RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v1i43.2820

YOLO architectures comparison for urban cyclist detection in an autonomous driving environment

Comparación de arquitecturas YOLO para la detección de ciclistas urbanos en un entorno de vehículos autónomos

Jorge David[1], **Mateo Quintero**[1], **Paula Ortíz**[1], **Luís Gómez**[1]
MSc. Mauricio Arias-Correa[1]

[1] ***Instituto Tecnológico Metropolitano,*** *Laboratorio de Visión Artificial y Fotónica, Facultad de Ingenierías, Medellín, Colombia.*

Correspondence: *mauricioarias@itm.edu.co*

Received: *October 15, 2023.* ***Accepted:*** *December 17, 2023.* ***Published:*** *March 13, 2024.*

How to cite: *J. A. David Rodríguez, P. A. Ortiz Santana, M. Quintero Restrepo, L. M. Gómez Meneses, and M. Arias-Correa, "YOLO architectures comparison for urban cyclist detection in an autonomous driving environment", RCTA, vol. 1, no. 43, pp. 64–72, Mar. 2024. Recuperado de* https://ojs.unipamplona.edu.co/index.php/rcta/article/view/2820

Abstract: The World Health Organization (WHO) states that over 55% of road traffic accident fatalities involve vulnerable road users, including 3% who are cyclists. While autonomous vehicles are capable of detecting objects and individuals on roadways, the detection of cyclists and the prediction of their movements continue to pose significant challenges. This paper presents results from the comparison of YOLOv7, YOLOv8, and YOLO-NAS architectures for urban cyclist detection. The methodology ensures that the detectors were trained under the same conditions. Subsequently, they were evaluated using 111 cyclist images with metrics such as IoU, precision, and recall. The results highlight advantages and disadvantages within each architecture, suggesting a priority for either inference time or the quality of cyclist detection in future work.

Keywords: Yolo, VRU, deep learning, cyclists' detection, autonomous vehicle.

Resumen: La OMS establece que más del 55% de las muertes en accidentes viales son de usuarios vulnerables, incluyendo un 3% de ciclistas. Aunque los vehículos autónomos pueden detectar objetos y personas en las carreteras, la detección de ciclistas y la predicción de sus movimientos siguen siendo desafíos significativos. Este artículo presenta resultados al comparar las arquitecturas YOLOv7, YOLOv8 y YOLO-NAS para detectar ciclistas urbanos. La metodología garantiza que los detectores se entrenaron bajo las mismas condiciones. Luego, se evaluaron con 111 imágenes de ciclistas utilizando métricas como IoU, precision y recall. Los resultados destacan ventajas y desventajas en cada arquitectura, lo que sugiere priorizar el tiempo de inferencia o la calidad de la detección de ciclistas en futuros trabajos.

Palabras clave: Yolo, VRU, deep learning, detección de ciclistas, vehículo autónomo.

1. INTRODUCTION

According to the 2018 World Health Organization (WHO) report, traffic injuries are the eighth leading cause of death for people of all ages (1.35 million in 2016), with more than half of them falling into the category of pedestrians, cyclists, and motorcyclists, referred to as vulnerable road users (VRUs). A Vulnerable Road User (VRU) is a road user with an increased risk of being injured or killed in traffic because they are not surrounded by protective cover that would significantly reduce the severity of an accident [1]. This definition encompasses all types of pedestrians, cyclists, motorcyclists, as well as individuals with disabilities or reduced mobility. Correctly identifying VRUs is one of the most challenging perception tasks for autonomous vehicles (AVs).

Due to the current and future surge in autonomous driving vehicles [2], it is crucial to develop effective vulnerable road user (VRU) detection systems for autonomous vehicles (AVs). Several studies have proposed solutions primarily for pedestrian [3], [4], [5], [6], [7], [8]. In contrast, the detection of cyclists has not received the same emphasis, probably because it has been identified as one of the most challenging perception tasks faced by an AV [9]. [10]. This complexity is related to factors such as the visual complexity of cyclists, the variety of possible orientations, tilts, and elevations, different aspect ratios, diverse appearances, occlusions, reflections, shadows, and backgrounds that can confuse detectors [11].

In order to enhance the integration of autonomous vehicles (AVs) into traffic in the coming years, it will be necessary to improve their vulnerable road user (VRU) detection capabilities, especially with regard to cyclists. In many countries, cyclists and vehicles share the road, and it would be unacceptable for injuries and fatalities among cyclists to increase due to collisions with AVs.

All of the above justifies the development of urban cyclist detection systems as vulnerable road users, to be implemented in autonomous [12].

Some authors have proposed cyclist detection systems based on machine learning techniques [13], but the progress of deep learning architectures has demonstrated the effectiveness of detectors based on YOLO architectures, surpassing the performance of other architectures [14].

This article proposes a comparison of cyclist detectors trained with the models of the three latest YOLO architectures: YOLOv7, YOLOv8, and YOLO-NAS. The proposed methodology includes the collection and curation of a dataset, the training of the architectures, and the use of appropriate metrics to evaluate their performance. The results and their analysis are presented in detail as input for future projects aiming to optimize and apply them in the field of autonomous driving, particularly in one of its crucial branches: human factors in autonomous driving.

2. METHODOLOGY

Figure 1 illustrates the flowchart representing the applied methodology for comparing the YOLO architectures. The process began with the creation of the dataset, which was used to train each of the three architectures in the subsequent stage: YOLOv7, YOLOv8, and YOLO-NAS. The prediction results were then used to calculate performance metrics such as Intersection over Union (IoU), Recall, Precision, and inference time for each architecture. With these results as the output of the process, an objective comparison is generated.

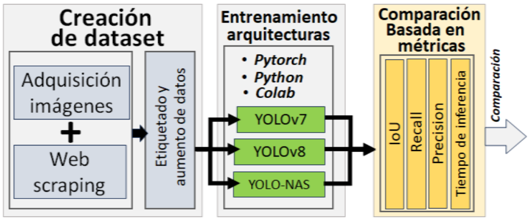

***Fig. 1**. Flowchart of the developed methodology during the process.*
***Source**: own elaboration.*

Because the training of YOLO detectors should be focused on the detection of urban cyclists, who would eventually be captured on the road by a moving autonomous vehicle (contributing to the improvement of collision avoidance strategies between such vehicles and cyclists as vulnerable road users), it was decided to create a dataset consisting of a combination of images acquired from the front windshield of a moving vehicle and publicly available images (without download and usage restrictions) of urban cyclists extracted from the web through a web scraping software called Bitzi, previously developed at the Metropolitan Technological Institute of Medellin and registered with the National Copyright Office in the year 2015 [15].

From the initially collected dataset, two major groups were selected. The first group consisted of images featuring a single cyclist, while the second group comprised images with multiple cyclists. Subsequently, data augmentation was performed on the selected RGB images to diversify the dataset and enhance the performance of the models to be trained. While it is common to use data augmentation for geometric transformations such as rotation, scale, and translation, adjustments can also be made to brightness, contrast, saturation, and hue, as well as applying focus changes, adding noise, among others [16], [17]. In our case, data augmentation involved changes in the brightness and contrast of the images to simulate different lighting conditions. Adjustments in saturation and hue were made to impact the appearance of colors. Random noise was added to simulate adverse weather conditions, and blur filters were applied to modify the appearance of cyclists in the images. The transformations applied to the dataset can be observed in Figure 2.

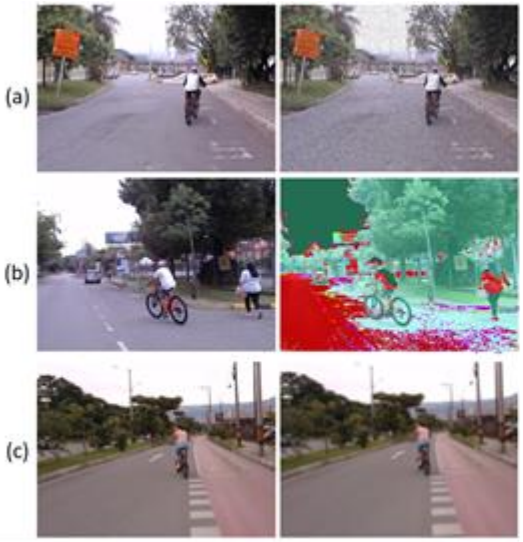

Fig. 2*. Data augmentation applied to the labeled dataset's RGB images. In the left column, original RGB images are presented, while the corresponding transformations are shown in the right column. In row (a), salt and pepper noise was applied to the image, in (b) an HSV color space filter, and in (c), a blur was applied.*

Source*: own elaboration*

The YOLO ("You Only Look Once") architectures have been chosen to train the detectors due to their ability to detect objects in a single pass through an image in real-time, making them particularly efficient in terms of processing speed. The YOLO approach is based on a single convolutional neural network (CNN) that takes an image as input and produces outputs in the form of bounding boxes that identify and locate objects in the image. This allows YOLO to simultaneously detect multiple objects in a single image [18].

The latest versions, YOLOv7, YOLOv8, and YOLO-NAS, represent the cutting edge of object detection, and their application in cyclist detection will help establish the advantages of these architectures over others reported in recent works for cyclist orientation detection, such as [13].

YOLOv7 [19] follows the same architecture as the original YOLO, consisting of three fundamental components: the backbone, the neck, and the head. Each of these components plays a crucial role in the image recognition and object detection process. The purpose behind the development of YOLOv7 was to design an architecture capable of predicting bounding boxes with higher accuracy than other models while maintaining the same speed in inference [20], [14]. To achieve this purpose, YOLOv7 introduced an improvement in the efficiency of the YOLO backbone, taking into account memory requirements and gradient descent backpropagation to enhance the learning capabilities of the network. It also includes re-parametrization planning, allowing internal variation of precision levels and inference speeds based on the application.

In 2023, YOLOv8 [21] emerged with the same characteristics as its predecessor YOLO architectures but incorporated an anchor-free model, enabling it to directly predict the center of an object. This mitigates challenges associated with regions around images, such as a lack of generalization and difficulty in handling irregularities, thereby reducing the number of prediction rectangles. Additionally, YOLOv8 improves the speed of the candidate detection generation process after the inference. YOLOv8 also incorporates an image augmentation strategy (Spatial Pyramid Pooling Feature), allowing the model to learn objects in new positions, partial occlusions, and against variations in surrounding pixels [22].

YOLO-NAS is an innovative object detection model that maximizes the latest advances in Deep Learning technology, overcoming limitations of previous YOLO versions. The term "NAS" refers to "Neural Architecture Search," which involves optimizing algorithms to automate the process of designing neural network architectures [23]. The primary goal of YOLO-NAS is to achieve an optimal balance between model accuracy, computational complexity, and model size. This architecture represents the forefront of real-time object detection [14].

The training of each of the three YOLO architectures was conducted on a Colab Pro computing machine provided by Google Colab. The dataset was distributed in three different proportions for training: 72.5% of the images (1689 images) were allocated to the training set, 20% (465 images) to the validation set, and another 7.5% (176 images) to the test set. The machine's specifications are presented in Table 1.

Table 1: *Characteristics of the assigned computing machine -Colab Pro-.*

Device	Characteristic
CPU	Intel Xeon E5-2686 v4 (2.4GHz, 12 cores)
GPU	NVIDIA Tesla T4 CUDA cores: 72 Memory: 16 GB GDDR6 Bandwidth: 320 GB/s
Memory	32 GB
Storage	500GB

The training characteristics were also the same, and expressed as hyperparameters, they can be seen in Table 2.

Tabla 2: *Hyperparameters assigned for the training of YOLO architectures.*

Hyperparameter	Value
warmup_initial_lr	1e-6
lr_warmup_epochs	2.0
initial_lr	0.001
max_epochs	30
batch size	16

All code implementations were developed using the Python programming language on Colab, with the PyTorch framework and loading pre-trained models (transfer learning) for the three YOLO architectures. In Table 3, the sources of the models are presented.

Table 3: *Origin of the pre-trained YOLO models.*

Modelo	Repositorio
YOLOv7	https://github.com/WongKinYiu/yolov7
YOLOv8	https://github.com/ultralytics/ultralytics
YOLO-NAS	https://github.com/Deci-AI/supergradients/blob/master/YOLONAS.md

Comparing network architectures, especially classifiers and object detectors in images, requires an evaluation through performance metrics. Metrics are calculated for a set of images based on the predictions made by the detectors (in this case, for cyclists). These predictions can be incorrect (False Positives FP, False Negatives FN) or correct (True Positives TP, True Negatives TN), and the ratios of these values are used to compute metrics such as Error, Accuracy, Precision, Recall, F1-score, and mAP. Depending on the specific requirements, some or all of these metrics may be utilized. In this study, the metrics IoU (Intersection over Union), precision, and recall, as well as the relationship between the latter two, will be employed to assess the performance of the trained architectures.

Precision is a metric used in object detection tasks to assess the accuracy of positive predictions made by the model. It helps determine the reliability of the model in identifying positive instances, minimizing false positives. A higher precision indicates a lower rate of falsely predicted positive instances [24].

$$Precision = \frac{TP}{TP + FP} \qquad (1)$$

Precision is calculated using Equation 1, where TP represents the number of correctly predicted positive instances, and FP represents the number of instances falsely predicted as positive.

Recall measures the proportion of actual positive instances correctly identified by the model. Recall quantifies the model's ability to correctly detect and capture object instances. A higher recall indicates a lower rate of missed detections [24].

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \qquad (2)$$

Recall is calculated using Equation 2, where TP represents the number of instances correctly predicted as positive, and FN represents the number of instances falsely predicted as negative.

In the following section, the obtained results will be presented, including details of the application of other metrics and the generation of curves, relevant to the comparison process.

3. RESULTS

In Table 4, you can observe the training times for each evaluated YOLO architecture. It is important to note that the training settings and processing machines used were the same, as specified in the methodology section.

Tabla 4: *training time.*

Model	Training time (h)
YOLOv7	3.5
YOLOv8	3.83
YOLO-NAS	2.33

Under the same training conditions, the YOLOv8 architecture required more time (3.83 hours), while the YOLO-NAS architecture was trained in just 60% of that time.

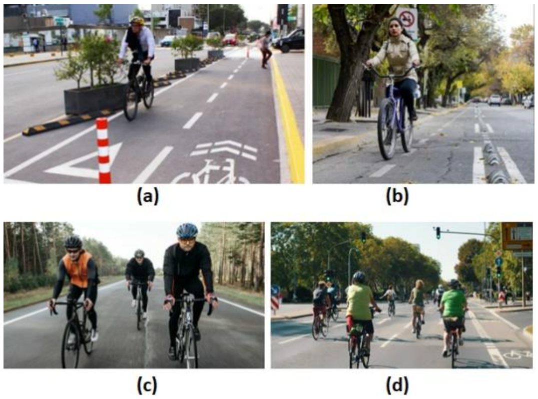

Fig. 3. *Images evaluated. In (a) and (b), images of a cyclist. In (c) and (d), images of multiple cyclists.*
Source*: own elaboration*

For a total of 111 images of cyclists (52 images of a single cyclist and 58 of multiple cyclists), ranging from images with a perfectly defined instance (a cyclist perfectly defined and without occlusions) to images with multiple instances with occlusions, as shown in Figure 3, values of predictions affecting the calculation of precision and recall metrics were obtained. These values include true positives (TP), false positives (FP), and false negatives (FN), and the results can be observed in Table 5. The table also includes the Intersection over Union (IoU) with a threshold value of 50% overlap. This indicates that a detection is considered effective if the Jaccard index (IoU) between the detection and the ground truth is greater than 0.5 [25], as illustrated in Figure 4.

Table 5*: Values of relevant predictions for each YOLO model.*

Arquitecture	TP	FP	FN	IoU
YOLOv7	182	100	91	0.4043
YOLOv8	240	42	49	0.6083
YOLO-NAS	244	38	26	0.7002

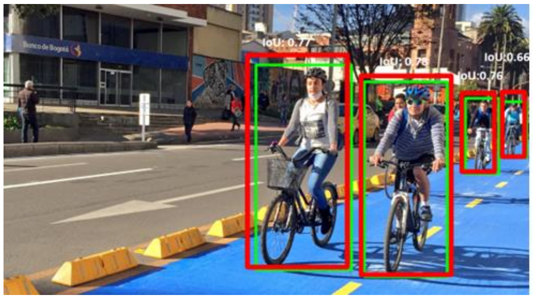

Fig. 4. *Calculation of the Intersection over Union (IoU) for an image from the evaluation group. The green box indicates the ground truth region, while the red box indicates the detection region.*
Source*: own elaboration*

From Table 5, it can be inferred that YOLO-NAS generates a bounding box closer to the ground truth region of the evaluated image group. The TP, FP, and FN predictions from Table 5 were used to calculate the performance metrics recall (proportion of true positives with respect to all cyclists present in the ground truth) and precision (proportion of true positives with respect to all detections made). The results can be observed in Table 6.

Table 6: *Performance evaluation results with precision and recall.*

Arquitecture	Precision	Recall
YOLOv7	0.6612	0.6695
YOLOv8	0.8854	0.8439
YOLO-NAS	0.8825	0.8836

Table 6 highlights the superior performance values for each metric. In terms of precision, YOLOv8 and YOLO-NAS outperform YOLOv7, indicating that the trained versions v8 and NAS are generating detections with fewer false positives and, therefore, are more accurate in detecting images where there is a higher probability of confusing cyclists with other road actors (such as motorcyclists). In the case of recall, it is the YOLO-NAS architecture that surpasses the others, indicating that this detector is more effective in finding all cyclists present in the image. This metric is of utmost relevance to our work because in an environment where autonomous vehicles are the predominant road actors, the best detector will be the one that detects most cyclists and reacts accordingly.

Figure 5 depicts the Precision-Recall (P-R) curve for the three trained YOLO detectors, using an IoU threshold of 0.5. It is clear from the curve that the detection algorithm achieving the best balance between recall (0.8836) and precision (0.8825) is YOLO NAS.

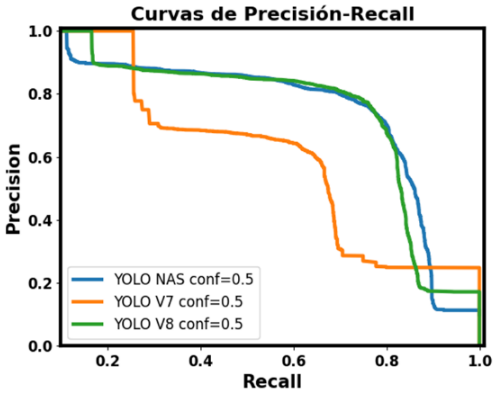

Fig. 5. *Precision-Recall curve for the evaluated YOLO detectors.*
Source*: own elaboration*

The superiority of YOLO-NAS over the other detectors is evident. On the other hand, YOLOv7 generates a significant number of false positives compared to the other two architectures.

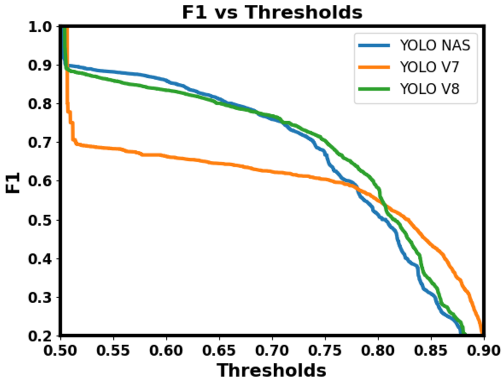

Fig. 6. *Behavior of the detectors when the threshold is varied (F1 vs. Thresholds curve).*
Source*: own elaboration*

Despite the good result in the precision metric for the YOLOv8 architecture detector, the number of detected cyclists is lower than the quantity detected by YOLO-NAS.

The system's behavior when varying the threshold to determine if a detection is entirely correct or not can be observed in Figure 6.

The F1 vs. Threshold curve shows how the F1 metric varies as the model's decision threshold is adjusted. As the threshold increases, the model's tolerance decreases, and it can be observed that the YOLOv7 architecture performs much worse than the other architectures. The IoU threshold at 0.5 appears to be the most suitable for deciding if a detection is correct or not.

The inference times for each architecture based on the number of cyclist instances in the images can be seen in Table 7. Due to the complexity of the models of YOLOv8 and YOLO-NAS compared to YOLOv7, the time it takes for YOLOv7 to detect cyclists is lower. The time required for YOLO-NAS to detect multiple cyclists in an image is up to 5 times higher than the time required by YOLOv7.

Table 7: *Inference time (Ti) for each architecture.*

Arquitecture	Ti for 1 cyclist (ms)	Ti multiple cyclist (ms)
YOLOv7	8.3	15.5
YOLOv8	9.4	84.3
YOLO-NAS	29.2412	79.1275

4. CONCLUSIONS AND FUTURE WORK

In this article, a performance evaluation has been conducted on the YOLO architecture versions known as YOLOv7, YOLOv8, and YOLO-NAS when applied to the detection of urban cyclists. Our goal has been to assess their effectiveness in addressing challenges associated with detecting cyclists on urban roads in an environment where, eventually, autonomous vehicles would be the predominant road actors. To achieve this, we trained the YOLO architectures with a combined dataset, consisting of our own images and images obtained through web scraping, all manually labeled.

We implemented a robust training and testing strategy. The strategy included training with common features, such as the database, the number of epochs, hyperparameters, and test images.

The evaluation of the YOLO architectures utilized the Recall and Precision metrics based on the data of True Positives (TP), False Positives (FP), and False Negatives (FN), with an IoU of 50% in the detection of cyclists across 111 images with diverse characteristics and instances of cyclists.

This methodology allowed us to establish a solid foundation for comparing the performance of the YOLO architectures in question when applied to the detection of cyclists. It enabled us to draw meaningful conclusions for the next stage in the development of a larger-scale project that incorporates both cyclists and autonomous vehicles.

The conducted experiments allowed us to establish that the YOLO-NAS architecture demonstrated superior performance in both recall and IoU for the region of cyclist detection in the images. YOLOv8 showed better results in precision but lagged behind YOLO-NAS in recall. On the other hand, YOLOv7 performed below the other two architectures in both metrics, IoU, prediction values, and even in training time.

Regarding the visual results (purely qualitative), the architectures exhibited limited performance when scenes were distant or had poor lighting conditions. Additionally, in some cases, cyclists were not detected when partially obscured by different objects or other cyclists. In scenes with favorable lighting conditions, unobstructed cyclists (or low occlusion), as well as a common aspect ratio between cyclist and bicycle, YOLOv8 and YOLO-NAS outperformed YOLOv7, successfully detecting all instances of cyclists.

In an environment where autonomous vehicles are the predominant road actors in urban areas, a cyclist detector should meet the following basic requirements: detect all cyclists and do so in the shortest possible time. In that line of thought, YOLO-NAS is a detector with excellent performance concerning metrics, but the time it requires to detect a cyclist (or multiple cyclists) in an image is up to 5 times greater than the fastest detector (YOLOv7). Thus, under the training and evaluation conditions conducted in this study, YOLOv8 emerges as the architecture with the best performance for detecting cyclists on urban roads from an autonomous vehicle.

In summary, when evaluating different architectures for cyclist detection on the road, it is concluded that there is no one-size-fits-all best model. The choice of the model will depend on the specific needs of the application. For future work, it would be beneficial to expand the analysis with other datasets containing greater diversity and a larger number of images, under more comprehensive training conditions, and with faster processing machines. Future work will enable fine-tuning of the presented architectures through hyperparameter optimization.

In general, this study serves as a foundation for future research efforts aimed at enhancing the effectiveness of cyclist detection systems from autonomous vehicles and minimizing the indicators of injured or deceased cyclists on the road due to collisions with motorized vehicles, whether autonomous or not.

REFERENCES

[1] Flohr, F. B. (2018). Vulnerable Road User Detection and Orientation Estimation for Context-Aware Automated Driving.

[2] Thrun, S. (2010). Toward robotic cars. Communications of the ACM, 53(4), 99–106. https://doi.org/10.1145/1721654.1721679

[3] Alhajyaseen, W. K. M., Asano, M., & Nakamura, H. (2012). Estimation of left-turning vehicle maneuvers for the assessment of pedestrian safety at intersections. IATSS Research, 36(1), 66–74. https://doi.org/10.1016/j.iatssr.2012.03.002

[4] Brohm, T., Haupt, K., & Thiel, R. (2019). Pedestrian Intention and Gesture Classification Using Neural Networks. ATZ Worldwide, 121(4), 26–31. https://doi.org/10.1007/s38311-019-0006-6

[5] Chen, Y. Y., Jhong, S. Y., Li, G. Y., & Chen, P. H. (2019). Thermal-Based Pedestrian Detection Using Faster R-CNN and Region Decomposition Branch. Proceedings - 2019 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2019, 2019–2020. https://doi.org/10.1109/ISPACS48206.2019.8986298

[6] Heo, D., Nam, J. Y., & Ko, B. C. (2019). Estimation of pedestrian pose orientation using soft target training based on teacher-student framework. Sensors (Switzerland), 19(5). https://doi.org/10.3390/s19051147

[7] Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian detection based on yolo network model. Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018, 1547–1551. https://doi.org/10.1109/ICMA.2018.8484698

[8] Murphey, Y. L., Liu, C., Tayyab, M., & Narayan, D. (2018). Accurate pedestrian path prediction using neural networks. 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings, 2018-

Janua, 1–7. https://doi.org/10.1109/SSCI.2017.8285398

[9] Fairley, P. (2017). Self-driving cars have a bicycle problem [News]. IEEE Spectrum, 54(3), 12–13.

[10] Mannion, P. (2019). Vulnerable road user detection: state-of-the-art and open challenges. 1–5. http://arxiv.org/abs/1902.03601

[11] Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., & Gavrila, D. M. (2016). A new benchmark for vision-based cyclist detection. IEEE Intelligent Vehicles Symposium, Proceedings, 2016-Augus(Iv), 1028–1033. https://doi.org/10.1109/IVS.2016.7535515

[12] Kress, V., Jung, J., Zernetsch, S., Doll, K., & Sick, B. (2019). Pose Based Start Intention Detection of Cyclists. 2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, 2381–2386. https://doi.org/10.1109/ITSC.2019.8917215

[13] Garcia-Venegas, M., Mercado-Ravell, D. A., Pinedo-Sanchez, L. A., & Carballo-Monsivais, C. A. (2021). On the safety of vulnerable road users by cyclist detection and tracking. Machine Vision and Applications, 32(5), 109.

[14] Casas, E., Ramos, L., Bendek, E., & Rivas-Echeverría, F. (2023). Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection. IEEE Access.

[15] BITZI, software image scraping. (2017). Instituto Tecnológico Metropolitano. Medellín. https://doi.org/10.1109/mspec.2017.7864743

[16] Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017, October). Research on data augmentation for image classification based on convolution neural networks. In 2017 Chinese automation congress (CAC) (pp. 4165-4170). IEEE.

[17] Wickramanayake, S., Hsu, W., & Lee, M. L. (2021). Explanation-based data augmentation for image classification. Advances in Neural Information Processing Systems, 34, 20929-20940.

[18] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).Clymer, J. R. (1992). "Discrete Event Fuzzy Airport Control". IEEE Trans. On Systems, Man, and Cybernetics, Vol. 22, No. 2.

[19] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7464-7475).

[20] Yasir, M., Zhan, L., Liu, S., Wan, J., Hossain, M. S., Isiacik Colak, A. T., ... & Yang, Q. (2023). Instance segmentation ship detection based on improved Yolov7 using complex background SAR images. Frontiers in Marine Science, 10, 1113669.

[21] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics. URL: https://github.com/ultralytics/ultralytics.

[22] Xia, K., Lv, Z., Zhou, C., Gu, G., Zhao, Z., Liu, K., & Li, Z. (2023). Mixed Receptive Fields Augmented YOLO with Multi-Path Spatial Pyramid Pooling for Steel Surface Defect Detection. Sensors, 23(11), 5114.

[23] Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Tan, K. C. (2021). A survey on evolutionary neural architecture search. IEEE transactions on neural networks and learning systems.

[24] Padilla, R., Netto, S. L., & Da Silva, E. A. (2020, July). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE.

[25] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[26] Li, X., Li, L., Flohr, F., Wang, J., Xiong, H., Bernhard, M., Pan, S., Gavrila, D. M., & Li, K. (2017). A unified framework for concurrent pedestrian and cyclist detection. IEEE Transactions on Intelligent Transportation Systems, 18(2), 269–281. https://doi.org/10.1109/TITS.2016.2567418

[27] Lin, Y., Wang, P., & Ma, M. (2017). Intelligent Transportation System(ITS): Concept, Challenge and Opportunity. Proceedings - 3rd IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2017, 3rd IEEE International Conference on High Performance and Smart Computing, HPSC 2017 and 2nd IEEE International Conference on Intelligent Data and Securit, 167–172. https://doi.org/10.1109/BigDataSecurity.2017.50

[28] World Health Organization. (2018). Global status report on road safety 2018. In Global status report on road safety 2018: Summary (No. WHO/NMH/NVI/18.20).