INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 1, PP. 51-58
Manuscript received November 14, 2023; revised March, 2024. doi: 10.24425/ijet.2024.149514

The system of headlights operation recognition using the digital twin method

Aleksander Dawid, and Paweł Buchwald

Abstract—**Virtual digital representation of a physical object or system, created with precision through computer simulations, data analysis, and various digital technologies can be used as training set for real life situations. The principal aim behind creating a virtual representation is to furnish a dynamic, data-fueled, and digital doppelgänger of the physical asset. This digital counterpart serves multifaceted purposes, including the optimization of performance, the continuous monitoring of its well-being, and the augmentation of informed decision-making processes. Main advantage of employing a digital twin is its capacity to facilitate experimentation and assessment of diverse scenarios and conditions, all without impinging upon the actual physical entity. This capability translates into substantial cost savings and superior outcomes, as it allows for the early identification and mitigation of issues before they escalate into significant problems in the tangible world. Within our research endeavors, we've meticulously constructed a digital twin utilizing the Unity3D software. This digital replica faithfully mimics vehicles, complete with functioning headlamp toggles. Our lighting system employs polygons and normal vectors, strategically harnessed to generate an array of dispersed and reflected light effects. To ensure realism, we've meticulously prepared the scene to emulate authentic road conditions. For validation and testing, we integrated our model with the YOLO (You Only Look Once) neural network. A specifically trained compact YOLO model demonstrated impressive capabilities by accurately discerning the status of real vehicle headlamps. On average, it achieved an impressive recognition probability of 80%, affirming the robustness of our digital twin.**

Keywords—**digital twin; road safety; headlamps operation detection; lighting model; Unity3D; YOLOv7**

I. Introduction

THE lighting system of a vehicle plays a crucial part in guarantee the safety of both the driver and other road users. This intricate car's subsystem serves several crucial functions, including road illumination, signaling drivers intentions, and enhancing visibility when it is less light. Amongst the components of this system, the headlamps emerge as paramount. Positioned prominently at the vehicle's front end, headlamps bear the primary responsibility of casting light upon the road ahead. These essential fixtures typically come equipped with both high and low beams, tailored to suit various lighting conditions. In contemporary times, with many roadways illuminated by street lamps, drivers might inadvertently overlook the need to activate their headlamps. This oversight has the potential to precipitate automobile accidents or pose hazards to pedestrians, underscoring the critical importance of headlamp usage. Modern automatic headlamp systems have evolved to include sensors that intelligently activate and deactivate the headlights based on prevailing lighting conditions. However, it's worth noting that numerous countries have implemented regulations stipulating that headlights should be activated whenever a vehicle is in motion. This is a measure aimed at enhancing road safety. Despite the convenience of automatic systems, it's crucial to recognize that permanent usage of car's headlights can potentially accelerate their wear and lead to premature failure. The National Highway Traffic Safety Administration in the United States, as documented in [1], has recognized inadequate lighting as a contributing factor in roughly 2% of all vehicle accidents. This underscores the pivotal role of reliable lighting systems in vehicles. Additionally, statistics provided by The American Automobile Association (AAA), as referenced in [2], reveal that over half of all pedestrian fatalities occur between 6 p.m. and midnight. Consequently, it is imperative for vehicle owners to proactively inspect and maintain their lighting systems to guarantee optimal functionality. By doing so, they not only ensure their own safety but also maximize visibility while operating their vehicles, thereby contributing to overall road safety. In certain road situations, fellow road users may attempt to communicate with a driver to encourage them to activate their vehicle's lighting system. Typically, this communication is conveyed through brief or extended flashes of their vehicle's headlights. These headlight signals serve as a non-verbal means of conveying messages and warnings among drivers on the road. In contemporary road infrastructure systems, a proliferation of cameras is readily observable, predominantly employed for the purpose of monitoring road traffic. These cameras serve as a valuable resource for detecting anomalies in vehicle lighting systems. The scientific community has extensively explored techniques for the automatic detection of vehicles, primarily within the context of collision avoidance systems [3, 4], with cameras serving as a primary tool for this task. However, it's worth acknowledging that vehicle detection remains a challenging endeavor due to the extensive diversity in vehicle shapes, the often cluttered and complex road environments, and the ever-changing lighting conditions. To bolster the robustness of these detection systems, machine learning (ML) techniques

Aleksander Dawid, and Paweł Buchwald are with WSB University, Department of Transport and Computer Science, Poland. (e-mail: adawid@wsb.edu.pl, pbuchwald@wsb.edu.pl).

have come to the forefront [5–7]. Among these, convolutional neural networks (CNNs) have made substantial strides in advancing vehicle detection devices. These systems find prominent application in both autonomous vehicles and the realm of road safety. They represent a significant technological stride forward, enhancing the capability to detect and respond to vehicles in real-time, thereby contributing to safer and more efficient roadways. Our previous preliminary work on headlamps failure detection shows some successes in that matter [8].

In our current research project, we are focused on employing machine learning (ML) algorithms to ascertain the functioning status of a vehicle's headlights. Our approach involves utilizing a training dataset composed of images featuring 3D models of vehicles equipped with headlamps. These images serve as the foundation for training our model. The ultimate objective is to leverage Convolutional Neural Networks (CNNs) to perform real-time detection using a camera system, thereby determining whether the headlights of a given vehicle are in the activated (on) or deactivated (off) state. This application of ML and CNN technology holds the promise of providing a practical and automated solution for assessing the operational status of vehicle headlights, which has significant implications for road safety and overall vehicle maintenance.

II. Vehicle lighting safety parameters

All road-legal vehicles have headlights. It is required for them to emit a white (or selectively white) light. In the case of xenon lamps, they can emit colors that span from light blue to almost violet, as long as their brightness falls within acceptable parameters. New regulatory changes have introduced the use of separate daytime running lights, with LED lighting becoming increasingly common. LED lights not only exhibit greater durability but also consume significantly less energy. Over time, older or poorly maintained headlights can become discolored and develop a yellowish tint, impairing their ability to properly illuminate the road. Ensuring that the lights on our vehicles not only have the correct color but also match in color is crucial. This issue is typically less problematic with traditional and halogen bulbs since replacing one burnt-out bulb is straightforward. Xenon lamps, for example, present a unique challenge as their filaments degrade over time, making it extremely challenging to match the brightness of a new lamp with its predecessor. The comprehensive technical standards for operating motor vehicle headlights are outlined in the regulations of the Economic Commission for Europe of the United Nations [9]. Collecting a sufficient number of photographs depicting motor vehicles with both correct and incorrect lighting samples can be a complex undertaking. Creating a substantial dataset necessitates a significant amount of time and runs the risk of not capturing vehicles with specific lighting irregularities. Furthermore, assembling a database of vehicle photos may entail handling sensitive data, raising concerns related to data protection laws. To tackle these challenges and acquire suitable images of motor vehicles displaying diverse lighting attributes, it was decided to create a digital simulator based on the concept of a digital twin. This approach offers a controlled and efficient means to generate the needed data for research and analysis while circumventing the complexities associated with real-world data collection.

III. Unity3D engine

The "digital twin" concept in computer simulation involves generating a virtual representation of a tangible object or system to replicate its functionality and experiment with various operational scenarios or solutions. Digital twins are invaluable for emulating diverse operational states of real systems, testing innovative solutions, and forecasting future system conditions. This approach proves especially advantageous for intricate systems or facilities that are impractical or impossible to construct or study in the physical realm. Moreover, digital twins offer a safe and cost-effective alternative when real-life testing is prohibitively expensive or hazardous. In our study, we harnessed the Unity3D environment to construct a simulator capable of configuring the lighting parameters of a motor vehicle and environmental lighting conditions. The simulator affords the flexibility to customize camera settings for specific scenes, facilitating the generation of a wide array of motor vehicle images with varying lighting attributes and external circumstances. To execute the simulation, we employed digitized models to replicate the environment and the car itself. By integrating the camera component, we obtained graphical representations showcasing examples of vehicles with both correctly functioning lighting systems and those exhibiting lighting irregularities. Leveraging the digital twin concept, which faithfully mimics the operation of a car's lighting system, we could simulate the activation or deactivation of vehicle lighting and manipulate its parameters as needed. The simulator also streamlines the process of automatically generating vehicle images from different distances and angles. Additionally, it offers functionality for adjusting environmental lighting settings, enabling the recreation of scenarios such as driving in daylight, nighttime conditions, or challenging lighting conditions like driving against the sun. Moreover, the simulator permits the introduction of simulated failures and irregularities, including variations in parameters for both headlights, or values for lighting emission angles that would be deemed unacceptable under typical road conditions. By embracing this digital twin approach within our experiment, we've created a versatile and robust tool for simulating and assessing various aspects of vehicle lighting, enhancing our ability to research, analyze, and improve the performance and safety of automotive lighting systems. In our simulation scene setup, we employed standard assets such as “Terrain” and “Skymaps”. We integrated motor vehicle models, acquired through scanning and photogrammetric techniques [10], which were developed based on BluePrint images and sourced from freely available FBX files. However, the most critical aspect of scene preparation for creating target motor vehicles images was modelling of light conditions. Within the Unity3D environment, we had access to various types of lights, including point, spot, directional, and area lights, all of which were instrumental in crafting the desired scene (Fig. 1). Table 1 outlines the parameters of the Light object in the Unity3D environment, enabling us to fine-tune the applied light source to meet our requirements. It's important to note that the appearance of objects within the scene, as captured in the created images, depends not only on the external lighting parameters but also on the environmental lighting settings configured through the Lighting menu. Unity3D provides the capability to define the environmental material (skybox), composed of six photos reflecting the surroundings of the scene.

These environmental parameters also offer control over ambient light color, which influences the visual characteristics of all illuminated objects. By leveraging these capabilities, we could meticulously shape the lighting conditions and environmental factors to create a highly realistic and dynamic simulation environment for our research.

Fig. 1. The light object switched on Unity 3D scene

TABLE I
THE UNITY3D LIGHT SETTINGS

Setting name	Description
_type	The type of the light source.
_range	The range of the emitted light
_spot_angle	Angle of spot light.
_color	The color of emitted light.
_mode	Set the CPU/GPU load.
_intensity	brightness of the light source.

The quality of the created snapshots is also influenced by the camera object settings within the Unity3D environment, which can accurately replicate the camera's physical characteristics. By manipulating the position properties of both the camera object and the car model, it becomes feasible to simulate the car's movement and capture virtual photos from predefined distances. Furthermore, the "enabled" property of the light class object, employed to simulate the car's lights, enables the activation or deactivation of the car lights. Leveraging the capabilities offered by the Unity 3D environment, it becomes possible to generate a substantial number of sample images without the need for prolonged observations or waiting for a car with the desired lighting parameters to appear. This expedited and controlled process streamlines the generation of diverse training data for machine learning applications.

IV. PREPARING A CAR MODEL WITH LIGHTS

One of the methods of preparing a model of realistic cars is the use of photogrammetric methods, which allow you to reflect the grid of polygons representing the body of the car, as well as the textures responsible for the appearance of individual elements. The use of photogrammetry requires taking a series of photos of the object that show its appearance from all sides. Based on these photos, it is possible to create an object using software such as Meshroom. It is an open-source software that is based on the runtime environment for the applications of the AliceVision Photogrammetric Computer Vision framework. (https://github.com/alicevision/Meshroom). Unfortunately, the created model file has some inaccuracies and must be corrected manually, e.g. using the Blender environment. The appearance of the geometry of the car object obtained by the photogrammetric method is presented in Figure 2.

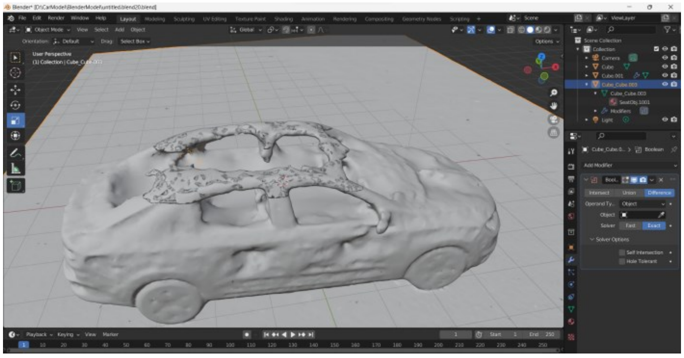


Fig. 2. Scanned 3D Car Object in Blender Environment

The quality of the generated model relies on numerous factors, such as the quality of the photographic images employed in the photogrammetry process, the positioning of the camera, and the ambient lighting conditions. In the process of preparing the model, it was noticed that shiny objects photographed in natural lighting (e.g. car body) in the target model are reproduced with inaccuracies that must be corrected manually. The Xiaomi Redmi Note 9 smartphone was employed for photography, boasting a camera with the following specifications: 48 MP (f/1.79, 1/2", 0.8µm, 79°). Additionally, the camera utilized a Time of Flight (ToF) sensor. This ToF component is capable of emitting infrared light and subsequently measuring the speed at which it bounces back from objects. The sensor comprises two essential parts: firstly, a diode responsible for emitting infrared radiation, and secondly, a specialized light-sensitive matrix. By precisely calculating the time it takes for the reflected light to return, the camera not only estimates the distance of individual objects from the sensor but also discerns their shapes with remarkable precision. It's worth noting that there are also devices available on the market equipped with LIDAR sensors. This device employs a technique that involves measuring distances by emitting laser light towards the target and subsequently gauging the reflected light with a sensor. Discrepancies in the time it takes for the laser beam to return and any alterations in its wavelength are harnessed to construct a three-dimensional model. Employing this methodology can yield superior outcomes when capturing the structure of the model [11]. The lack of availability of more advanced research equipment, and the relatively long time of manual correction of the model, meant that ready-made 3D models were also used to differentiate the cars used for testing. One of the problems that had to be solved to achieve the most realistic appearance of the model was to reflect the appearance of the lights, and to place in the hierarchical structure of the model appropriate light sources that would best reflect the realistic appearance of the real car. Prepared model files allow you to reflect their geometry and appearance through hierarchical mapping of individual elements of the car. A representation of the object hierarchy of an example car model is shown in Figure 3.

Fig. 3. Hierarchical car model with added lights

In the provided example, we employ a method that approximates light using two distinct sources of emission for each reflector. Within the Unity 3D environment, there exists the capability to manipulate light reflections through objects that interact with various emission sources, notably point lights and spotlights. Point lights are positioned at specific coordinates in space and emit light uniformly in all directions. The incident light's trajectory onto a surface corresponds to the line extending from the point of impact back to the center of the illuminated object. As you move farther away from the light source, the light's intensity diminishes, adhering to an inverse square law, where the light intensity decreases in proportion to the square of the distance from the source. On the other hand, spotlights are characterized by their distinct location and a defined range over which their light is cast. Spotlights are further constrained by an angle, giving rise to a cone-shaped illumination pattern. The central axis of this cone extends forward from the spotlight, with light intensity diminishing as one moves towards the outer edges of the spotlight's cone.

V. Neural network

In our research, we harnessed the power of YOLO for the detection of proper car lighting. YOLO, which stands for "You Only Look Once," is a real-time object detection algorithm that has garnered widespread recognition in computer vision. The core concept behind YOLO is to execute object detection in a single pass, eliminating the need for a separate region proposal step, a characteristic found in other algorithms dedicated to object detection. This distinctive feature makes YOLO significantly faster and more efficient compared to its counterparts. The YOLO algorithm operates by breaking down the input image into a grid of cells, where each cell is entrusted with the task of detecting objects within its designated region. Leveraging a convolutional neural network (CNN), the algorithm predicts the likelihood of an object's presence within each cell and the coordinates of the bounding box encapsulating the object. Furthermore, YOLO incorporates a crucial step known as Non-Maxima Suppression (NMS). This step serves the purpose of eliminating redundant bounding boxes that may correspond to the same object. This is essential because without this step, the same object can be detected multiple times in different cells. It's noteworthy that YOLOv7 represents the latest iteration of the YOLO algorithm [12], embodying advancements and refinements to enhance its performance and accuracy. YOLO's unique approach and real-time capabilities make it a formidable tool in various applications, including our endeavour to detect proper car lighting.

VI. Training

In our training model we have used two distinct classes that we've labeled as "lights_on" and "lights_off," signifying the respective states of motor vehicle headlamps. To train the network for the detection of these two states, we utilized a dataset comprising 872 snapshots per each class. This dataset was then partitioned into a training dataset, encompassing 748 snapshots, and a validation dataset comprising 124 images. This division equates to 86% of the snapshots for training and 14% for validation, maintaining a consistent format for each image. For uniformity, all images were set to a resolution of 640x640 pixels, employing the RGB color model, and saved in JPEG file format. Table 2 and Figure 2 provide examples of images sourced from our training set. It's worth noting that all these images were captured within our digital twin simulation hosted in the Unity3D engine. The vehicle photos adhered to the rules outlined in Table 2, covering scenarios with both states headlights. Additionally, snapshots of the digital motor vehicle simulation were taken under three distinct lighting conditions, named night, dawn, and midday, as depicted in Figure 2. We did not apply any further enhancements or alterations to these images using external image editing software. To facilitate YOLO network training, we labeled the images using a labelling program developed in Python, with OpenCV as a core component. In our training dataset, we utilized a standard tiny YOLO model, which is optimized for edge IoT computing devices such as the Jetson Nano, ensuring efficient computational performance. The training procedures were conducted using a GPU, specifically the NVIDIA GeForce GTX 960M with CUDA 5.0 capability, to expedite the training process and improve model performance.

TABLE II
THE UNITY3D CAMERA COORDINATES RELATIVE TO THE CAR MODEL IN (X,Y,Z) VECTOR FORMAT. ALL DISTANCES ARE USING METERS UNITS.

The names of the distance	The names of viewpoints			
	(left)	(right)	(center)	(top)
(near)	(-2,1.5,3)	(2,1.5,3)	(0,1.5,3)	(0,3,3)
(middle)	(-2,1.5,15)	(2,1.5,15)	(0,1.5,15)	(0,3,15)
(far)	(-2,1.5,30)	(2,1.5,30)	(0,1.5,30)	(0,3,30)

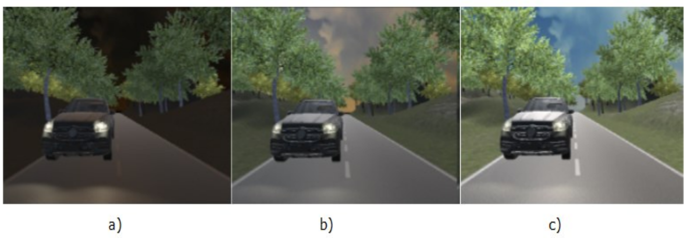
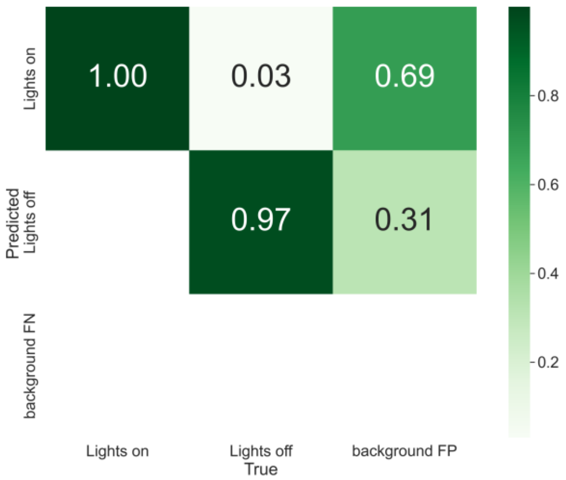


Fig. 4. The vehicle model captured in snapshots under three distinct time of day, like a) night, b) dawn, and c) midday

VII. Results

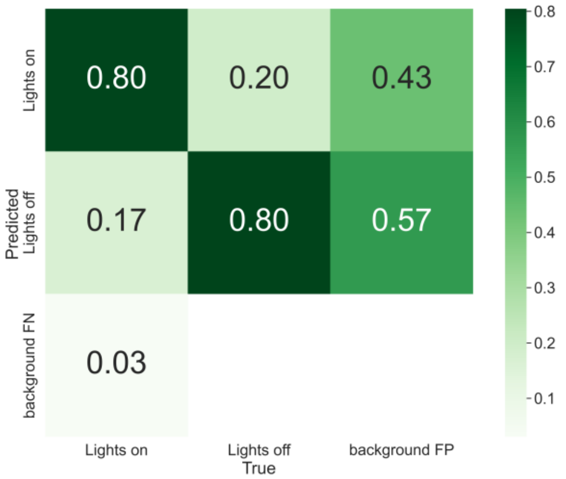
In our experiment, we have used a total of 872 images taken from the Unity3D scene as the training dataset for the YOLOv7 network. Our training process extended over 500 epochs. To assess the quality of our model's inference, we utilized a confusion matrix, as illustrated in Figure 5. This matrix is structured with actual class columns and predicted class rows, and its performance hinges on the values along its diagonal. Higher values on the diagonal signify better performance. Upon inspecting the matrix, we observe that our network demonstrates an impressive ability to identify 3D virtual motor vehicles with headlights on with 100% accuracy and cars with headlights off with 97% accuracy. The "Background FP" in Figure 5 refers to instances where the network incorrectly identifies background objects as belonging to one of the classes. However, a pertinent question arises: how well does our network perform when faced with real-life images of cars? Is the level of graphical detail from our 3D simulation sufficient to train the YOLOv7 network to recognize cars with both headlamps on and off in real-world photos? To investigate this, we compiled a set of 110 photographs featuring actual cars with their headlamps in both states. We utilized the testing tool integrated into the YOLO software to evaluate our trained model against these real images. Under real-world conditions, the confusion matrix exhibits



slightly different results, as depicted in Figure 6.

Fig. 5. The confusion matrix after the training using digital twin set

Fig. 6. Trained network confusion matrix for the images of the real-life photos



In this scenario, the inference quality for both classes is approximately 80%. Alongside less precise class recognition, the YOLO network also occasionally identifies non-existing objects. The term "Background FN" in the confusion matrix alludes to instances where the detector misses trash or non-trash objects, erroneously categorizing them as other background objects. These findings underscore the nuances and challenges associated with transitioning from computer-generated 3D graphics to real-world images. While our model excels in the former environment, it faces some hurdles when confronted with the inherent complexities of real-life photography. An illustration of the worst-predicted batch is presented in Figure 7.

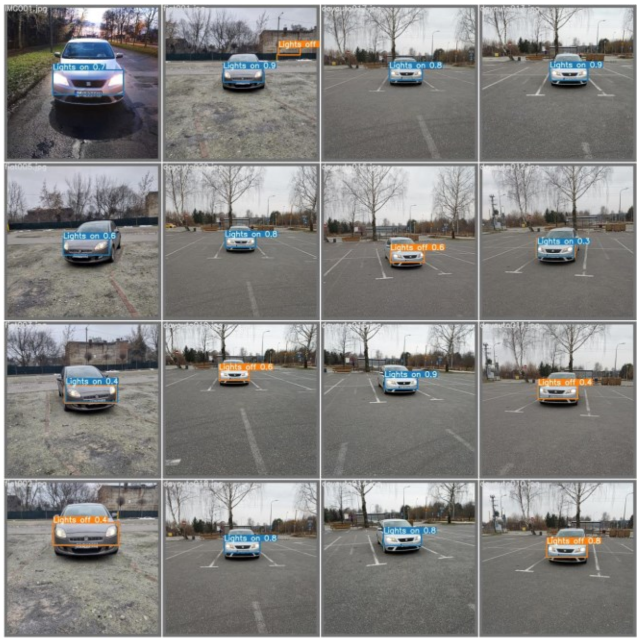


Fig. 7. The worse predicted batch example

Within this batch, we can observe 16 images, each accompanied by bounding boxes denoting the class labels and the associated probabilities. Notably, five of these images depict instances of false recognition of switched-on headlamps. Additionally, one image is erroneously classified as both switched-on and off headlamps simultaneously. This misclassification is attributable to the inherent disparities between real-world lighting conditions and the simplified lighting model employed in computer graphics. Consequently, we encounter instances where the network infers the presence of non-existent group of objects in the example set of images. For instance, the network occasionally identifies switched-off headlamps as part of the background structure or building in the image. These challenges underscore the intricacies of training a model to accurately recognize objects under real-world conditions, where lighting variations and complex environments can introduce complexities that differ from the controlled settings of computer-generated graphics. Precision (P) in recognition is quantified by considering both true positive (TP) and false positive (FP) cases, and it is calculated using the formula P = TP / (TP + FP).

This measure of precision is typically determined at a specified confidence level. To ascertain the confidence level at which the precision reaches 100%, we can refer to Figure 8, which displays the precision-confidence curve. As depicted, for all classes, the precision attains a perfect score of 100% at a confidence level of 86.4%. This finding informs us that, under the specified conditions and confidence threshold, the model achieves a flawless precision rate, correctly identifying objects without any false positives when the confidence score reaches 86.4%.

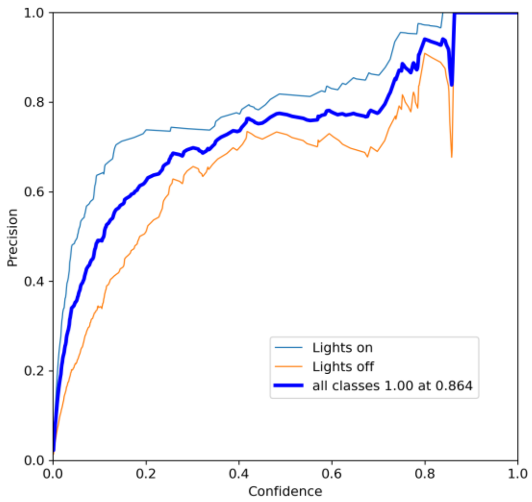


Fig. 8. The precision-confidence dependency curve

Another crucial parameter for evaluating the performance of machine learning models is recall, also known as the true positive rate or sensitivity. Recall (R) is calculated using the formula R = TP / (TP + FN), with FN representing the instances of false negatives. In the batch of pictures (Fig. 7), there is an example of a false negative (FN) case where the class was detected but inaccurately so. Recall can be understood as the ratio of true positive predictions to all actual positive instances within the dataset. Like precision, recall is evaluated with respect to confidence levels. Figure 9 presents the recall-confidence curve, offering insights into how recall varies at different confidence thresholds.

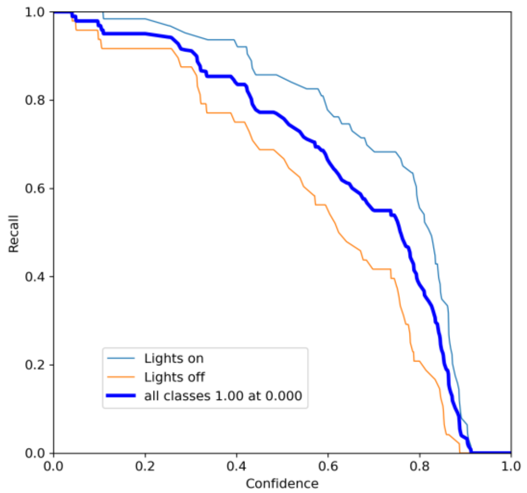


Fig. 9. The recall-confidence dependency curve

When the threshold is set at 0% confidence, we obtain all predictions from our machine learning model, irrespective of their correctness. At the 50% confidence threshold, we still retain 78.6% of predictions. Notably, the recall curve gradually diminishes to zero as we approach the 90% confidence threshold. This signifies that if we set the confidence threshold to 0.9 in our trained YOLO model, we would receive no predictions. For an ideal classifier, the recall curve should be constant with a value of 1.0 across all confidence levels, indicating perfect recall where all actual positive objects are correctly identified by the model. The optimal scenario for an algorithm entails achieving both high precision and high recall, effectively balancing accurate positive predictions and comprehensive coverage of actual positive instances.

This equilibrium ensures the algorithm's effectiveness in accurately identifying relevant items while minimizing false positives and false negatives. To comprehensively evaluate the quality of a classifier, the precision-recall (PR) curve, as illustrated in Figure 10, is commonly employed. A high area under the PR curve signifies a superior classifier. Upon reviewing this plot, it's evident that our classifier exhibits a stronger recognition performance for the headlamp on class compared to the headlamp off class. For all groups of objects, the ideal confidence threshold that achieves a balanced compromise between precision and recall is established at 83.3%. This insight informs us that, under these conditions, our classifier achieves a favorable equilibrium between making accurate positive predictions and ensuring comprehensive coverage of positive instances, marking it as a proficient performer.

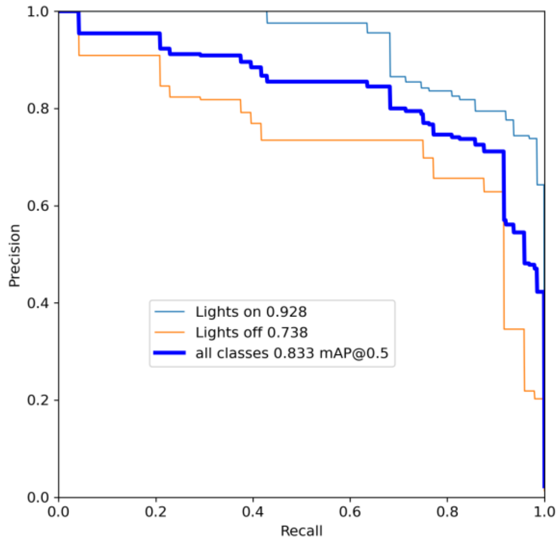


Fig. 10. The plot of precision-recall curve

When considering the practical application of our classifier, such as in a road lighting sensor system, it becomes crucial to determine the optimal confidence threshold. To achieve this, we create the F1 score curve, which represents a specialized instance of the broader Fβ function [13]. The F1 score curve, as depicted in Figure 11, allows us to visualize the trade-off between precision and recall. By conducting this analysis, we can pinpoint the optimal confidence threshold for our classifier, which is determined to be 0.417. At this threshold, the F1 score function attains its peak value, signifying that it achieves the most effective equilibrium between precision and recall. This threshold is particularly valuable when deploying the classifier in real-world scenarios, ensuring that it delivers a strong performance by optimizing both the accuracy of positive predictions and the coverage of positive instances.

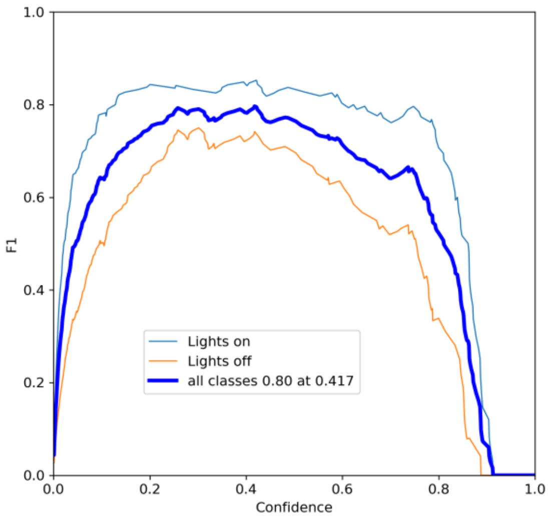


Fig. 11. The dependance of F1 score function on confidence

Enhancing the robustness of a network by generating additional spatially distorted images and incorporating them into the training set is a well-established technique in machine learning [14-19]. This method has demonstrated its effectiveness in improving the quality of trained ML models.
It helps the network generalize better and perform well even when presented with data that deviates from its training samples. Our work highlights how even a simple model of lighting in a 3D scene can yield impressive results in recognizing the status of vehicle headlights. This underscores the power of digital twins in the machine learning training process. Our research serves as a testament to the utility of digital twins in training machine learning models. It's evident that the costs associated with training neural networks using digital twins are substantially lower than using real-life images. Moreover, the process of learning can be more streamlined and automated when leveraging digital twins, offering a practical and cost-effective approach to model development and training.

VII. Conclusions

The obtained results show that digital twins built on the basis of basic 3D visualization environments can be used to generate training datasets. Even a simple digital twin built on polygons was able to generate a training dataset for a neural network. A YOLO v7 network trained on the generated images demonstrated the ability to recognize the state of vehicle headlights with an average probability of 80% during tests on 110 real vehicle images. Although we attempted to improve the model's performance by generating more training images and increasing the number of training epochs, we did not observe a significant improvement in recognition accuracy. We assume that the limitation lies in the quality of computer-generated graphics used in our simulations. For the implementation of the digital twin application, we used the basic version of Unity 3D. The use of the pro version, which offers greater graphical capabilities, especially in the area of lighting generation, could further enhance the realism of the training images. Lighting conditions in simulations differ from real scenarios with headlights, where light is usually reflected in concave mirrors and refracted through a glass headlight cover. More accurate modeling of all aspects of automotive headlights' operation could also yield better results in terms of realism. Using ray tracing instead of polygon and normal methods, which requires increased image preparation costs, could lead to significant improvement. This approach would involve modeling the interior of headlights with a focus on details for each vehicle. By applying this improved graphics and refining our digital twins, we expect to achieve better overall prediction accuracy in the future. However, the question of the extent of accuracy improvement requires further empirical research. A drawback of the approach proposed in our research was also the use of a limited number of car models in the preparation of the digital twin. Due to the specific appearance features of individual car models and even groups of models, it would be necessary to expand the database of modeled objects when implementing an improved version of the digital twin. The preparation of a vehicle model is a time-consuming process, even when using reverse engineering methods based on photogrammetry. Therefore, in building an extended database of car models, we plan to use modern graphic hardware to digitize objects using photogrammetric methods and utilize cloud computing services that offer 3D object digitization based on a series of real object photographs. The database of vehicle models can also be expanded by acquiring ready-made car models from commercial platforms. When analyzing the progress and potential of continuous development of 3D visualization engines and digital twins, it is important to note that achieving a higher degree of realism in generated training images is possible. Through achieving better realism and expanding and diversifying the training image database while tuning neural network parameters, we plan to achieve even better prediction results for real-world images.

References

[1] Home | NHTSA, https://www.nhtsa.gov/, last accessed 2023/01/17.

[2] AAA | American Automobile Association, https://www.ace.aaa.com/, last ac-cessed 2023/01/17.

[3] Z. Sun, G. Bebis, and R. Miller,"On-road vehicle detection: a review", IEEE Trans. Pattern Anal. Mach. Intell., vol 28, pp. 694–71, 2006. doi:10.1109/TPAMI.2006.104

[4] A. Mukhtar, L. Xia, and T. Boon Tang,"Vehicle Detection Techniques for Collision Avoidance Systems: A Review", IEEE Trans. Intell. Transp. Syst. Vol 16, pp. 2318–2338, 2015. doi:10.1109/TITS.2015.2409109

[5] A. Dawid, "PSR-based research of feature extraction from one-second EEG signals: a neural network study", SN Appl. Sci., vol. 1, article number 1536, 2019. doi:10.1007/s42452-019-1579-9

[6] A. Bhattacharyya, M. Sharma, R. B. Pachori, P. Sircar, and U. R. Acharya, "A novel approach for automated detection of focal EEG signals using empirical wavelet transform.", Neural Comput. Appl., vol. 29, pp. 47–57, 2018. doi:10.1007/s00521-016-2646-4

[7] E.S. Madhan, S. Neelakandan, R. Annamalai, R., “A Novel Approach for Vehicle Type Classification and Speed Prediction Using Deep Learning.”, J. Comput. Theor. Nanosci. vol. 17, pp. 2237–2242, 2020. doi:10.1166/jctn.2020.8877

[8] A. Dawid, P. Buchwald, B. Pawlak, "The Digital Twin to Train a Neural Network Detecting Headlamps Failure of Motor Vehicles", In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk (eds)

Dependable Computer Systems and Networks. DepCoS-RELCOMEX 2023. Lecture Notes in Networks and Systems. Springer, Cham., vol 737, pp. 29-38, 2023. doi:10.1007/978-3-031-37720-4_3

[9] Regulation No 48 of the Economic Commission for Europe of the United Nations (UNECE) — Uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signalling devices [2019/ 57].

[10] Y. Kim, H. Gwak, J. Oh, M. Kang., J. Kim, H. Kwon, and S. Kim, “CloudNet: A LiDAR-Based Face Anti-Spoofing Model That Is Robust Against Light Variation.” IEEE Access. vol. 11, pp. 16984–16993, 2023. doi:10.1109/ACCESS.2023.3242654.

[11] D. Palka, M. Sobota, P. Buchwald, "3D Object Digitization Devices in Manufacturing Engineering Applications and Services.", Multidiscip. Asp. Prod. Eng., vol. 3, pp. 450–463, 2020. doi:10.2478/mape-2020-0038

[12] C-Y Wang, A. Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", http://arxiv.org/abs/2207.02696, 2022. doi:10.48550/arXiv.2207.02696

[13] C. Goutte, E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation", In: Losada, D.E. and Fernández-Luna, J.M. (eds.) Advances in Information Retrieval.. Springer, Berlin, Heidelberg, pp. 345–359, (2005). doi:10.1007/978-3-540-31865-1_25

[14] K. Szyc, "An Impact of Data Augmentation Techniques on the Robustness of CNNs", In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, (eds.) “New Advances in Dependability of Networks and Systems.”, Springer International Publishing, Cham, pp. 331–339, 2022. doi:10.1007/978-3-031-06746-4_32

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", J. Big Data., vol 6, article number 60, 2019. doi:10.1186/s40537-019-0197-0

[16] A. Pilch and H. Maciejewski, "Labeling Quality Problem for Large-Scale Image Recognition", In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, (eds.) “New Advances in Dependability of Networks and Systems.”, Springer International Publishing, Cham, pp. 206–216, 2022. doi:10.1007/978-3-031-06746-4_20

[17] K. Szyc, "Determining the Minimal Number of Images Required to Effectively Train Convolutional Neural Networks". In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds) Theory and Applications of Dependable Computer Systems. DepCoS-RELCOMEX 2020. Advances in Intelligent Systems and Computing, Springer, Cham. vol 1173. pp 652–661, 2022, doi:10.1007/978-3-030-48256-5_64

[18] A. Rusiecki, "Standard Dropout as Remedy for Training Deep Neural Networks with Label Noise.", In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds) Theory and Applications of Dependable Computer Systems. DepCoS-RELCOMEX 2020. Advances in Intelligent Systems and Computing, Springer, Cham. Vol 1173, pp 534–542, 2020. doi:10.1007/978-3-030-48256-5_52

[19] J. Mazurkiewicz and A. Cybulska, (2020). Softcomputing Art Style Identification System. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds) Engineering in Dependability of Computer Systems and Networks. DepCoS-RELCOMEX 2019. Advances in Intelligent Systems and Computing, Springer, Cham., vol 987., pp 321–330, 2019, doi:10.1007/978-3-030-19501-4_32