UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Visión Electrónica
https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRÓNICA

A RESEARCH VISION

Efficient live video traffic detection with YOLO convolutional neural architectures

Detección eficiente de tráfico de video en vivo con arquitecturas neuronales convolucionales YOLO

Ricardo Alfonso Gómez Suárez[1], Julieth Alejandra Sandoval Estupiñán[2], Paola Andrea Mateus Abaunza[3]

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:
Enviado: 28/06/2025
Recibido: 01/07/2025
Aceptado: 17/11/2025

Keywords:
BNN
CNN
Computer Vision
OpenCV
Python
YOLO

Palabras clave:
BNN
CNN
Visión por Computador
OpenCV
Python
YOLO

ABSTRACT

Advancements in computer vision have revolutionized machines' ability to recognize and classify objects in real-time. This study explored the convolutional neural network architectures YOLOv5 and YOLOv8 for precise and efficient real-time object identification. A custom dataset was created, and Python and OpenCV were utilized for the annotation and training processes of the models. The results demonstrated the high accuracy and detection speed of both models, even in live video environments.

RESUMEN

Los avances en el campo de la visión artificial han revolucionado la capacidad de las máquinas para llevar a cabo la identificación y clasificación de objetos en tiempo real. En este estudio, se examinaron las arquitecturas de redes neuronales convolucionales YOLOv5 y YOLOv8 con el propósito de lograr una identificación precisa y eficiente de objetos en escenarios de tiempo real. Para este propósito, se creó un conjunto de datos personalizado, y se emplearon Python y OpenCV para llevar a cabo la anotación y el entrenamiento de los modelos. Los resultados evidenciaron la destacada precisión y velocidad de detección de ambos modelos, inclusive en situaciones de transmisión de vídeo en directo.

1. BSc. In Electronic Engineering, Specialist in Engineering Management Systems, from the Pontificia Universidad Javeriana. Mobile Communications Specialist, Master in Information and Communications Sciences, student Doctorate in Engineering from the Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Current position: Professor Researcher, ECCI University, Bogotá, Colombia. Email: rgomezs@ecci.edu.co ORCID: ORCID:https://orcid.org/0000-0002-1142-9406

2. BSc. In industrial electronics technology, student of Electronic Engineering at the ECCI University, Bogotá, Colombia. Email: juliethа.sandovale@ecci.edu.co ORCID: https://orcid.org/0009-0005-0022-1668

3. BSc. In Electronic Engineering, MSc. in Electronic Engineering, Pontificia Universidad Javeriana, Bogota, Colombia. Current Position: Professor Researcher, Universidad Nacional Abierta y a Distancia, UNAD, Bogotá, Colombia. Email: paola.mateus@unad.edu.co ORCID: https://orcid.org/0000-0002-0638-1815

Cite this article as: R. A. Gómez-Suárez, J. A. Sandoval-Estupiñán, P. A. Mateus-Abaunza, "Efficient live video traffic detection with YOLO convolutional neural architectures", Visión Electrónica, vol. 19, no. 2, 2025.

1. Introduction

In this study, comprehensive analyses of two state-of-the-art architectures, YOLOv5 and YOLOv8, were conducted, with the objective of achieving real-time road actor identification. This approach was based on the use of custom datasets, supported by the powerful computational tools of Python and OpenCV [1]. Both models exhibited exceptional performance in detecting objects, even in challenging environments, positioning them as highly effective tools for applications requiring instantaneous responses [2]. These achievements not only contribute to practical engineering knowledge but also unlock a series of new possibilities in various industries, driving technological advances such as their application in autonomous trucks and industrial automation, which, in turn, raises efficiency and safety standards [3].

2. Conceptual Framework

2.1. Machine Learning and Neural Networks

The field of machine learning focuses on developing algorithms and models that make data-driven decisions, enabling machines to learn and improve their performance through experience. In the field of real-time object detection, machine learning plays an essential role in training models to recognize and categorize objects using datasets that have been previously labeled [4].

Neural networks, inspired by the functioning of the human brain, are widely used in machine learning. Among them, convolutional neural networks (CNN) are especially effective in tasks related to computer vision, such as the detection and classification of objects in images and videos [5].

Two notable examples of CNN architectures for real-time object detection are YOLOv5 and YOLOv8. These models acquire visual knowledge to quickly and accurately identify and locate objects [6]. The conceptual framework of this study involves the application of machine learning and CNNs to address the challenges associated with real-time object detection [7]. The combination of these technologies enables fast and precise object identification in real-time applications, with significant implications in areas such as autonomous driving, security, and surveillance [8].

2.2. Governance and Emerging Technologies

A point of paramount importance that demands special attention is the potential impact on equity and justice. It is essential to ensure that real-time road actor identification systems do not inadvertently perpetuate prejudice or discrimination against certain groups of road users [9]. Unintended biases present in these systems could have severe consequences, generating unequal treatment and unfair results.

Furthermore, although technological advances enable massive data collection, it is of vital importance to prioritize the protection of personal data and comply with privacy regulations when collecting and processing visual information. Achieving an adequate balance between using gathered data to improve road safety and guaranteeing people's privacy rights constitutes a complex task that requires extremely cautious policy formulation and execution [9].

2.3. Environment and Mobility

The application of YOLOv5 and YOLOv8 in real-time object detection has the potential to improve both road safety and traffic management. However, it is essential to evaluate and address the environmental and mobility implications of this technology [10]. The use of cameras and sensors can affect biodiversity, so it is essential to minimize adverse effects on local environments. In this sense, it is a priority to focus on energy efficiency and the use of renewable energy sources [10].

Regarding mobility, real-time object detection provides valuable information to improve transportation planning and promote sustainable mobility options [11]. Optimizing traffic lights and improving public transport systems can help reduce traffic congestion and greenhouse gas emissions generated by private trucks [11]. In addition, this technology promotes sustainable transport modes, such as carpooling and active mobility, by offering accurate data on available services, resulting in decreased emissions and more efficient use of resources [10].

To guarantee a responsible use of this technology, policies must address ethical, data protection, and environmental issues throughout the entire life cycle of these systems [10]. Collaboration between the public, private, and academic sectors becomes an essential component to find effective solutions that drive sustainability and responsible mobility [11]. In summary, responsible implementation and the adoption of solid policies play a crucial role in maximizing the benefits of this technology while minimizing negative impacts on the environment and urban mobility.

2.4. Education and Training in Object Detection Technologies

Education and training play an essential role in the successful implementation of object detection technologies, such as YOLOv5 and YOLOv8. It is crucial to establish a solid foundation in the fundamental concepts of machine learning and convolutional neural networks [12-13].

Acquiring specific knowledge about the characteristics, advantages, limitations, and appropriate configurations of these models is essential for a variety of applications [13]. In the training process, it is necessary to incorporate practical examples and use cases that allow effective application in real-world situations [14].

Likewise, it is equally important to foster awareness of the ethical challenges linked to privacy, discrimination, and biases present in machine learning models [14]. Participants must be prepared to responsibly address these ethical and legal issues in practical applications. Ultimately, education and training empower students and professionals to employ object detection technologies efficiently and ethically [14].

2.5. Bogotá as an Innovative City in the Identification of Road Actors

Bogotá, the capital of Colombia, has become a pioneer in the adoption of road actor identification technologies, including YOLOv5 and YOLOv8. By implementing advanced object recognition systems, the city has significantly improved road safety, transport efficiency, and traffic management [15]. Intelligent cameras and sensors strategically placed throughout Bogotá allow efficient traffic monitoring and real-time detection of trucks, pedestrians, and bicycles, providing crucial data for informed decision-making in transport and road safety [16].

Furthermore, Bogotá has been at the forefront of implementing sustainable mobility policies based on road actor identification. Leveraging information from object recognition systems, the city has implemented measures to improve traffic flow, optimize traffic light timings, and promote the use of ecological transport options such as public transport and active mobility. These initiatives have effectively reduced congestion, decreased polluting emissions, and fostered a more efficient and sustainable transport network [17-18].

3. Model Description

3.1. Traffic Flow Analysis

The object detection approach, based on the YOLOv5 and YOLOv8 architectures, is dedicated to analyzing vehicle flow in real-time, making use of computer vision and machine learning techniques. Its main task consists of identifying, classifying, and tracking various types of moving trucks on the roads [19]. The study of traffic flow is of crucial importance in understanding road congestion, optimizing transport management, and improving urban planning. The efficiency with which this model addresses these tasks is based on its deep convolutional neural network architecture [20].

Initially, a detection layer is responsible for recognizing trucks in images or image sequences by using feature maps to identify distinctive aspects of the trucks, such as their shapes, edges, and textures. Subsequently, object detection algorithms locate and mark each vehicle in the scene [20]. Once detected, a classification layer assigns labels to each vehicle, differentiating between cars, trucks, motorcycles, bicycles, and others, which provides detailed information about traffic flow and facilitates more specific analyses [21].

In addition, the model incorporates a tracking layer that monitors the trajectory of trucks over time, allowing the analysis of vehicle flow behavior, including aspects such as speed, direction, movement patterns, and accident rates, Figure 1. This precise tracking ensures that the traffic flow count remains updated and reliable in a given area [22-23].

To improve the precision and robustness of the model, a training process is carried out using labeled datasets containing images of trucks and annotations indicating their location and class. Through machine learning algorithms, the weights of the convolutional neural network are optimized to achieve more precise detections and classifications under a variety of lighting conditions and environments [2].

Figure 1: Road accidents in Colombia during 2020 and 2021.

Actor vial

Condición	2020	2021	Variación	% V
Usuario de moto	2908	4312	1404	48,28 %
Peatón	1128	1566	438	38,83 %
Usuario de vehículo	624	852	228	36,54 %
Usuario de bicicleta	433	471	38	8,78 %
Usuario otros	22	13	-9	-40,91 %
Sin Información	343	56	-287	-83,67 %
Total	**5458**	**7270**	**1812**	**33,20 %**

Source: Own work based on National Road Safety Agency data [11].

3.2. YOLO Implementation in Python: Code and Model Application

The implementation of the YOLO (You Only Look Once) model in Python has stood out as a highly effective tool for the precise and efficient identification of road actors in real-time. YOLO is based on a convolutional neural network architecture that enables the simultaneous detection and classification of multiple objects in an image [1].

To use the YOLO model in Python, it is necessary to install the required libraries and dependencies, such as OpenCV and PyTorch. Once these are configured, the following steps can be followed for its implementation:

1. **Download and configure the model:** First, proceed to download and configure the pre-trained YOLO model. Versions such as YOLOv3, YOLOv4, YOLOv5, or YOLOv8 are available in public access repositories. Once the model is downloaded, it is loaded using the PyTorch library [1].

2. **Data preparation:** Before carrying out the detection of road actors, a preparation of the input data is required. This involves obtaining images or accessing a real-time video stream. The images or video frames must undergo a resizing and normalization process, ensuring they adapt to the requirements of the YOLO model [1-27], Figure 2.

Figure 2: Fragment of the image dataset used for YOLO training.

Source: Authors [27].

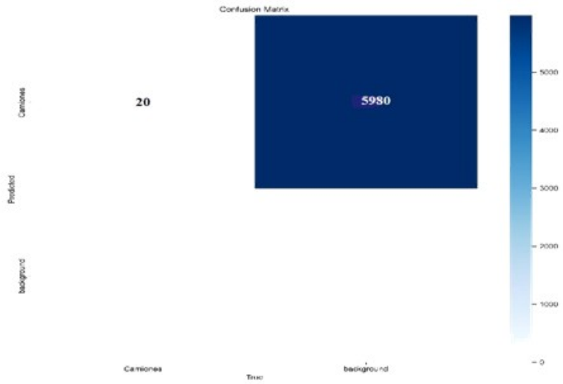
3. **Road actor classification:** Once the input data is ready, it is possible to perform road actor classification using the YOLO model. This implies passing the images or video frames through the model and obtaining the corresponding detections. Each detection will provide information such as bounding box coordinates, class label, and confidence associated with that detection [1].
4. **Detection refinement after processing:** Once detections are obtained, a post-processing stage can be carried out to clean and refine the results. This process may involve removing low-confidence detections, non-maximum suppression to eliminate redundant detections, and assigning labels to detections following a classification scheme [1].
5. **Visualization of results or run set:** Finally, it is possible to visualize the results of road actor detection on the road. This could include superimposing bounding boxes and class labels on the images or videos, allowing a visual representation of the detections made by the YOLO model [1].

3.3. Confusion Matrix and Performance Metrics

A confusion matrix is a table that presents the actual classes of objects and the classes predicted by the model, Figure 3. In the context of object detection, the rows of the matrix represent the actual classes of the objects, while the columns represent the classes predicted by the model [28]. The confusion matrix is divided into four main quadrants:

- **True Positives (TP):** Indicates the quantity of objects that the model has correctly detected and classified as positive.
- **True Negatives (TN):** Indicates the quantity of objects that the model has correctly classified as negative.
- **False Positives (FP):** Indicates the quantity of objects that the model has incorrectly classified as positive.
- **False Negatives (FN):** Indicates the quantity of objects that the model has incorrectly classified as negative.

Figure 3: Confusion matrix evaluating the truck image training model.

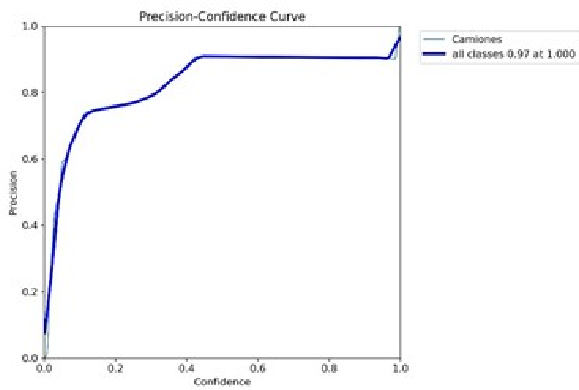


Source: Authors [28].

This curve, Figure 4, is generated by plotting precision as a function of various confidence thresholds. It shows how precision varies with different confidence values, allowing researchers to examine the trade-off between precision and the total number of detections. A higher confidence threshold leads to fewer detections, but they are more precise.

The precision-confidence curve is fundamental for adjusting model parameters and achieving the desired balance between precision and detection rate within the context of YOLO-based object detection [29].

Figure 4: Confidence-precision curve for the truck image training process.



Source: Authors [29].

The file "$results.png$" visually represents the training process of the YOLOv8 model. It contains different types of information, including:

1. **Loss charts:** These graphs show how the loss function evolves over different epochs or iterations during trai-

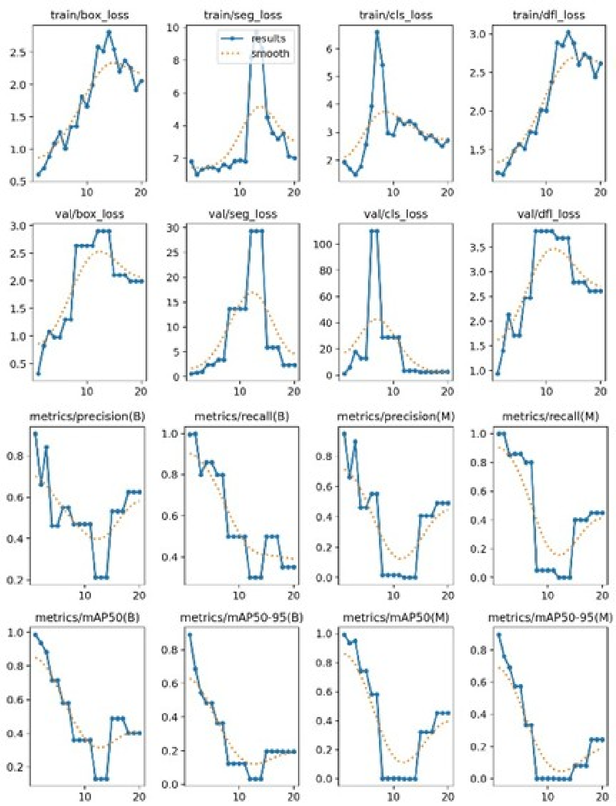
ning. Its purpose is to evaluate the model's learning progress and its improvement over time.

2. **Precision curves:** These curves illustrate how the model's accuracy increases during the training process. Precision curves can be organized by class or include metrics such as average precision or mean IoU for detections.

3. **Results visualization:** In this section, visual examples of training images with detections made by the model at different stages of the training process are presented. This provides a visual evaluation of the improvement in the model's object detection capability.

The fundamental purpose of ”$results.png$” is to provide researchers and developers with a visual representation that summarizes and describes the progress and performance of the model's training, Figure 5. This facilitates monitoring, analysis, and adjustments needed to improve the model and its configuration [30].

Figure 5: Output Results.png"for truck image training.



Source: Authors [30].

Training images generated within the context of YOLO represent the results of object detection during the model's training phase. After each iteration, the model makes predictions and produces these images showing detected objects alongside corresponding bounding boxes and labels. These images play a critical visual role in evaluating the model's performance as it learns to detect objects, adjusting bounding boxes, and assigning appropriate labels.

It is crucial to highlight that these training images are specific to the training stage and are used to monitor and analyze the model's intermediate performance. Once the model is trained, it is evaluated using other images in real-world scenarios [31], Figure 6.

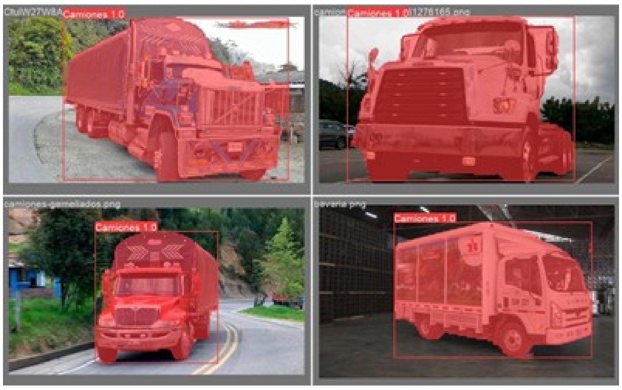
Figure 6: Output "Trainïmages for truck training.

Source: Authors [31].

In the YOLO training process, validation images are used to evaluate the model's performance on new and previously unseen data. These images are different from the training dataset and play a fundamental role in estimating the model's capability to generalize its knowledge to unfamiliar situations.

To perform this evaluation, cross-validation techniques are used during the YOLO training process. This implies evaluating precision, recall, confidence scores, and other metrics of the model using this independent validation dataset, which includes validation images, Figure 7. This methodology ensures a complete evaluation of the model's capabilities on data that has not been previously observed [32].

Figure 7: Output "ValÏmages for truck training.

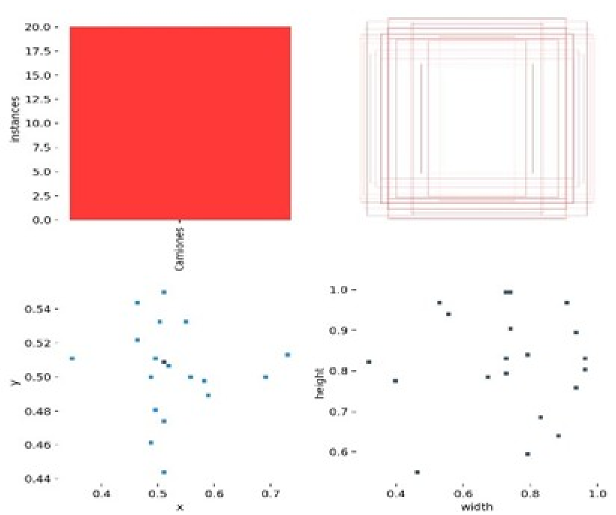


Source: Authors [32].

During the YOLO training process, the "`labels.png`" file serves to provide a visual representation of the labels or categories assigned to detected objects in the images used for training. This representation shows the original images with bounding boxes surrounding the detected objects, each labeled with its corresponding category.

The assignment of these labels, Figure 8, is done manually during the training process and consists of the object's class and the bounding box location. The generated image has an essential purpose in allowing supervision of the model's learning progress, ensuring accurate label assignment to objects, and observing how bounding boxes adapt during training. This visual feedback improves the capability to verify and refine the labels used to train the YOLO model, which ultimately contributes to improving its performance and precision in object detection [33].

Figure 8: Output of label correlation graphs for truck image training.

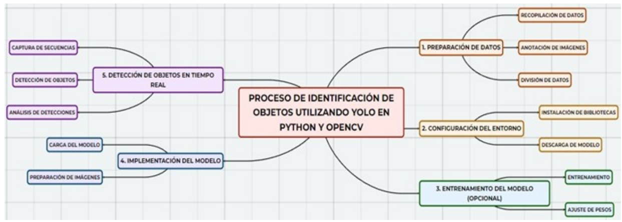


Source: Authors [33].

The implementation of YOLO in Python demands powerful hardware to perform real-time inference. The model's performance is subject to variations depending on the size and the input data used. The training process, Figure 9, covers data collection, labeling, and model training, which can require several hours of dedication [34].

Once training is concluded, it is imperative to evaluate the model using a test dataset to measure its performance and, if necessary, make adjustments. Once this evaluation is completed, the model can be implemented in real-time systems, such as security cameras, autonomous trucks, or drones.

Figure 9: General training scheme used for image recognition with YOLO.



Source: Authors [34].

3.4. Key Fundamentals in YOLO for Real-Time Road Actor Detection

Within the field of artificial intelligence and computer vision, YOLO (You Only Look Once) has emerged as a highly efficient and precise approach for object detection in images. In particular, YOLO has been widely used in road actor detection, such as trucks, pedestrians, and traffic signs, in applications related to autonomous driving and road safety. To achieve precise, real-time road actor detection, YOLO relies on a series of key fundamentals, which are examined below:

- **Loss function:** The loss function is a tool that quantifies the discrepancy between the model's predicted output and the actual data. This function is used to train the model, adjusting its parameters to minimize the loss function (Equation 1).

The YOLO loss function consists of several different components. Each component is designed to penalize specific errors in the model's predictions. The loss function components are divided as follows:

- **Box loss:** This component penalizes incorrect predictions in relation to the object's bounding box.
- **Class loss without object:** This component penalizes erroneous class predictions when no object is present.
- **IoU loss:** This component penalizes incorrect predictions in relation to the Intersection over Union (IoU) metric between predicted bounding boxes and training labels.
- **Compensation loss:** This component penalizes incorrect predictions regarding displacement adjustment between predicted bounding boxes and training labels.

The global loss function of YOLO is defined as the sum of all these components. In this equation, the different components of the loss function are described, identifying each of them as follows:

$$L = L_{cls} + L_{box} + L_{noobj} + L_{iou} + L_{comp} \tag{1}$$

Where:

- L_{cls}: Corresponds to classification loss.
- L_{box}: Represents the loss related to the bounding box.
- L_{noobj}: Indicates the classification loss without an object.
- L_{iou}: Refers to the IoU (Intersection over Union) loss.
- L_{comp}: Refers to the compensated loss.

- **Gradient descent:** Gradient descent is a technique used to minimize the loss function, adjusting model parameters in the direction where the loss function decreases most rapidly. This optimization is carried out according to Equation (2).

$$\frac{\partial L}{\partial \theta} = \frac{\partial L_{cls}}{\partial \theta} + \frac{\partial L_{box}}{\partial \theta} + \frac{\partial L_{noobj}}{\partial \theta} + \frac{\partial L_{iou}}{\partial \theta} + \frac{\partial L_{comp}}{\partial \theta} \tag{2}$$

Where:

- θ: Refers to the model parameters.
- L: Corresponds to the loss function.
- L_{cls}: Represents classification loss.
- L_{box}: Indicates loss related to bounding boxes.
- L_{noobj}: Is the classification loss when no object is present.
- L_{iou}: Is the loss related to the intersection.
- L_{comp}: Refers to the compensated loss.

- **Convolutional Neural Network:** A convolutional neural network (CNN) is a type of neural network structure specifically designed for image processing. In the context of YOLO, CNNs are employed for the task of identifying and categorizing objects present in images. This is described by Equation (3).

$$y = f(Wx + b) \tag{3}$$

Where:

- y: Is the output of the CNN.
- f: Is the activation function.
- W: Are the weights of the CNN.
- b: Are the biases of the CNN, and.
- x: Is the input to the CNN.

- **Vault and Regression:** Refers to a structure used for storing data, and in the context of YOLO, vaults are used to store both model weights and training data. Regression is classified as a type of machine learning approach used to make predictions related to continuous values. In YOLO, regression techniques are used to predict both the location and the type of objects present in images. This is described by Equation (4).

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix} \tag{4}$$

Where:

- B: Is the vault.
- b_{ij}: Is the object in cell i, row j.
- m: Is the number of cells in the vault, and.
- n: Is the number of objects in each cell.

Each object in the vault is represented as a five-dimensional vector, Equation (5):

$$[x, y, w, h, c] \tag{5}$$

Where:

- x and y: Are the coordinates of the bounding box center.
- w and h: Are the width and height of the bounding box.
- c: Is the object class.

The vault has the function of producing the YOLO output, which is presented as a list of objects. Each of these objects includes information such as a bounding box, a class, and a probability. The probability reflects the possibility that the object contained in the bounding box corresponds to the specific mentioned class.

4. Conclusions

The YOLO (You Only Look Once) model offers several significant advantages in the task of road actor identification, standing out for its high precision, efficiency, and real-time operation capability. Its capacity to detect and classify multiple objects simultaneously makes it highly suitable for traffic environments, benefiting traffic management and road safety.

Furthermore, its ease of implementation through open-source options such as Python has contributed to its growing popularity. However, it is important to point out that it presents some limitations, such as difficulty in detecting small or very close objects, as well as challenges in unfavorable lighting conditions or obstructions.

References

[1] Y. Shao, D. Zhang, H. Chu, X. Zhang, and Y. Rao, “A Review of YOLO Object Detection Based on Deep Learning,” 2021.

[2] S. Bhasin, “Real-time Object Detection with YOLO, OpenCV and Python,” 2019.

[3] Suresh et al., “Object Detection with YOLO for Intelligent Traffic Monitoring System,” 2020.

[4] Y. Liu, Q. Shi, W. Guo, and W. Liao, “A Real-time, Mobile-object Detection Approach for Unmanned Aerial Vehicle Based Forest Fire Surveillance System,” 2020.

[5] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of YOLO Algorithm Developments,” 2022.

[6] M. Tucci and A. B. (n.d.), “YOLO-S: A Lightweight and Accurate YOLO-like Network for Small Target Selection in Aerial Imagery.”

[7] A. Sharma, J. Pathak, M. Prakash, and J. N. Singh, “Object Detection using OpenCV and Python,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 8, no. 6, pp. 2736-2741, 2020.

[8] “Procesamiento de Imágenes y Visión Artificial con MATLAB Video,” Mathworks.com, 2021. https://la.mathworks.com/videos/image-processing-and-computer-vision-with-matlab-1597884648964.html (accessed Jul. 25, 2023).

[9] R. A. Gonzalez, R. Ferro, and D. Liberona, “Government and governance in intelligent cities, smart transportation study case in Bogotá Colombia,” vol. 11, no. 1, pp. 25–34, Mar. 2020. doi: https://doi.org/10.1016/j.asej.2019.05.002.

[10] B. E. Pineda, C. H. Muñoz, and H. Gil, “Aspectos relevantes de la movilidad y su relación con el medio ambiente en el Valle de Aburrá: una revisión,” *Ingeniería Y Desarrollo*, vol. 36, no. 2, pp. 489–508, 2018. https://www.redalyc.org/journal/852/85259689013/html/

[11] “AI for the Planet: Highlighting AI innovations for sustainable mobility and smart cities,” UNESCO.org, 2023. https://www.unesco.org/es/articles/ia-por-el-planeta-destacando-las-innovaciones-de-ia-para-la-movilidad-sostenible-y-las-ciudades

[12] C. A. Gómez Zapata, “Reconocimiento de objetos del hogar, usando redes neuronales convolucionales para personas con discapacidad visual,” *Revista Científica de Ingeniería y Tecnología*, vol. 2, no. 2, pp. 1-10, 2019. https://dialnet.unirioja.es/descarga/articulo/7436051.pdf.

[13] X. B. Olabe (n.d.), “ARTIFICIAL NEURAL NETWORKS AND THEIR APPLICATIONS,” Ehu.eus. Retrieved on July 8, 2023, from: https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

[14] J. Murgui and A. J. García-Sánchez, “Clasificación y reconocimiento de imágenes con redes neuronales para aplicaciones industriales,” 2018. URL: https://riunet.upv.es/bitstream/handle/10251/115464/Murgui.pdf?sequence=1

[15] G. Ortiz and A. I. Sánchez, “Emprendimiento y tecnologías de la información y la comunicación en Bogotá,” *Cuadernos de Administración*, vol. 36, no. 67, pp. 199-211, 2020.

[16] J. Torres and H. Acosta, “La innovación en el ecosistema emprendedor de Bogotá,” *Cuadernos de Administración*, vol. 35, no. 64, pp. 251-262, 2019.

[17] F. Uribe and J. Guzmán, “La colaboración público-privada en el fomento de la innovación en Bogotá: el caso de la identificación de objetos en el contexto vial,” *Revista Internacional de Gestión y Economía Aplicada*, vol. 11, no. 1, pp. 89-101, 2021.

[18] Research Center of the Universidad Distrital Francisco José de Caldas, 2023. https://revistas.udistrital.edu.co/index.php/visele/article/view/18942/18701

[19] R. C. U. Chiroma, “Vehicle detection, counting, and classification in traffic videos: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 3773-3785, 2021.

[20] S. S. Rao, “Vehicle detection and identification using computer vision and deep learning techniques,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 2827-2836, 2018.

[21] M. A. H. Akhand, “Vehicle Recognition from License Plate Number using Deep Learning,” *arXiv preprint arXiv:1903.09203*, 2019.

[22] S. M. García Ávila, C. A. Vega Camacho, J. V. Cadena López, R. A. González Bustamante, and P. A. Mateus Abaunza, “Diseño y aplicación de una herramienta para identificar y clasificar motocicletas mediante una red neuronal convolucional,” researchgate.net, 2021. doi: https://doi.org/ISBN:978-958-53278-6-3

[23] V. Sichkar, “Traffic Signs Detection by YOLO v3, OpenCV, Keras,” Kaggle.com, Apr. 15, 2022. https://www.kaggle.com/code/valentynsichkar/traffic-signs-detection-by-yolo-v3-opencv-keras (accessed Jul. 25, 2023).

[24] Motor Colombia, “7,270 deaths in traffic accidents in 2021,” Feb. 23, 2022. https://www.motor.com.co/industria/7.270-muertos-en-accidentes-de-transito-en-2021-20220124-0001.html

[25] R. Jiménez Moreno, O. Avilés, and D. M. Ovalle, “Red neuronal convolucional para discriminar herramientas en robótica asistencial,” *Visión Electrónica*, vol. 12, no. 2, pp. 208–214, Oct. 2018. doi: https://doi.org/10.14483/22484728.13996

[26] L. L. Hurtado-Cortés and J. A. Forero-Casallas, “Identification and fault detection in actuator using NN-NARX,” *Visión Electrónica*, vol. 2, no. 2, pp. 304–312, Dec. 2019. doi: https://doi.org/10.14483/22484728.18432

[27] Authors, “Fragment of the image training set for YOLO,” [Figure], 2023.

[28] Authors, “Confusion matrix of a training session on truck images,” [Figure], 2023.

[29] Authors, “Confidence-precision curve for truck image training,” [Figure], 2023.

[30] Authors, “Output Results.png for truck image training,” [Figure], 2023.

[31] Authors, “Output Train.png for truck image training,” [Figure], 2023.

[32] Authors, “Output Val.png for truck training,” [Figure], 2023.

[33] Authors, “Output of label correlation graphs for truck image training,” [Figure], 2023.

[34] Authors, “General training scheme used for image recognition with YOLO,” [Figure], 2023.