Impact of the Preprocessing Stage on the Performance of Offline Automatic Vehicle Counting using YOLO

Daniel Valencia, Elena Muñoz, and Mariela Muñoz-Añasco

Abstract— **Vehicle counting systems detect, classify, and count vehicles with sensors or image processing, providing valuable information for road management. Image processing systems provide detailed information on vehicle flow with adequate lighting conditions and a higher computational cost compared to sensor systems. The image processing systems with higher accuracy require higher computational cost. This feature limits the number of application cases in cities with low technology level. This research analyzes urban vehicle counting using an automatic image processing system using YOLOv5 in the vehicle detection-classification stage and the SORT algorithm in the tracking stage. The study used videos recorded from a pedestrian bridge in Popayan, Colombia, for an exploratory study of the influence of preprocessing operations on the performance of a low-tech vehicle counting system. The study performed a comparative statistical analysis to determine the impact of different settings on system performance. An ANOVA analysis evaluates the incidence of frame cut and reshape on YOLO processing. The results indicate that a 30% cut of the image area prior to YOLO processing produces the lowest weighted average error. In addition, the frame reshape only increases the processing time. The study proposes improvements in the performance of an offline automatic vehicle counting system from the video preprocessing stage.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/8943

Index Terms— **Image processing, Object tracking, Traffic control, Vehicle detection.**

I. INTRODUCTION

Constant urban growth generates many vehicles on the roads of a city at different hours of the day. Uncontrolled vehicular flow generates traffic congestion and increases urban travel time [1]. Road management regulates vehicular flow through the distribution of roads, the choice of the direction of vehicular travel, and the planning of new roads for the city [2]. Accurate monitoring of urban vehicular flow is essential to ensuring efficient road mobility management [1]. Vehicle counting determines the number, speed, and types of vehicles traveling on a road [2]. Urban road management should establish adequate use of roads to minimize traffic jam [3], and automated vehicle counting provides valuable information for road management [4].

Automatic vehicle counting uses magnetic induction sensors, radar, infrared, and image processing [5] [6]. These systems detect vehicles, count their number, measure speed, and classify vehicle types [6]. As nonintrusive methods, they collect detailed information about traffic patterns and support the optimization of vehicular flow [7]. Sensor-based vehicle counting methods calculate the number and speed of vehicles but have difficulty distinguishing among distinct types of vehicles [8]. On the other hand, image processing under proper lighting conditions measures the number and types of vehicles, speed, flow patterns, and roadway trends [9].

Vehicle counting methods with image processing have different advantages and limitations for collecting detailed information on urban traffic flow. One of the advantages is their ability to capture detailed visual information on vehicle quantity and classification, movement pattern analysis, and speed. However, the main disadvantage is the computational cost of implementing image processing compared to the use of sensors [10], this limits the number of application cases in cities with low technology level**.** Vehicle counting by image processing has two stages: detection-classification and tracking. For the information it provides, the stage of greatest impact is detection-classification. Both static and dynamic methods can be used in this stage [11]. Static methods use techniques such as histograms of oriented gradients, background subtraction, or successive comparisons of frames to determine the presence, type, and speed of vehicles in a certain region of the analyzed video [12]. Dynamic methods use trained convolutional neural network (CNN) to identify, count, and classify vehicles with speed measurement [13]. The limitations of static methods are adverse weather conditions and lighting variations, depending on their calibration [14]. Dynamic methods are adaptive to different environmental scenarios, but they increase the computational cost compared to static methods [15]. A commonly used dynamic method is YOLO (You Only Look Once) because it is a CNN with high processing speed [16], [17]. In the vehicular tracking stage, there are methods such as least squares [18], particle filtering [19], color histogram tracking [20], feature tracking [21], and SORT (simple online real-time tracking) [22], [23]. Any vehicle tracking method relies on the information provided by the vehicle detection-classification method.

Multiple projects have studied automatic vehicle counting using YOLO for vehicle detection-classification. Jiao and Wang proposed a system with vehicle tracking using a Kalman filter to

D. Valencia, E. Muñoz, and M. Muñoz-Añasco are with Universidad del Cauca, Popayán, Colombia (e-mails: danielvalencia@unicauca.edu.co, elenam@unicauca.edu.co, and mamunoz@unicauca.edu.co).

make vehicle inferences under occlusion conditions, reducing the computational cost by implementing masks in each frame of the videos [24]. Bisht et al. incorporated the detection-classification of high-priority vehicles such as ambulances, fire trucks, and police patrols under low, medium, and high vehicular traffic conditions. The mean average precision (mAP) of YOLO was 62.2% with 35 frames per second (FPS) of processing [25]. Dave et al. analyzed different versions of YOLO for the vehicle detection-classification stage, evaluating accuracy and processing time with up to 50% object overlap in day and night illumination conditions, but found that shorter processing time causes lower accuracy independent of illumination type [26]. In general, in research related to automatic vehicle counting using YOLO, the higher the processing speed or the older the YOLO version for low computational cost, the less likely it is to achieve vehicle classification accuracy above 90%—and there is no information on the type of YOLO configuration prior to implementation [27].

Regarding vehicular image preprocessing, Jeong et al. [28] generate images by placing the objects of interest on real backgrounds to increase the size of the YOLOv2 training dataset. For license plate recognition with YOLOv3, Setiyono et al. [29] increased the brightness and reduced the noise of the training dataset images, achieving 98.2% accuracy. For the detection of various vehicles and pedestrians in cloudy weather conditions with YOLOv3, Liu et al. [30] varied the contrast and brightness of the images prior to YOLO processing, achieving a mAP of 72.03% with haze. These researches performed image preprocessing to increase YOLO accuracy without analyzing the impact of these techniques on processing time.

The present research developed a vehicle counting system using YOLOv5 in the detection-classification stage and SORT in the tracking stage. The SORT algorithm relies exclusively on the information provided by YOLO. The goal of the developed system is to ensure adequate vehicle counting with low computational cost. The evaluation of the vehicle counting system used videos recorded from a pedestrian bridge in the city of Popayan, Colombia. To develop an exploratory study of the influence of preprocessing operations on the performance of the proposed system, its algorithm allows generating 10, 30, and 50% vertical cuts and reshapes from 640 x 480 pixels to 640 x 640 pixels of the video frames before YOLO processing. An ANOVA analysis evaluated the incidence of frame cuts and reshapes prior to YOLO processing. The performance metrics for this research are the weighted average relative counting error and the processing time increment. An ANOVA with Tukey HSD (honestly-significant-difference) test indicate that a 30% cut of the image area before YOLO processing provides the lowest weighted average error compared to all other cases. The reduction in processing time is proportional to the frame cut performed, with the best case being a 50% cut. Regarding the frame reshape, it only influences the increase in processing time, so its implementation is not advisable.

II. Materials and Methods

This research evaluates the influence of image preprocessing for YOLO used in a vehicular counting system with low computational cost. This section analyzes the limitations of YOLO, such as input image size and object display resolution. Followed by the YOLO architecture selection according to the stablished conditions. As such, the study proposes to analyze the impact of two preprocessing operations on the performance of YOLO in an automatic vehicle counting system: cutting and reshaping images.

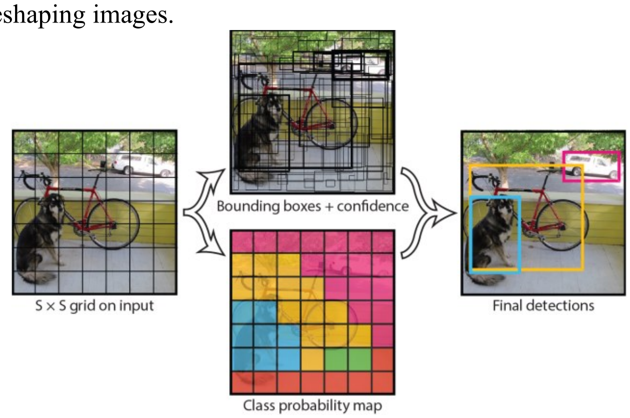


Fig. 1. YOLO operating scheme [31].

The following two sections describe the vehicular flow videos used along with the dataset, training, and validation of the CNN YOLO used in the study. The next section explains the performance metrics proposed for the different tests with the levels of frame cut (0%, 10%, 30%, and 50%) and frame reshape (640x480 pixels and 640x640 pixels) evaluated in the vehicular counting videos. The last section describes the statistical analysis with ANOVA analysis and the Tukey HSD test.

A. Image Preprocessing for YOLO

YOLO is known for its ability to detect and classify objects in images and videos with a higher processing speed than other CNN algorithms, such as faster region-based convolutional neural network [31]. The visual processing starts by dividing the input image into a grid with SxS frames (where S is a predetermined number multiple of 32). For each frame and each possible associated anchor box, YOLO predicts the coordinates (x-center, y-center, width, height) of the bounding box, the probability that it contains an object, and the probability of a specific class. Thresholds on confidence and class predictions filter out irrelevant or unreliable detections. To reduce the redundancy of detections, non-maximum suppression eliminates duplicate detections based on the overlap and confidence of the detections. The final output is a list of bounding boxes along with the class labels and associated confidence probabilities (Fig. 1) [31].

Since its initial structure presented in 2016 [32], YOLO has performed object detection with image splitting with the SxS grid. Likewise, the prediction confidence depends on the input image resolution. Two limitations of YOLO are the size of the input image (the larger the image, the longer the processing time) and the display resolution of the objects. If there are too many small objects with respect to the SxS grid, there is a lower possibility of correctly detecting the object. Because of these limitations, this research proposes to analyze the influence of two image preprocessing operations on the performance of YOLO in an offline automatic vehicle counting system, as follows:

- Cut: In all vehicle counting systems [27], the counting region is close to the recording point; the information of interest is in a certain area. Therefore, the frame cut eliminates regions that do not influence the vehicle count and classification. This study analyzes a cut of 0 (C0), 10 (C10), 30 (C30), and 50% (C50) of the background region of each of the video frames.
- Reshape: In training, YOLO resizes images to a square shape for faster identification. If the object of interest has a low resolution with respect to the SxS grid, the predictive confidence of YOLO decreases. A reshape step before sending the video frames to YOLO generates the objects of interest to be as square as possible to increase the probability of a correct prediction. This study analyzes the influence of normal-sized frames (R0-640x480 pixels) and resized frames (R1-640x640 pixels). To ensure correct object detection, the study does not consider a resized frame of 480x480 pixels.

B. YOLO Architecture Selection

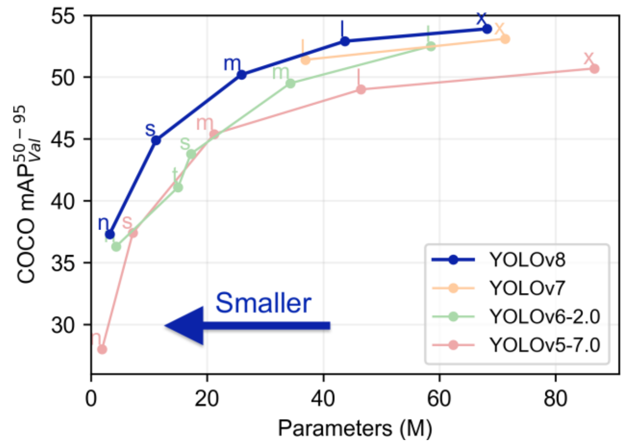
Initially, this research considered the evaluation of YOLO v3, v4, v5, v6, v7, v8 for vehicle detection and classification [27][31]. YOLOv3 is the first version of YOLO with backbone, but the newer versions are built with upgrades of its. The architectures of YOLOv4 and YOLOv5 are similar, and YOLOv5 has a continuous upgrade reached in the end of 2022 the version 7.0. Therefore, the YOLO versions evaluated are v5, v6, v7, and v8. The selection process is based on the metrics of AP (average Precision) and FPS (frame per second) with an image size of 640 pixels described in the review by Terven et al. [32]:

- YOLOv5-7.0: 50.7% AP and 200 FPS on a NVDIA V100.
- YOLOv6: 52.8% AP and 98 FPS on a NVDIA Tesla T4.
- YOLOv7: 53.1% AP and 114 FPS on a NVDIA V100.
- YOLOv8: 53.9% AP and 280 FPS on a NVDIA A100 and TensorRT

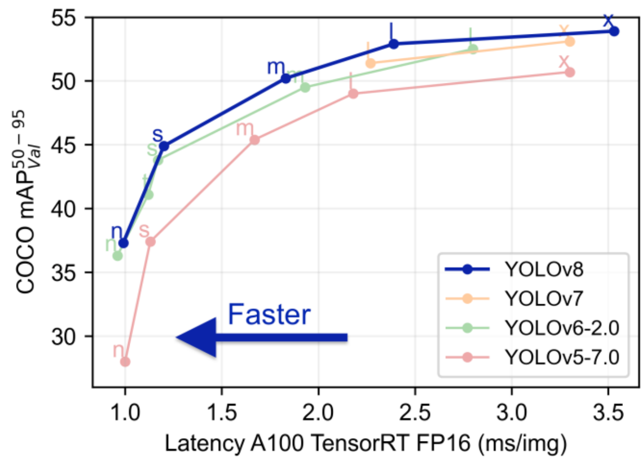
YOLOv8 has the highest FPS for an image size of 640 pixels, followed by YOLOv5-7.0. The difference between the lowest (YOLOv5-7.0) and the highest (YOLOv8) AP is 3.2%. To achieve this difference, YOLOv8 requires a higher computational cost to implement the model with a complex backbone, neck and head structure (Fig. 2) [32]. According to the criterion of low computational cost, this study worked with YOLOv5-7.0 architecture.

C. Traffic Flow Videos

The investigation by Hurtado-Gomez et al. [33] provided the videos analyzed for vehicle counting for the present investigation. The location for the vehicle counting study was near a three-lane traffic signalized intersection in the city of Popayan, Colombia. The recording of the videos was from a 4.5-m high pedestrian bridge with a 45° camera zenithal tilt, according to previous studies [27]. The videos are in mp4 format with a resolution of 640x480 pixels and a duration between 3 and 4 minutes.



a. COCO mAP vs Parameters in millions.



b. COCO mAP vs Latency A100 Tensor RT.

Fig. 2. Performance comparison of YOLO object detection models [32].

To ensure a constant vehicular flow and adequate illumination, the selected videos were recorded in the morning and afternoon hours, giving a total of six videos for vehicle counting analysis. A seventh video was used to extract frames and generate a specific training dataset (see Section 2.4.2), ensuring that in each selected video, objects of the classes of interest (bus-B, car-C, motorcycle-M, and truck-T) are present. Subsequently, as a point of comparison for the automatic vehicle counting system, the present study performed manual vehicle counting on the selected videos with total of vehicle count (VC) and video duration in seconds (VD) (Table I).

TABLE I
MANUAL VEHICLE COUNT OF THE VEHICULAR FLOW VIDEOS

	Count Video 1	Count Video 2	Count Video 3	Count Video 4	Count Video 5	Count Video 6
B	4	11	9	17	14	15
C	65	104	103	94	82	69
M	47	39	42	42	38	35

T	5	12	2	15	15	13
VC	129	168	159	171	151	136
VD	231	219	245	224	222	233

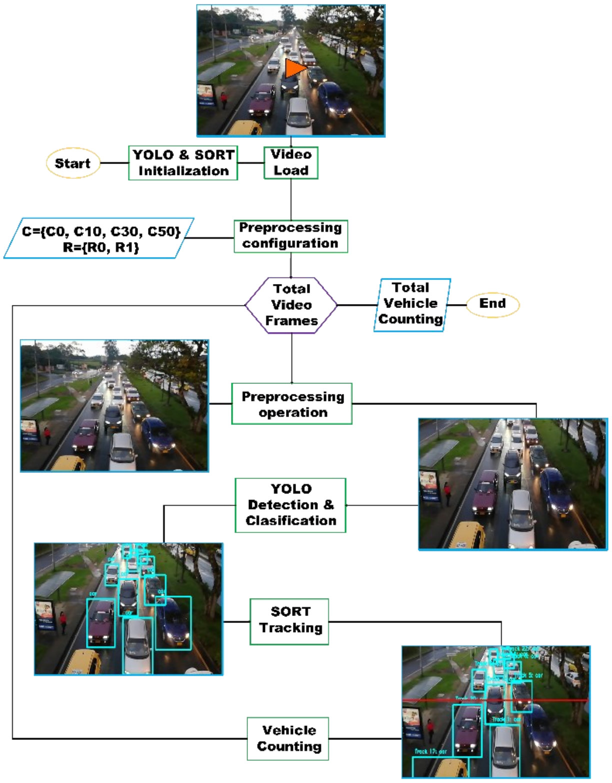


Fig. 3. Diagram of the operation of the proposed system.

D. Automatic Offline Vehicle Counting System

The proposed system has two stages: the detection-classification stage using YOLOv5-7.0 [32] and the tracking stage using SORT. The choice of YOLOv5-7.0 over other versions is because of its ease of use, adaptability to various requirements, performance, and speed. The tracking stage uses SORT because this algorithm works with bounding boxes. YOLO provides these components, allowing proper integration of the two stages of operation [22]. An Intel Core i7 12650H computer with an NVIDIA GeForce RTX 3060 graphics card (6 GB) and 16 GB of RAM developed all the tests of the vehicle counting system.

The operating process of the developed system (Fig. 3) is as follows:

1. Algorithm initialization: selection of the relevant YOLO (YOLO trained with the types of vehicles for detection in the videos) and configuration of the SORT tracking algorithm (maximum number of frames without receiving a tracking update, minimum number of tracks, and intersection over union threshold to generate tracking).
2. Video load: loading of the study vehicle flow video.
3. Preprocessing configuration: adjustment of the processing variables to generate a vertical cut of the frame (0, 10, 30, or 50% of the final part of the frame) and a reshape of the frame (640x480 or 640x640 pixels).
4. Video processing: playback of the video frame-by-frame with preprocessing configured followed by vehicle detection, classification, and tracking. The algorithm counts vehicles as each one crosses a vertical threshold.
5. Display of results: once the video processing is finished, the algorithm displays the vehicle counting information obtained (number of buses, cars, motorcycles, and trucks).

E. YOLO for Vehicle Counting

- YOLO training

There are five standard versions of YOLOv5: nano (N), small (S), medium (M), large (L), and extra-large (X). The difference between each is the number of neural layers and interconnections between them that make up the CNN. The YOLOv5 X version has the highest confidence level in the classification of the object of interest and the longer the processing time.

All versions of YOLOv5 have been pretrained with the common objects in context (COCO) image dataset. This dataset has more than 200,000 labeled and segmented images for training in the recognition of 80 classes of objects such as people, animals, cars, airplanes, and ships, among others. The use of a pretrained YOLOv5 in a specific case does not ensure high accuracy in the recognition of the objects of interest. In this research, the training of YOLOv5 presents a general phase for matching the CNN with the classes of interest and a specific phase for recognition in the operating environment of the vehicular counting system.

- General training phase

In the general phase, using the Python library "FiftyOne", from the COCO images, the research created a dataset of 12,000 images with the classes of interest called COCO-vehicles (Fig. 4) to generate more detailed training.

With the COCO-vehicle dataset, the training of the five versions of YOLOv5 with learning transfer froze the 10 layers of the backbone. All training was done with 600 epochs and an early patience stop of 300 epochs. The YOLOv5 X version obtained the best confusion matrix; however, in preliminary tests with the vehicular counting system, the processing time increased by over 400% compared to the video time. For this reason, the proposed vehicular counting system implemented the YOLOv5 L version with the second-best confusion matrix (Fig. 5), and higher latency than X version (see Fig. 2).

Fig. 4. Confusion matrix of YOLOv5 L with COCO-vehicle.

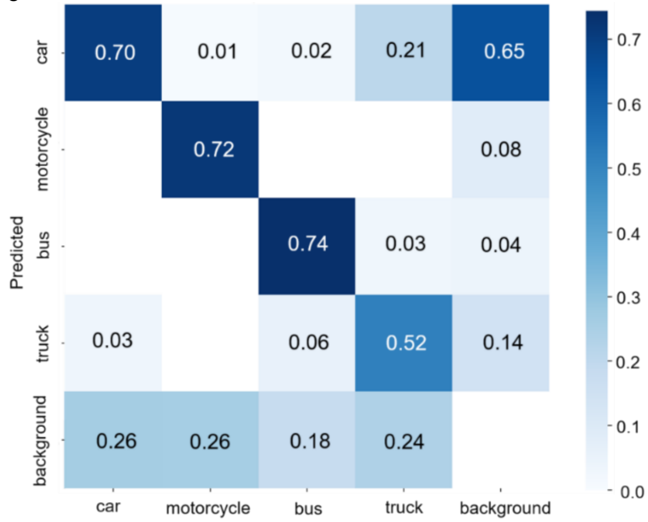


Fig. 5. Confusion matrix of YOLOv5 L with COCO-vehicle.

Fig. 6. Training image of the specific dataset.

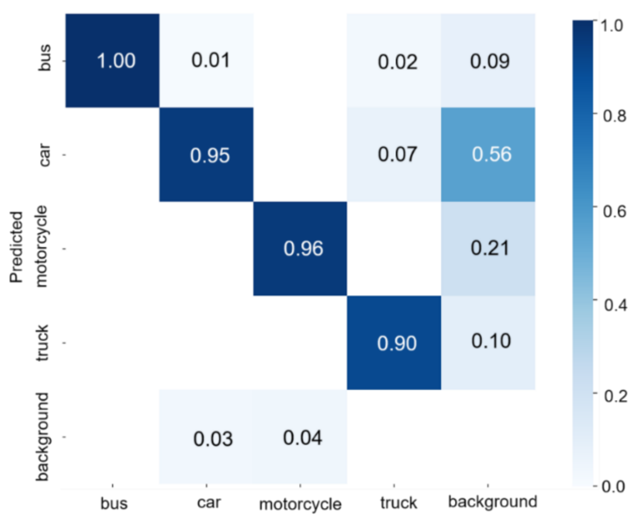


Fig. 7. Confusion matrix of YOLOv5 L with COCO-vehicle and specific dataset.

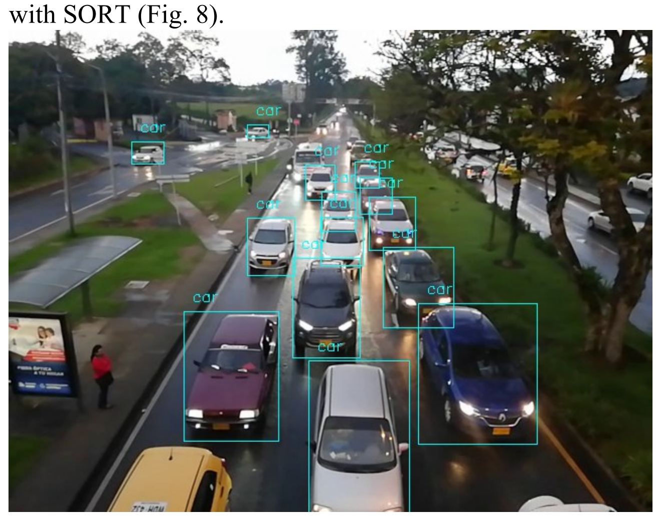
- Specific training phase

From the seventh vehicle flow video provided by Hurtado-Gomez et al. [33] with road traffic and lighting conditions like the study videos, the specific training phase extracted frames in which it identified the classes of interest with their respective segmentation in Roboflow [34]. The data augmentation process with the distortion, exposure, saturation, brightness, vertical rotation, and Gaussian noise filters provided 212 images (Fig. 6).

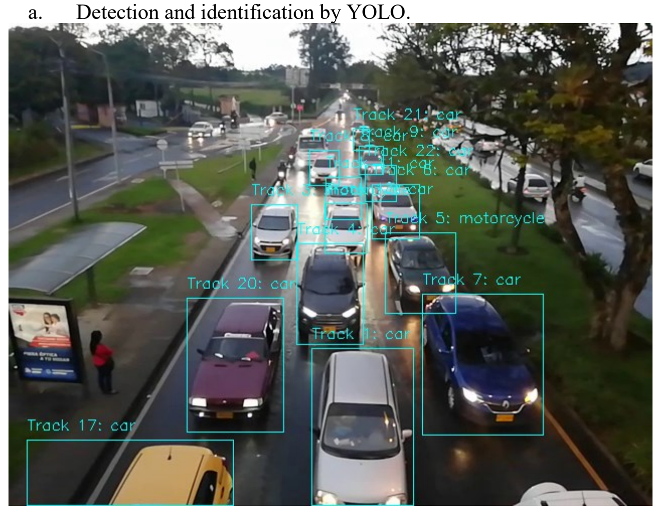
The second YOLOv5 L training was with a specific dataset and transfer learning by freezing the 10 layers of the backbone. The training had 600 epochs, and an early patience stop of 300 epochs. The confusion matrix associated with the specific training has a correct prediction percentage above 90% (Fig. 7).

- Recognition of classes of interest

After successful validation of YOLOv5 L, a video analysis verified the proper detection and classification of vehicles of interest in the vehicle counting videos for subsequent tracking with SORT (Fig. 8).



a. Detection and identification by YOLO.



b. Tracking by SORT

Fig. 8. Developed automatic vehicle counting process.

F. Performance Metrics of the Proposed Vehicle Counting System

Because the proposed system provides the count of vehicles of interest in each of the vehicular flow videos as performance metrics, the present study from (1) proposes a weighted average relative counting error (WARCE) of the classes (2) and a processing time increment (PTI) for each preprocessing configuration (3).

$$E_R i = \frac{|C_s i - C_m i|}{C_m i} * 100\% \quad (1)$$

Where:

i: Is a particular class (B, C, M or T).

$E_R i$: Is the relative error associated to the class i

$C_s i$: System vehicle count associated to class i.

$C_m i$: Manual vehicle count associated to the class i

$$\overline{E_R} = \frac{\sum E_R i * C_m i}{\sum C_m i} * 100\% \quad (2)$$

Where:

$\overline{E_R}$: weighted average relative counting error (WARCE)

$\sum E_R i * C_m i$: Influence by the number of vehicles of the relative error associated with class i.

$\sum C_m i$: Total number of vehicles counted manually.

$$T_I = \frac{T_p}{T_v} * 100\%. \quad (3)$$

Where:

T_I : Processing time increment (PTI)

T_p: Processing time

T_v: Video duration time

Typically, automatic vehicle counting systems perform a global count of vehicles without detailing the counting error associated with each of the classes of vehicles studied, so they use accuracy according to the correct and erroneous detection of objects (4) [14].

$$A_D = \frac{TP}{TP+FP} * 100\% \quad (4)$$

Where:

A_D: Detection Accuracy.

TP: True Positives, value of correct identifications.

FP: False Positives, value of false identifications.

As a comparison metric with respect to other vehicle counting systems, the present research will use the count accuracy (A_C) with respect to the counting error [36] according to WARCE (5). This type of accuracy is based on detailed information about the counting error associated with each vehicle class studied.

$$A_C = 100\% - \overline{E_R} \quad (5)$$

G. Performance Evaluation of the Proposed Vehicle Counting System

According to the exploratory study of the influence of preprocessing operations on the performance of the proposed system, the analysis factors are the frame cut with levels of 0 (C0), 10 (C10), 30 (C30), and 50% (C50) and the frame reshape of 640x480 (R0) and 640x640 pixels (R1). With the WARCE and PTI data for each of the videos in all combinations of the factors, an ANOVA analysis with an α of 0.05 evaluates the influence of these on the performance of the vehicle counting system. Subsequently, a Tukey HSD test determines the best possible preprocessing configuration according to the established factors.

The present research proposes two hypotheses:

- There is an influence of the frame cut in the video preprocessing stage on the performance of a vehicle counting system that uses YOLO for vehicle detection and classification.
- There is an influence of the frame reshape in the video preprocessing stage on the performance of a vehicle counting system using YOLO for vehicle detection and classification.

III. Results and Discussion

A. Measurement of Vehicle Capacity

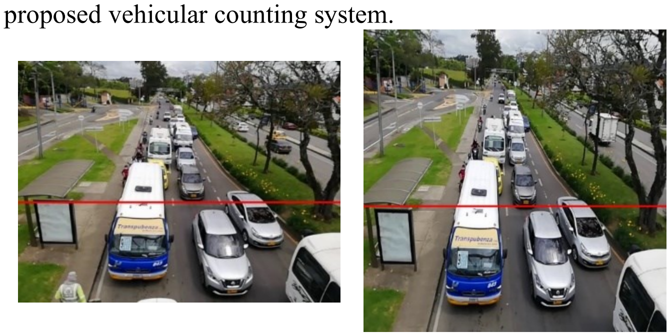
In all the selected videos, the developed system performed the vehicle counting with each of the eight possible combinations between the cut and reshape levels of the frame (C0-R0, C10-R0, C30-R0, C50-R0, C0-R1, C10-R1, C30-R1, and C50-R1; see Fig. 9).

Regardless of the combination, the count threshold was the same reference region for the vehicles of interest (Table II).

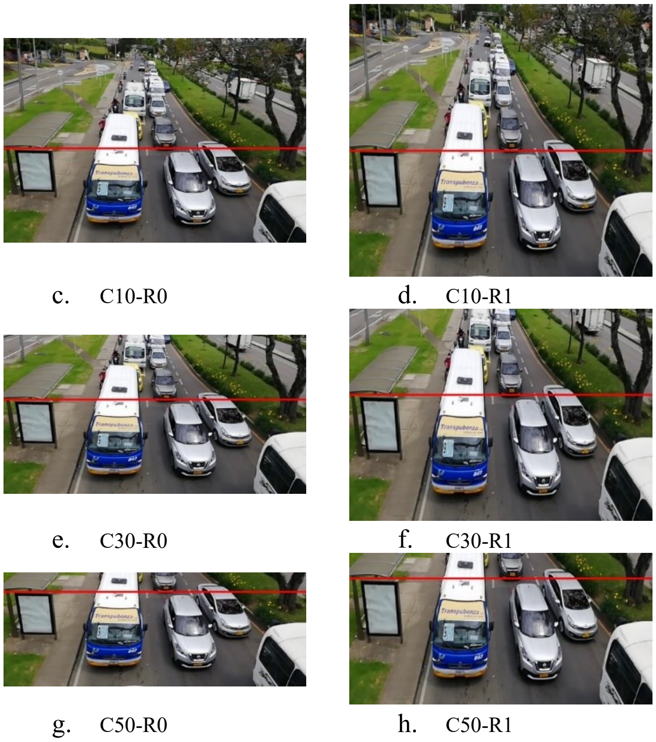
The information processing of all videos calculates WARCE and TPI metrics to statistically analyze the influence of the preprocessing performed on each frame (Table III).

B. Statistical Analysis

The data obtained satisfy with an adequate and statistically reliable distribution because the statistical residuals do not present normality, equality of covariance, or dependence among them. After corroborating the statistical validity of the experimental data, the ANOVA analysis and Tukey HSD test for both performance metrics evaluated the incidence of cut, reshape, and videos studied in the results obtained. Regarding this last parameter, for WARCE (Table IV) and TPI (Table VI), there is no influence of the processed vehicular flow video, i.e., there is homogeneity between the studied videos, and the videos do not affect the results of the performance metrics of the proposed vehicular counting system.



a. C0-RO b. C0-R1



c. C10-R0 d. C10-R1

e. C30-R0 f. C30-R1

g. C50-R0 h. C50-R1

Fig. 9. Preprocessing configurations for offline automatic vehicle counting system.

TABLE II
COUNTING AND PROCESSING TIME OF THE PROPOSED VEHICLE COUNTING SYSTEM

	C0-R0	C10-R0	C30-R0	C50-R0	C0-R1	C10-R1	C30-R1	C50-R1
				Video 1				
B	1	6	3	1	2	5	3	1
C	78	75	66	71	80	77	69	83
M	40	38	50	46	42	42	49	42
T	5	2	1	5	1	2	3	3
VC	124	121	120	123	125	123	124	129
VD	331.08	345.05	320.58	322.17	432.66	432.13	430.25	317.91
				Video 2				
B	27	18	10	10	12	17	6	17
C	89	93	94	108	101	98	104	109
M	36	39	50	27	37	32	42	26
T	15	13	7	13	14	16	8	13
VC	167	163	161	158	164	163	160	165
VD	313.22	348.35	298.47	276.22	363.05	338.99	313.04	275.69
				Video 3				
B	6	6	12	13	8	7	8	15
C	111	106	105	109	97	95	89	94
M	24	28	29	19	21	20	29	20
T	3	7	2	0	2	5	2	1
VC	144	147	148	141	128	127	128	130
VD	377.79	293.99	341.73	342.57	474.79	457.24	399.63	349.75
				Video 4				
B	24	22	15	12	19	25	20	19
C	105	111	97	106	102	101	96	118
M	40	33	49	41	46	44	50	33
T	14	9	14	14	17	13	13	10
VC	183	175	175	173	184	183	179	180
VD	331.79	355.34	307.48	314.56	391.67	403.06	318.65	252.26
				Video 5				
B	6	10	8	5	3	8	11	9
C	87	79	89	97	98	89	85	95
M	45	40	41	35	40	40	39	32
T	12	20	11	14	10	13	18	13
VC	150	149	149	151	151	150	153	149
VD	327.24	360.88	314.14	268.86	412.94	413.09	339.68	290.31
				Video 6				
B	9	11	11	5	3	5	5	5
C	81	75	69	75	86	92	75	74
M	35	37	38	32	30	25	38	39
T	9	8	12	15	13	10	13	9
VC	134	131	130	127	132	132	131	127
VD	380.28	389.46	327.06	311.06	448.64	424.83	364.49	302.17

TABLE III
PERFORMANCE METRICS OF THE PROPOSED VEHICLE COUNTING SYSTEM

Performance metrics	C0-R0	C10-R0	C30-R0	C50-R0	C0-R1	C10-R1	C30-R1	C50-R1
				Video 1				
WARCE (%)	19.01	19.83	7.44	8.26	21.49	17.36	7.44	23.14
TPI (%)	143.32	149.37	138.78	139.47	187.30	187.07	186.26	137.62
				Video 2				
WARCE (%)	22.29	11.45	16.27	10.84	4.82	13.86	7.23	15.06
TPI (%)	143.02	159.06	136.29	126.13	165.78	154.79	142.94	125.89
				Video 3				
WARCE (%)	19.23	16.03	11.54	22.44	17.95	22.44	17.95	24.36
TPI (%)	154.20	120.00	139.48	139.83	193.79	186.63	163.11	142.76
				Video 4				
WARCE (%)	12.50	22.02	7.74	11.31	9.52	11.31	8.93	23.81
TPI (%)	148.12	158.63	137.27	140.43	174.85	179.94	142.26	112.62
				Video 5				
WARCE (%)	15.44	9.40	13.42	18.79	22.82	11.41	6.71	17.45
TPI (%)	147.41	162.56	141.50	121.11	186.01	186.08	153.01	130.77
				Video 6				
WARCE (%)	16.67	12.88	6.06	15.91	25.76	34.85	14.39	17.42
TPI (%)	163.21	167.15	140.37	133.50	192.55	182.33	156.43	129.69

- *Influence of Preprocessing on WARCE*

For the WARCE case, with an α of 0.05 and a p-value of 0.009359, only the frame cut influences the results (Table IV), which shows that the variation of the processing area affects the accuracy of the proposed vehicle counting system.

In comparing each of the cut levels using a Tukey HSD test with an α of 0.05, the 30% cut (C30) of the frame is different from all other cuts (Table V). Analyzing the thresholds in each case of the C30 is the lowest value, indicating that this cut generates lower WARCE.

TABLE IV
ANOVA OF WARCE

	Degrees of freedom	Sum of squares	Mean squares	F-Value	P-Value
Cut	3	414.2951	138.0984	4.460804	0.009359
Reshape	1	53.57179	53.57179	1.730457	0.196905
Video	5	253.4114	50.68228	1.637121	0.17598
Residues	35	1083.536	30.95818		

TABLE V
TUKEY HSD TEST OF WARCE FOR FRAME CUT

	Difference	Lower threshold	Upper threshold	P-Value
C10-C0	-0.38901	-6.51501	5.736999	0.998179
C30-C0	-6.86452	-12.9905	-0.73852	0.023091
C50-C0	0.109214	-6.01679	6.235219	0.999959
C30-C10	-6.47551	-12.6015	-0.34951	0.034944
C50-C10	0.49822	-5.62779	6.624225	0.996204
C50-C30	6.973734	0.847729	13.09974	0.020499

TABLE VI
ANOVA OF TPI

	Degrees of freedom	Sum of squares	Mean squares	F-Value	P-Value
Cut	3	10051.71	3350.569	28.50876	1.64E-09
Reshape	1	4223.493	4223.493	35.93615	7.85E-07
Video	5	1226.361	245.2722	2.086931	0.090402
Residues	35	4113.469	117.5277		

TABLE VII
TUKEY HSD TEST OF TPI FOR FRAME CUT

	Difference	Lower threshold	Upper threshold	P-Value
C10-C0	-0.4958	-12.4318	11.44023	0.999487
C30-C0	-18.4889	-30.4249	-6.55288	0.001026
C50-C0	-34.9801	-46.9161	-23.0441	1.6E-08
C30-C10	-17.9931	-29.9291	-6.05708	0.001413
C50-C10	-34.4843	-46.4203	-22.5483	2.21E-08
C50-C30	-16.4912	-28.4272	-4.55515	0.003653

- Influence of Preprocessing on TPI

For the case of TPI, the ANOVA test with an α of 0.05 cut and reshape of the frame influences the results (Table VI). This shows that cuts and reshapes of the processing area affect the processing time of the proposed vehicle counting system.

In a pairwise analysis of the different levels of the analysis factors using a Tukey HSD test, in the case of frame reshape, a comparison of R1–R0 with a difference of 18.7605372, a lower threshold of 12.40724811, an upper threshold of 25.11382628, and a p-value less than 0.0001 shows that a reshape of 640x640 pixels on the frames increases the processing time. Regarding the frame cut (Table VII), the only case where there is no statistical difference is between C0 and C10. In all other cases, a higher cut level means a shorter processing time.

C. Discussion

The ANOVA analysis with an α of 0.05 responds to the two hypotheses posed. The frame cut influences both WARCE (p-value 0.009359) and TPI (p-value less than 0.0001). However, the frame reshape influences only the TPI (p-value less than 0.0001).

Analyzing the Tukey HSD tests performed, a 30% frame cut generates the lowest WARCE. Regarding TPI, a frame cut of 50% achieves the lowest TPI value; however, a reshape of 640x640 pixels in the frames increases the TPI. With these results, considering that the best performance of the proposed offline automatic vehicle counting system is the lowest value of WARCE and TPI, it is advisable to perform in the preprocessing stage a frame cut of 30% without reshape (640x480 pixels). When averaging the performance metrics associated with these parameters (C30-R0) in each of the vehicle flow videos, there is an average WARCE of 10.42%—a count accuracy of 89.58% and a TPI of 138.95%.

The influence of cutting the frame on the weighted average error comes from dropping unnecessary information in the frames to perform the vehicle count, considering that excessive cutting makes it difficult to properly identify objects close to the vehicle count region. In addition, the frame cut generates a smaller image size so that YOLO will process and identify the possible objects of interest in less time. For the same reason, frame reshape generates a larger image, which implies a longer processing time.

Regarding other automatic vehicle counting systems, the counting accuracy of the developed system has similar performance (Table VIII). All the systems analyzed perform offline processing with recorded videos of roads in different countries with daylight conditions, considering both detection accuracy and counting accuracy as performance metrics.

All the research uses CNN-based methods; only Doménech-Asensi et al. propose the use of a Bayesian network. In nighttime illumination conditions, this research obtained an accuracy of 69.75% [35]. The reported accuracies are between 80% and 90%; Castelló et al. report the lowest accuracy as they analyze each of the classes (person, bicycle, car, motorcycle, bus, train, and truck) with different YOLO activation functions to reduce processing time [38]. In Gomaa et al., all the tests performed report an accuracy of 100% with no differences between the different classes of vehicles and a high processing time compared to YOLOv2 [37]. In this aspect, the investigations with the count analysis by the classes of interest report lower accuracy compared to the vehicle count analysis globally. Song et al. perform a video preprocessing step, placing a mask over the frames to remove regions that are not highways [36]. The present investigation obtained an average count accuracy of 89.58% considering each of the study classes by generating a 30% cut of the frames in the video preprocessing stage.

IV. CONCLUSIONS

This research developed an automatic offline vehicle counting system using YOLO in the vehicle identification and classification stage and SORT in the tracking stage. The validation of this system was done with videos of vehicle flow in the city of Popayan, Colombia, under morning and afternoon lighting conditions on sunny days with traffic associated with working days.

TABLE VIII
COMPARISON OF AUTOMATIC VEHICLE COUNTING SYSTEMS

Research	Processing algorithm	Accuracy	Accuracy value (%)	Count analysis
Gomaa et Al. [35]	CNN	A_C	96.30	Global
Gomaa et Al.[36]	Faster – RCNN	A_C	100.00	Global
Castelló et Al.[37]	YOLOv3/v4	A_D	60.38	By class
Wieczorek et Al. [38]	CNN	A_C	82.50	By class
Doménech-Asensi et Al. [39]	Bayesian Network	A_D	86.50	Global
Song et Al.[40]	YOLOv3	A_D	88.00	By class
Own	YOLOv5	A_C	89.58	By class

This research proposed an exploratory study of the influence of preprocessing operations on the performance of the proposed system to improve its operation, the study evaluated the influence of frame cut and reshape. The performance metrics are the WARCE of vehicles with respect to the manual count and TPI with respect to the video duration time. In all the trials performed, the frame cut positively influences the reduction of both performance metrics; however, the frame reshape has a negative influence, generating a higher TPI. With the tests performed and the results obtained, the best preprocessing configuration is a 30% cut frame without reshape, obtaining the lowest WARCE and the second lowest TPI.

According to the result of this exploratory study, possible future works must evaluate the proposed vehicle counting system with a higher number of videos in different weather and lighting conditions, and even in different cities. Other works may focus on an exhaustive benchmarking of different YOLO architectures with the preprocessing operations before the stage of vehicle detection-classification.

ACKNOWLEDGEMENTS

The authors thank to engineers Hurtado, Romo, Cabrera, de la Cruz and Molina for providing the traffic flow videos used in this research.

REFERENCES

[1] R. Baldacci, A. Mingozzi, and R. Roberti, “Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints,” Eur J Oper Res, vol. 218, no. 1, pp. 1–6, 2012. doi.org/10.1016/j.ejor.2011.07.037.

[2] L. Li, R. Jiang, Z. He, X. M. Chen, and X. Zhou, “Trajectory data-based traffic flow studies: A revisit,” Transp Res Part C Emerg Technol, vol. 114, pp. 225–240, 2020. doi.org/10.1016/j.trc.2020.02.016.

[3] C. R. Bennett, H. De Solminihac, and A. Chamorro, “Data collection technologies for road management,” 2006.

[4] H. Yu et al., “Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives,” Transp Res Part C Emerg Technol, vol. 127, p. 103101, 2021. doi.org/10.1016/j.trc.2021.103101.

[5] V. Shepelev, S. Aliukov, K. Nikolskaya, and S. Shabiev, “The capacity of the road network: Data collection and statistical analysis of traffic characteristics,” Energies (Basel), vol. 13, no. 7, p. 1765, 2020. doi.org/10.3390/en13071765.

[6] J. Evans, B. Waterson, and A. Hamilton, “Evolution and future of urban road incident detection algorithms,” J Transp Eng A Syst, vol. 146, no. 6, p. 3120001, 2020. doi.org/10.1061/JTEPBS.0000362.

[7] S. D. Khan and H. Ullah, “A survey of advances in vision-based vehicle re-identification,” Computer Vision and Image Understanding, vol. 182, pp. 50–63, 2019. doi.org/10.1016/j.cviu.2019.03.001.

[8] M. Uecker, T. Fleck, M. Pflugfelder, and J. M. Zøllner, “Analyzing Deep Learning Representations of Point Clouds for Real-Time In-Vehicle LiDAR Perception,” arXiv preprint arXiv:2210.14612, 2022. doi.org/10.48550/arXiv.2210.14612.

[9] S. Maity, A. Bhattacharyya, P. K. Singh, M. Kumar, and R. Sarkar, “Last decade in vehicle detection and classification: a comprehensive survey,” Archives of Computational Methods in Engineering, vol. 29, no. 7, pp. 5259–5296, 2022. doi.org/10.1007/s11831-022-09764-1.

[10] N. Arora and Y. Kumar, “Automatic vehicle detection system in Day and Night Mode: challenges, applications and panoramic review,” Evol Intell, vol.16, no.4, pp.1077–1095, 2023. doi.org/10.1007/s12065-022-00723-0.

[11] S. Sivaraman and M. M. Trivedi, “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis,” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773–1795, 2013. doi: 10.1109/TITS.2013.2266661.

[12] C.-C. R. Wang and J.-J. J. Lien, “Automatic vehicle detection using local features—A statistical approach,” IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, pp. 83–96, 2008. doi: 10.1109/TITS.2007.908572.

[13] N. Arora, Y. Kumar, R. Karkra, and M. Kumar, “Automatic vehicle detection system in different environment conditions using fast R-CNN,” Multimed Tools Appl, vol. 81, no. 13, pp. 18715–18735, 2022. doi.org/10.1007/s11042-022-12347-8.

[14] R. A. Hadi, G. Sulong, and L. E. George, “Vehicle detection and tracking techniques: a concise review,” arXiv preprint arXiv:1410.5894, 2014. doi.org/10.48550/arXiv.1410.5894.

[15] X. Luo et al., “Fast automatic vehicle detection in uav images using convolutional neural networks,” Remote Sens (Basel), vol. 12, no. 12, p. 1994, 2020. doi.org/10.3390/rs12121994.

[16] C.-J. Lin, S.-Y. Jeng, and H.-W. Lioa, “A real-time vehicle counting, speed estimation, and classification system based on virtual detection zone and YOLO,” Math Probl Eng, vol. 2021, pp. 1–10, 2021. doi.org/10.1155/2021/1577614.

[17] M. Majumder and C. Wilmot, “Automated vehicle counting from pre-recorded video using you only look once (YOLO) object detection model,” J Imaging, vol. 9, no. 7, p. 131, 2023. doi.org/10.3390/jimaging9070131.

[18] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, “Learning discriminative model prediction for tracking,” in Proceedings of the IEEE/CVF international conference on computer vision, pp. 6182–6191, 2019.

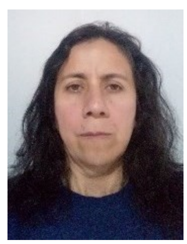
[19] Y. Fang, C. Wang, W. Yao, X. Zhao, H. Zhao, and H. Zha, “On-road vehicle tracking using part-based particle filter,” IEEE transactions on intelligent transportation systems, vol. 20, no. 12, pp. 4538–4552, 2019. doi: 10.1109/TITS.2018.2888500.

[20] Z. Soleimanitaleb, M. A. Keyvanrad, and A. Jafari, “Object tracking methods: a review,” in 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 282–288, 2019. doi: 10.1109/ICCKE48569.2019.8964761.

[21] S. Liu, D. Liu, G. Srivastava, D. Połap, and M. Woźniak, “Overview and methods of correlation filter algorithms in object tracking,” Complex & Intelligent Systems, vol. 7, pp. 1895–1917, 2021. doi.org/10.1007/s40747-020-00161-4.

[22] A. Bathija and G. Sharma, "Visual object detection and tracking using yolo and sort," International Journal of Engineering Research Technology, vol. 8, no. 11, 2019.
[23] X. Hou, Y. Wang, and L.-P. Chau, "Vehicle tracking using deep sort with low confidence track filtering," in 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–6. doi: 10.1109/AVSS.2019.8909903.
[24] J. Jiao and H. Wang, "Traffic behavior recognition from traffic videos under occlusion condition: a Kalman filter approach," Transp Res Rec, vol. 2676, no. 7, pp. 55–65, 2022. doi.org/10.1177/03611981221076426.
[25] A. Bisht, K. Ravani, M. Chaturvedi, N. Kumar, and S. Tiwari, "Indigenous design of a Traffic Light Control system responsive to the local traffic dynamics and priority vehicles," Comput Ind Eng, vol. 171, p. 108503, 2022. doi.org/10.1016/j.cie.2022.108503.
[26] P. Dave, A. Chandarana, P. Goel, and A. Ganatra, "An amalgamation of YOLOv4 and XGBoost for next-gen smart traffic management system," PeerJ Comput Sci, vol. 7, p. e586, 2021. doi.org/10.7717/peerj-cs.586.
[27] M. U. Arif, M. U. Farooq, R. H. Raza, Z. Lodhi, and M. A. R. Hashmi, "A Comprehensive Review of Vehicle Detection Techniques Under Varying Moving Cast Shadow Conditions Using Computer Vision and Deep Learning," IEEE Access, 2022. doi: 10.1109/ACCESS.2022.3208568.
[28] H. J. Jeong, K. S. Park, and Y. G. Ha, "Image Preprocessing for Efficient Training of YOLO Deep Learning Networks," in Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018, Institute of Electrical and Electronics Engineers Inc., pp. 635–637, 2018. doi: 10.1109/BigComp.2018.00113.
[29] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, "Number plate recognition on vehicle using YOLO - Darknet," in Journal of Physics: Conference Series, IOP Publishing Ltd, 2021. doi: 10.1088/1742-6596/1821/1/012049.
[30] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions," 2022. [Online]. Available: www.aaai.org. doi.org/10.1609/aaai.v36i2.20072.
[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." 2016. [Online]. Available: http://pjreddie.com/yolo/
[32] Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr., vol. 5, pp. 1680-1716, 2023. doi.org/10.3390/make5040083.
[33] J. Hurtado-Gómez, J. D. Romo, R. Salazar-Cabrera, Á. P. de la Cruz, and J. M. M. Molina, "Traffic signal control system based on intelligent transportation system and reinforcement learning," Electronics (Switzerland), vol. 10, no. 19, Oct. 2021, doi: 10.3390/electronics10192363. doi.org/10.3390/electronics10192363.
[34] B., Dwyer, J., Nelson, and T. Hansen, "Roboflow (Version 1.0) [Software]." Accessed: Mar. 25, 2024. [Online]. Available: https://roboflow.com.
[35] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R. I. Taniguchi, "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow," Sensors (Switzerland), vol. 19, no. 20, 2019, doi: 10.3390/s19204588.
[36] A. Gomaa, T. Minematsu, M. M. Abdelwahab, M. Abo-Zahhad, and R. ichiro Taniguchi, "Faster CNN-based vehicle detection and counting strategy for fixed camera scenes," Multimed Tools Appl, vol. 81, no. 18, pp. 25443–25471, 2022, doi: 10.1007/s11042-022-12370-9.
[37] V. O. Castelló, I. S. Igual, O. D. T. Catalá, and J. C. Perez-Cortes, "High-Profile VRU Detection on Resource-Constrained Hardware Using YOLOv3/v4 on BDD100K," J Imaging, vol. 6, no. 12, 2020, doi: 10.3390/jimaging6120142.
[38] G. Wieczorek, S. B. ud din Tahir, I. Akhter, and J. Kurek, "Vehicle Detection and Recognition Approach in Multi-Scale Traffic Monitoring System via Graph-Based Data Optimization," Sensors, vol. 23, no. 3, 2023, doi: 10.3390/s23031731.
[39] G. Doménech-Asensi, M. D. Cano, and V. Morales-Esteras, "On the Use of Bayesian Networks for Real-Time Urban Traffic Measurements: a Case Study with Low-Cost Devices," J Signal Process Syst, vol. 94, no. 3, pp. 293–304, 2022, doi: 10.1007/s11265-020-01601-7.
[40] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," European Transport Research Review, vol. 11, no. 1, 2019, doi: 10.1186/s12544-019-0390-4.

Daniel Valencia received the B.Sc. degree in Physical engineering in 2016 and Industrial Automation Engineering in 2018, from Universidad del Cauca - Popayán, Cauca, Colombia, has developed works in the areas of control, mathematical modelling computational vision, and discrete events systems.



Elena Muñoz España received the engineering degree in electronics and telecommunications engineering, and master's degree in electronics engineering from Universidad del Cauca. She is currently working as a full professor at the University of Cauca, Colombia. Her research interests include control systems, computer vision and medical image processing.

Mariela Muñoz received the B.Sc. degree in industrial engineering from Universidad Tecnológica de Pereira, Colombia, the M.Sc. degree in Automatic from Universidad del Cauca, Colombia, and the PhD degree in Automatic, robotic and informatics industrial from Universidad Politécnica de Valencia, España, 2015. She is currently a professor in Universidad del Cauca. Her research interests include discrete event systems, and fault diagnosis.