used to address this issue. However, this approach resulted in an overlapping problem, as illustrated in Fig. 2. To overcome this problem, a preprocessing design was proposed, which involved creating a filter to maximize the divergence between the two classes. Procedure 1 outlines the algorithm for creating the filter. First, the average night and day images were generated. Next, the night average image was subtracted from the day average image to identify the most significant differences, as illustrated in Fig. 3. Finally, the filter's pixel values were assigned using a 10-tier ranking algorithm that divides the values between 0 and 255 into 10 groups and assigns a coefficient that is between 0 and 9 based on the pixel value.

After creating the filter, the input images are convolved with the designed filter, and the resulting matrix is divided by the count of non-zero pixels. This process is equivalent to taking the average of the Hadamard product of the input image with the filter, excluding zero elements. As a result, more divergent image averages are obtained, as shown in Fig. 4. Using a simple threshold, a success rate of 99.92% was achieved.

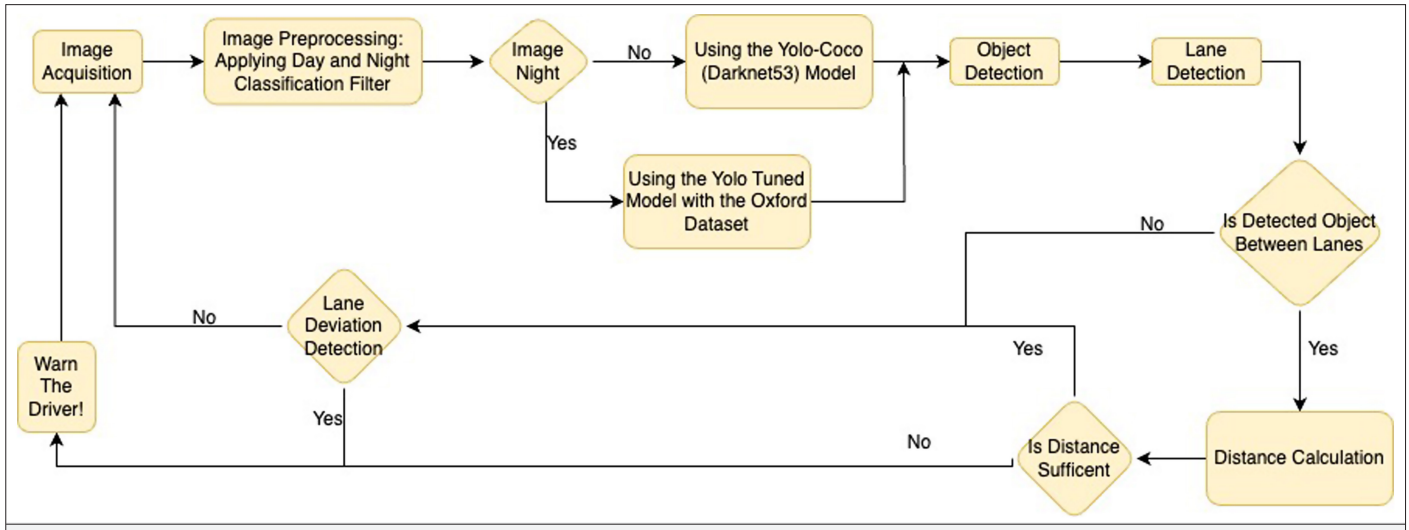


Fig. 1. The block diagram of proposed method.

Procedure 1 Filter Creation Algorithm

1: **procedure** FilterCreation (n_i, d_j)
2: $n_{avg} = \frac{1}{i} * \sum_i n_i$
3: $d_{avg} = \frac{1}{j} * \sum_j d_j$
4: $\Delta = n_{avg} - d_{avg}$
5: for k in len(Δ):
6:for l in len(Δ[k]):
7:fi[k][l] = 10-tier ranking algorithm (Δ[k][l])

Procedure 2 Filter Algorithm

1: **procedure** Filter (f_{ij}, fi_{ij}, threshold)
2: $f_{ij} \odot fi_{ij}$ (Hadamard Product)
3: $f_{avg} = \frac{1}{i*j} * \sum_j \sum_i f_{ij} \left(without\ 0\ valued\ pixels \right)$
4: $\Delta = f_{avg} - threshold$
5: if Δ > 0 (day)
6: else (night)

B. Object Detection

The selected model after day and night classification was utilized to enhance driver safety in real-time by implementing YOLOV3 and YOLOV4 as object detection algorithms, thereby improving system robustness and reducing latency. The architectures of YOLOV3 and YOLOV4 were retained as described in their respective papers and are depicted in Fig. 5 and Fig. 6, respectively. The last convolution layer of the models' output had 21 channels, given that only pedestrian and car classes were present, calculated using the formula (classes + 5) * 3 from YOLOV3's paper.

YOLOV3 utilized Darknet53 CNN and Feature Pyramid Network (FPN) [31] to aggregate parameters from different backbone levels for different detector levels. In contrast, YOLOV4 employed CSP-Darknet53 backbone and Path Aggregation Network (PANet) [32] and also employed Spatial Pyramid Pooling (SPP) [33].

C. Lane Estimation

After performing object detection, this paper also includes the estimation of road lanes to address two additional main issues. First, this study controls the road lanes in relation to the car to maintain the cruise track. Second, to increase the robustness of the system by decreasing false alarms, the position of the detected objects is controlled with respect to the road lanes, in other words, the estimated cruise track.

To achieve more accurate road lane estimation, we propose a region of interest (ROI) as a hyperparameter. This ROI covers the area of the car's potential track with a margin. Within this region, we suggest using a triangle whose base lies on the car's front bumper, and its top point indicates the most likely direction of the car. Subsequently, the region of interest mask will be applied to prevent extraneous parts of the car and other objects in the image. This approach provides an advantage in improving the robustness of the lane estimation process.

Subsequently, the image undergoes Canny edge detection to eliminate any noise, utilizing a 5×5 Gaussian filter, and to detect edges using a 3×3 Sobel Edge Kernel. Next, the Hough Transform is applied to identify the road lines in the filtered image. The Hough Transform utilizes the classic linear function formula (Formula 1) to detect road lines by converting x-y coordinates into slope and bias coordinates. It scans all points and generates linear lines in these coordinates. It then selects the most intersected points and creates a line with this linear function formula.

In the final step, the captured image is partitioned into two regions in the middle, and the Euclidean distance is applied to identify the two closest lanes to the central zero point. A linear regression

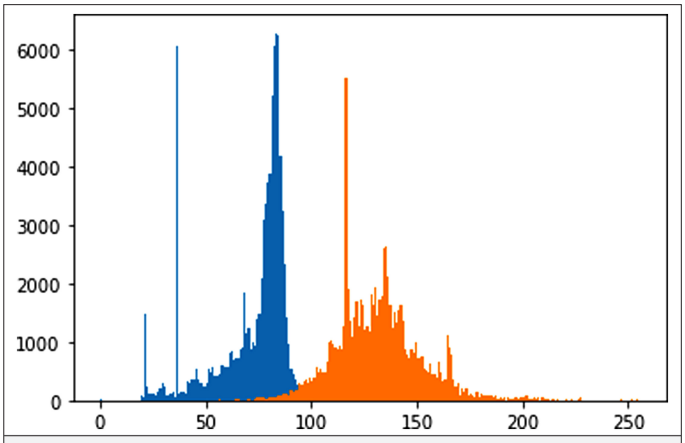


Fig. 2. Image pixel averages with filter.

Fig. 3. Night average image subtraction from day average image.

is performed using the x0-y0 and x1-y1 points to extrapolate the lane lines. The resulting values are compared with a predetermined threshold, and if they fall below it, an alert is issued to the driver via the interface. After the lane estimation process, the algorithm produces the final output, as illustrated in Figure 7.

$$\left(y_1 - y_0\right) = m\left(x_1 - x_0\right) + bias \tag{1}$$

D. Distance Calculation

The proposed driver assistance system employs a distance calculation method to assess the proximity of other vehicles and their potential threat to the user. Specifically, the system utilizes the coordinates of the bottom corners of the prediction boxes to generate feedback. These coordinates' X values are fed into the two linear lines' functions that the road lane detection algorithm determines. The resulting Y coordinate values are then compared to the Y coordinate values of the road lanes. If the Y value of a point is lower than the Y values of both lines, the object is deemed within the estimated road lanes and is considered a potential threat. For identified vehicles that could pose a threat, the system calculates the prediction box's area using the [length × width] formula. If the box's size

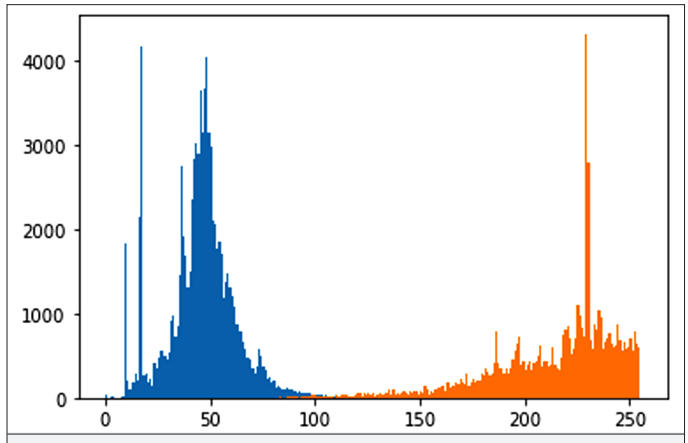


Fig. 4. Image pixel averages without filter.

exceeds a predefined threshold, the system classifies these vehicles as potential threats.

E. Feedback Selection

There are five states in the feedback mechanism. Also, simple example is shown in Fig. 8.

Procedure 1 Filter Creation Algorithm

```
if (Another vehicle in the lane is too close):
    display_stop_sign()
else if (Human detected inside the image):
    display_sidewalk_sign()
else if (Vehicle is in lane and away):
    display_slow_down_sign()
else:
    if (the vehicle is between the lanes):
        display_checked_sign()
    else:
        display_lane_control_correction_signs()
```

III. EXPERIMENTS

This section presents the experimental results and implementation details of the proposed system. First, the training procedure and dataset are described. Second, a detailed account of the hardware and software components is provided. Lastly, the results are presented in order to evaluate the performance of the proposed models.

A. Dataset

The methodology proposed in this paper utilizes two distinct models. The first model is pretrained with the COCO dataset [34], while the second model is fine-tuned with the ORC dataset [22] for various class labels. This approach is chosen because the pretrained model with COCO dataset performs poorly in night images but excels in day images. To maintain its efficiency on day images, a model selection algorithm is devised. Furthermore, this paper compares the

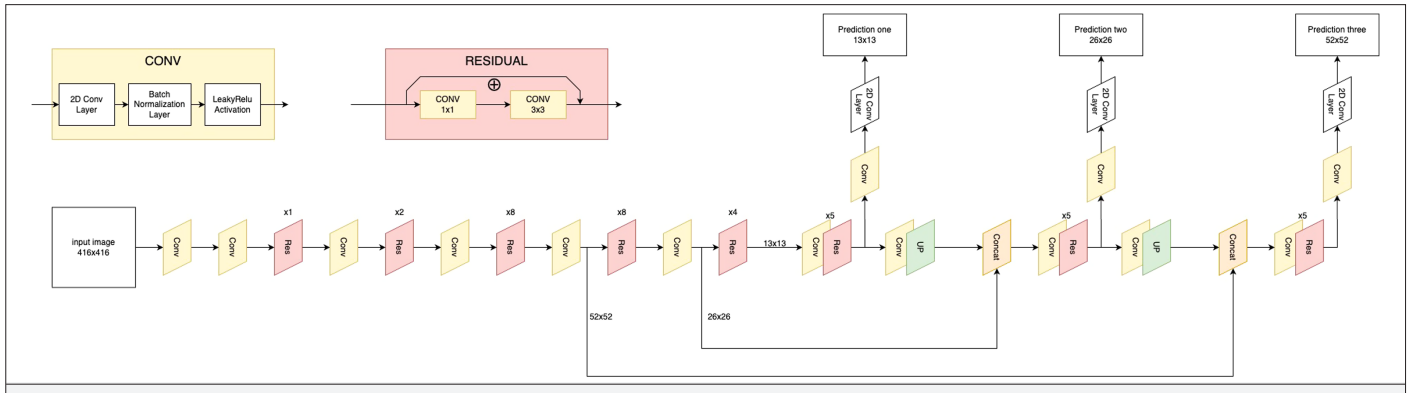


Fig. 5. YOLOV3 model architecture.

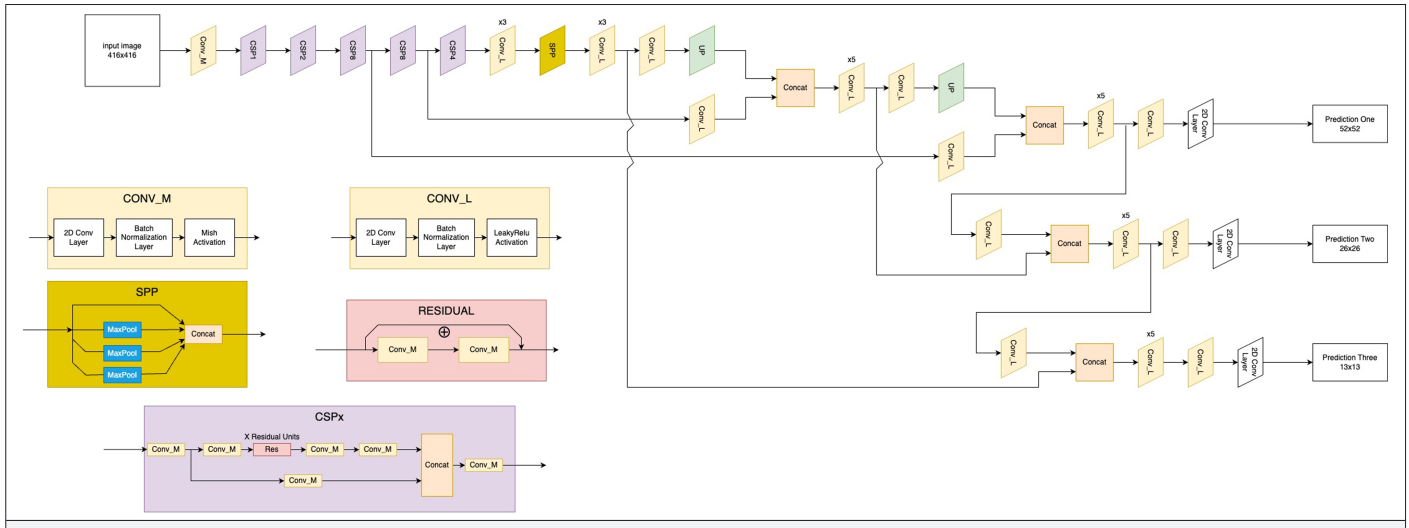


Fig. 6. YOLOV4 model architecture.

pixel probability mass functions of COCO and ORC, as shown in Fig. 9, Fig. 10, and Fig. 11 to elucidate the reasons for the pretrained model's limitations.

The ORC dataset was gathered by the Oxford Team between May 2014 and December 2015, using the ORC platform, an autonomous Nissan LEAF. The vehicle covered a route through central Oxford twice a week on average during this period, capturing a diverse range of images in various driving conditions.

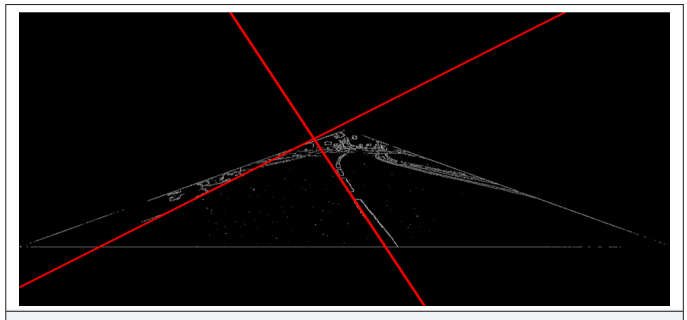


Fig. 7. Lane prediction algorithm output.

The dataset utilized in this study comprises data collected from seven distinct sensors and includes five distinct data types, as illustrated in Table I. Notably, the ORC dataset also encompasses a broad range of weather conditions, as demonstrated in Fig. 12. Specifically, the training dataset employed a Point Grey Bumblebee XB3 (BBX3-13S2C-38) trinocular stereo camera with a resolution of 1280 × 960 × 3, capable of capturing 16 frames per second using a 1/3″ Sony ICX445 CCD sensor. The dataset utilized in this study specifically features data captured by the Bumblebee XB3 camera, consisting of a total of 11.070.651 samples and occupying 13.78 TB of storage space. Additionally, Fig. 2 showcases a variety of samples captured by the Bumblebee XB3 under various environmental conditions.

For all the training processes, a training dataset comprising 18.505 night-time and 17.987 daytime images was selected for filter design, deep learning models, and lane estimation algorithms. To improve the quality and quantity of the deep learning model's dataset, several preprocessing techniques were employed, including random saturation, hue, and contrast shifting. As a result of these techniques, the size of the night training dataset was augmented to 100 000 images. The dataset was then split into three subsets, with 20% (20.000) allocated for testing, 10% (10.000) for validation, and the remaining 70% (70.000) for training the deep

Fig. 8. Example of the feedback.

learning model. It is important to note that the test and training datasets were separated prior to data augmentation. Therefore, during the fine-tuning process, the model did not encounter any variations in the test images.

B. Experiment Results

In the experiment section, three main results were obtained from the individual system components. First, the filter classification results were presented, including the results before and after applying the filter. Table II demonstrates that the filter produced with the ten-tier ranking algorithm significantly improved the overall success rate in percentage. Additionally, the impact of the filter produced with the ten-tier ranking algorithm can be observed in the dataset image pixel averages, as shown in Fig. 4 and Fig. 13.

Second, the success rates of YOLOV3 and YOLOV4 models pre-trained with COCO and fine-tuned with ORC models were compared on ORC night images. The architectures of both models are illustrated in Fig. 5 and Fig. 6. YOLOV3 uses Darknet53 CNN and FPN [32] as the method of parameter aggregation from different backbone levels for different detector levels, while YOLOV4 employs CSP-Darknet53 backbone and PANet [33]. Additionally, YOLOV4 uses SPP [33] as well.

The hardware used for training and evaluation included a GTX 1050 GPU with 640 CUDA cores, a base frequency of 1354 MHz, and a boost frequency of 1455 MHz. The PC was equipped with an Intel i7-7800X CPU with a clock speed of 3.8 GHz and 16GB of RAM. The operating system used was Ubuntu 20.04.

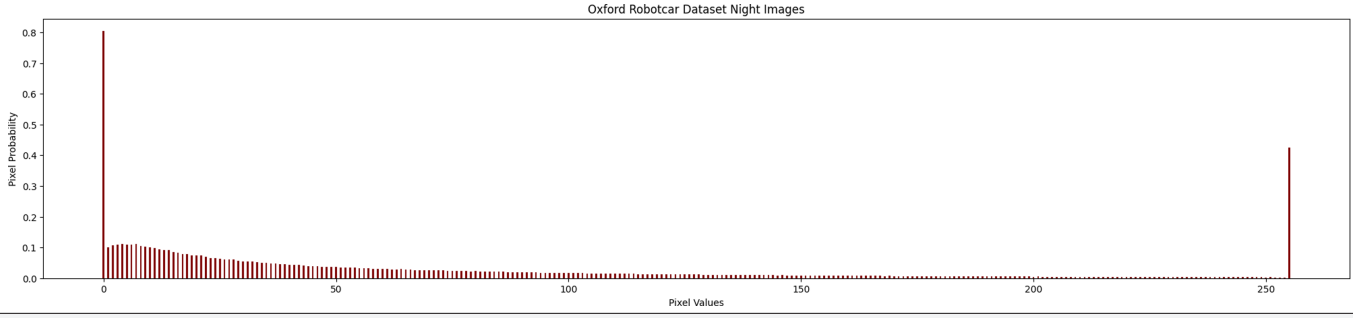


Fig. 9. ORC's night images' PMF. ORC, Oxford RobotCar; PMF, probability mass function.

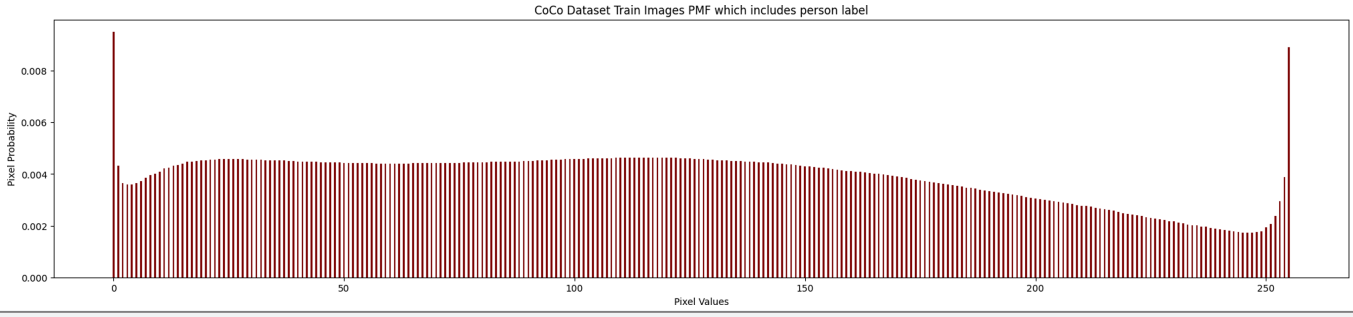


Fig. 10. COCO train images with car label's PMF. PMF, probability mass function.

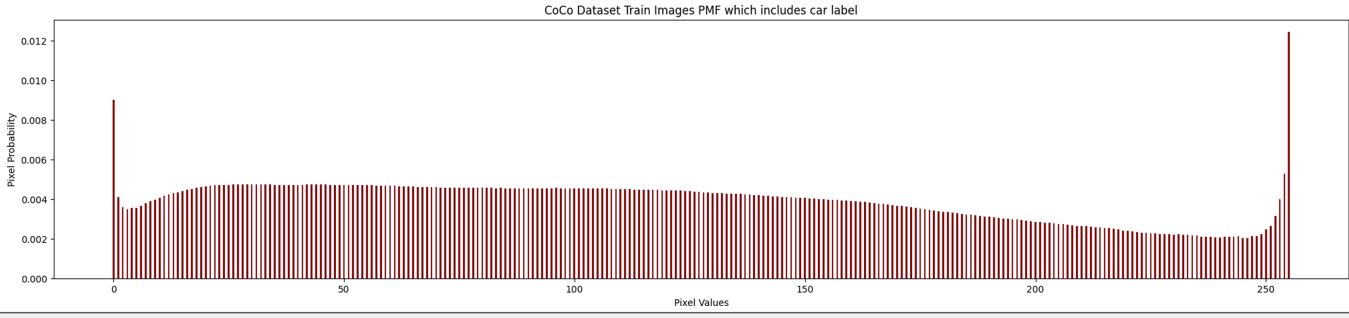


Fig. 11. COCO train images with person label's PMF. PMF, probability mass function.

TABLE I. COLLECTED DATA SUMMARY STATISTICS [24]

Sensor	Type	Count	Size
Bumblebee XB3	Image	11 070 651	13.78 TB
Grasshopper2	Image	8 485 839	9.08 TB
LMS-151	2D Scan	25 618 605	255.95 GB
LD-MRS	3D Scan	3 226 183	31.76 GB
SPAN-CPT GPS	3D Position	1 188 943	496 MB
SPAN-CPT INS	6DoF Position	11 535 144	4.74 GB
Stereo VO	6DoF Position	3 690 067	422 MB

In object detection models, IOU and mAP results are commonly used for evaluation, as reported in the literature [35]. Initially, YOLOV3 algorithm with pretrained backbone was implemented and evaluated but failed to produce satisfactory results, as illustrated in Fig. 14. Subsequently, YOLOV4 with pretrained backbone was implemented, but it also failed to perform well on the ORC dataset, as seen in Fig. 15 and Fig. 16. The IOU threshold of 50% was selected for evaluation, and the results are presented in Table III. The evaluation results show that there are significant reliability and overfitting issues with YOLO models that have pretrained backbones on COCO dataset. However, after fine-tuning both YOLOV3 and YOLOV4 algorithms with the ORC dataset, their evaluation metrics improved by more than 5 times. Test examples of the two models fine-tuned with the ORC dataset are shown in Figs. 17,18, 19, and 20.

This paper also includes a lane detection algorithm which aims to predict road lanes for enhancing driver safety. The table presents the

TABLE II. FIRST AND LAST CLASSIFICATION OF DAY AND NIGHT

	First Classification		Last Classification	
	Night	Day	Night	Day
Total images	18505	17987	18505	17987
Error count	26	441	0	29
Percentage of success	0.14%	2.45%	0.00%	0.16%
Total error percentage	1.28%		0.08%	

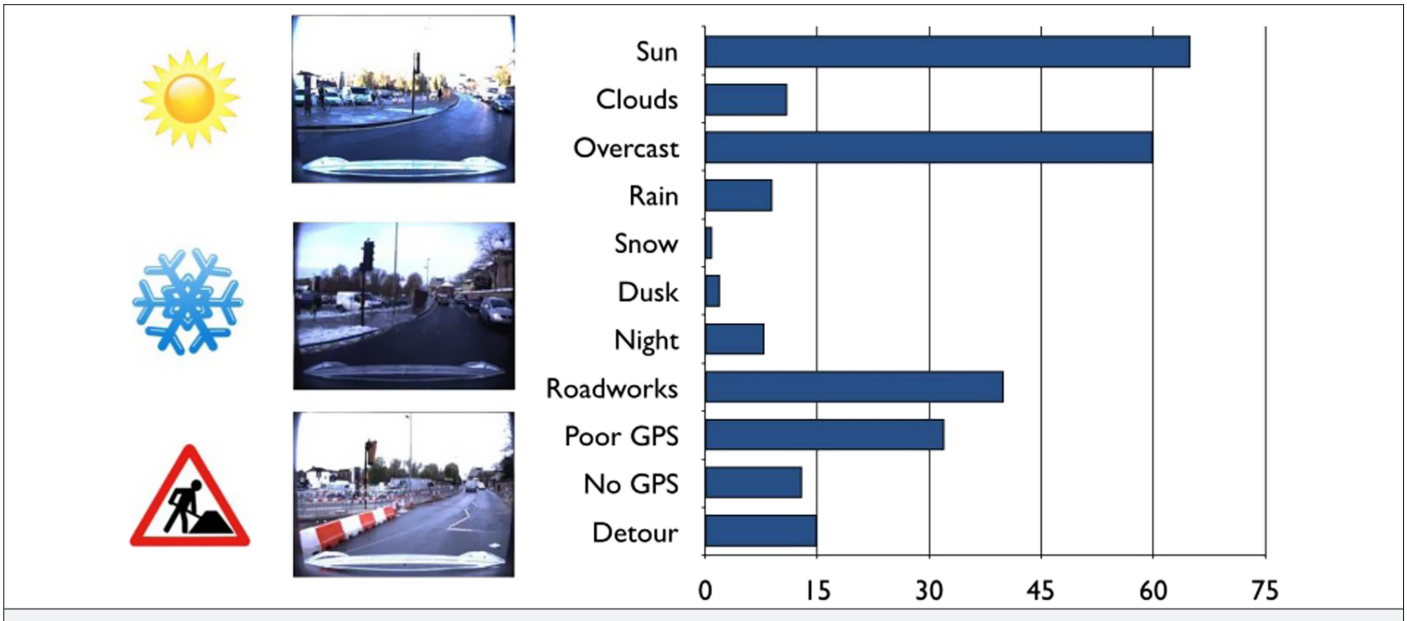


Fig. 12. Oxford RobotCar Dataset condition variousity [23].

Fig. 13. Sample pictures from dataset.

lane detection algorithm's results (Table IV). The same dataset used to test the deep learning models was also used to evaluate the lane prediction algorithm.

IV. CONCLUSION

The current investigation proposes a computer vision system that addresses the safety of nighttime driving using a complex framework. Despite the system's development, difficulties arose due to the YOLOV3 and YOLOV4 failures. While these models are highly accurate in daylight conditions, their performance dramatically decreases to IOU values of 0.09 and 0.17, respectively, in low-light conditions. Therefore, fine-tuning was crucial in this scenario. Following the fine-tuning process, YOLOV3 and V4 models achieved IOU scores of 0.92 and 0.95, respectively. However, to sustain the excellent object detection score, a filter classification system was created. With this filter, day and night classification accuracy of 99.92% was achieved. In addition to deep learning-based object detection, the proposed system offers superior safety measures compared to similar studies by detecting lane boundaries with an 88% accuracy and alerting the driver about the distance to other vehicles, pedestrians' presence, and lane violations.

Fig. 14. YOLOV3 pretrained test example.

Fig. 15. YOLOV4 pretrained test example.

Fig. 16. YOLOV4 pretrained test example.

TABLE III. TEST RESULTS MODEL WITH YOLOV3/YOLOV4 WITH COCO DATASET

	IOU	mAP
YOLOV3 pretrained with COCO	0.0936	0.1286
YOLOV4 pretrained with COCO	0.1698	0.1863
YOLOV3 fine-tuned with ORC	0.9155	0.9241
YOLOV4 fine-tuned with ORC	0.9514	0.9657

However, there is still room for further expansion and improvement. For instance, object movement prediction in traffic studies, which is well-documented in the literature, can be implemented in this project to enhance the system's performance. Additionally, this project is limited to simulation tests using the ORC dataset, and conducting experiments on real-world scenarios may provide valuable insights. With more budget, the development of specialized hardware can be considered. Furthermore, exploring newer versions of YOLO, such as

Fig. 17. YOLOV3 fine-tuned test example.

Fig. 18. YOLOV4 fine-tuned test example.

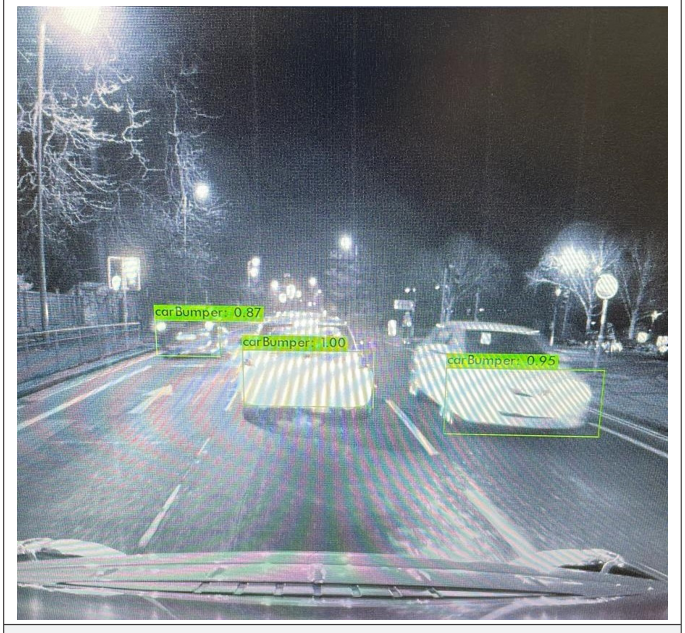


Fig. 19. YOLOV4 fine-tuned test example.

Fig. 20. YOLOV4 fine-tuned test example.

TABLE IV. ROAD LANE ESTIMATION CONFUSION MATRIX

		Ground Truth	
		Exist	**Not Exist**
Prediction	Exist	417	27
	Not Exist	33	23
Lanes' success percentage		88.00%	

YOLOv5, v6, or v7, can be a promising direction for future research. Another possible avenue for future research is the development or implementation of a meta-learning system to reduce dependence on specific datasets. Additionally, applying explainable artificial intelligence methods to understand YOLOv3 and YOLOv4 pretrained models on the dataset could provide valuable insights to address the dependency issue.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – B.K., A.T.; Design – B.K., A.T.; Supervision – M.T.; Data Collection and/or Processing – B.K., A.T.; Analysis and/or Interpretation – B.K., A.T.; Literature Review – B.K., A.T.; Writing – B.K., A.T.; Critical Review – M.T.

Declaration of Interests: The authors have no conflicts of interest to declare

Funding: The authors declared that this study has received no financial support.

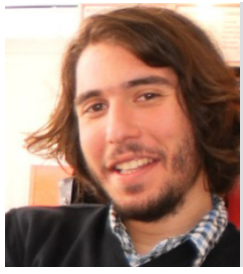
REFERENCES

1. TÜİK Kurumsal, 2019. Retrieved 31 December 2020, from https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2019-33628
2. National Highway Traffic Safety Administration, 2018. Retrieved 31 December 2020, from https://www-fars.nhtsa.dot.gov/Main/index.aspx
3. U.S. Department of Transportation, *NHTSA TRAFFIC SAFETY FACTS, 2020, Distracted Driving 2018*, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812926.
4. D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, 2010. [CrossRef]
5. M. McFarland, "*Tesla raises price of feature it calls 'full self-driving' to $15,000 | CNN Business*". CNN, Cable News Network. Available: https://edition.cnn.com/2022/08/22/business/tesla-fsd-price-increase/index.html
6. "*Autonomous vehicle technology - Auto-mat.ch*." [Online]. Available: https://www.automat.ch/wAssets/docs/170901_laekjautonomous_vehicle_technology.pdf. [Accessed: 15-May-2022].
7. "*Automated vehicles for safety*," NHTSA. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety
8. E. S. Uysal et al., "Exploring the limits of data augmentation for retinal vessel segmentation." *arXiv Preprint ArXiv:2105.09365*, 2021.
9. J. Chaki and N. Dey, *A Beginner's Guide to Image Preprocessing Techniques*. Boca Raton, United States of America: CRC Press, 2018.
10. T. Jaiswal and S. Rajesh, "Image noise cancellation by lifting filter using second generation wavelets (LFSGW)," International Conference on Advances in Recent Technologies in Communication and Computing. IEEE Publications, 2009, https://dl.acm.org/doi/proceedings/10.5555/1673067.
11. F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 1–33, 2012. [CrossRef]
12. F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed Poisson–Gaussian noise," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, 2011. [CrossRef]
13. I. Sobel et al., "A 3x3 Isotropic Gradient Operator for Image Processing," *Pattern Classification and Scene Analysis* , 1973.271272
14. J. Azzeh, B. Zahran, and Z. Alqadi, "Salt and pepper noise: Effects and removal," *JOIV Int. J. Inform. Vis.*, vol. 2, no. 4, pp. 252–256, 2018.
15. W. Liu, Z. Zhang, S. Li, and D. Tao, "Road detection by using a generalized Hough transform," *Remote Sens.*, vol. 9, no. 6, p. 590, 2017. [CrossRef]
16. J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vis. Graph. Image Process.*, vol. 44, no. 1, pp. 87–116, 1988.
17. M. Aryal, "*Object detection, classification, and tracking for autonomous vehicle*,"[Masters Theses. 912], 2018. https://scholarworks.gvsu.edu/theses/912
18. H. Gao et al., "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4224–4231, 2018.
19. X. Zhao et al., "Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sens. J.*, vol. 20, no. 9, pp. 4901–4913, 2020.
20. J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement," *arXiv Preprint ArXiv:1804.02767*, 2018.
21. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object Detection," *arXiv Preprint ArXiv:2004.10934*, 2020.
22. *Oxford RobotCar Dataset*, 2020. [Online]. https://robotcar-dataset.robots.ox.ac.uk/. [Accessed: 28-Aug-2022].
23. Z.Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019. [CrossRef]
24. R. Girshick, "Fast r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. CentroParque Convention Center in Santiago, Chile.
25. S. Ren et al, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
26. K. He et al, "Mask r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969. Venice, Italy.
27. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, 318–327, 2020. [CrossRef]
28. J. -a. Kim, J. -Y. Sung, and S. -h. Park, "Comparison of faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," in IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1–4. [CrossRef]. Seoul, South Korea.
29. K. HE et al, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
30. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Preprint ArXiv:1409.1556*, 2014.

31. T.-Y. Lin et al., "Feature pyramid networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
32. S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 8759–8768. [CrossRef]
33. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015. [CrossRef]
34. T.-Y. Lin et al., "'Microsoft coco: Common objects in context," Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Springer International Publishing, 2014.
35. N. Sharma and R. D. Garg, "Cost reduction for advanced driver assistance systems through hardware downscaling and deep learning," *Syst. Eng.*, vol. 25, no. 2, pp. 133–143, 2022.

Barışcan Kurtkaya is a senior Electronics and Communication Engineering student at Yıldız Technical University, Istanbul, Turkey. He is currently undergraduate researcher at the Princeton University. Also, he was an visiting research intern at AIS-Lab University of Milan and Meda-Lab at Yıldız Technical University. He has several awards with CAS-Marine semi-autonomous underwater vehicle systems. His research interests are mostly the area of self-supervised and/or unsupervised learning and work toward shifting and advancing our understanding of artificial intelligence.

Arda Tezcan graduated B.Sc. (2022) degree in Electronics and Communication Engineering from Yıldız Technical University (YTU), Istanbul, Turkey. He has several experience as an engineer in several companies. Also, he was an undergraduate research assistant at MedaLAB at Yıldız Technical University. Right now, he is working as a Cloud Big Data Engineer for Garanti BBVA Teknoloji since 2022. His research interests are in image processing, deep learning, and data analysis.



Murat Taşkıran received B.Sc. (2013), M.Sc. (2016), and Ph.D. (2022) degrees in Electronics and Communication Engineering from Yildiz Technical University (YTU), Istanbul, Turkey. Since 2022, he has been working as a lecturer in the Department of Electronics and Communication Engineering at YTU. His research interests are in image processing, neural networks, and randomness analysis.