Counting Vehicle with High-Precision in Brazilian Roads Using YOLOv3 and Deep SORT

Adson M. Santos, Carmelo J. A. Bastos-Filho, Alexandre M. A. Maciel, Estanislau Lima
Polytechnic School of Pernambuco - University of Pernambuco
Recife, Brazil - 52.720-001
{ams2, carmelofilho, amam, ebl2}@ecomp.poli.br

***Abstract*—The Brazilian National Department of Transport Infrastructure (DNIT) maintains the National Traffic Counting Plan (PNCT). The main goal of PNCT is to evaluate the current flow of traffic on federal highways aiming to define public policies. However, DNIT still performs the quantitative classificatory surveys not automated or with invasive equipment. It is crucial for conducting traffic studies to search for more modern solutions to accomplish a higher number of automated non-invasive, and low-cost classificatory surveys. This paper proposes a system that uses YOLOv3 for object detection and the Deep SORT for multiple objects tracking algorithms. From the results over real-world videos collected in Brazilian roads, we obtained a precision above 90 % in the global vehicle count. We also show that our proposal outperformed other previously proposed tools with 99.15% precision in public datasets. We believe this paper's proposal allows the development of a traffic analysis tool to be used for the automation of the volumetric traffic surveys, enabling to improve the DNIT agility and generating economy for the public coffers.**

I. Introduction

The knowledge regarding traffic distribution in highway traffic surveys is fundamental to predict the future traffic needs of road users, to evaluate the solutions adopted on the pavement, and to create the basis for road planning. Some of the traffic studies instruments are global volumetric and classification surveys, which allow the study of the quantitative of vehicles globally or by class [1].

Currently, the DNIT (National Department of Transport Infrastructure) of Brazil maintains the National Traffic Counting Plan (PNCT) [2]. This plan performs the quantitative classificatory surveys in specific points of the Brazilian federal highways. DNIT uses invasive and non-invasive solutions (without embedding in the pavement). DNIT deploys equipment consisting of piezoelectric sensors and inductive loop detectors placed inside the pavement in the invasive solution. In the non-invasive solution, DNIT counts and classifies vehicles using microwave technology. The microwave-based solution only assesses the vehicle's length, thus requiring the manual classification of the types of vehicles. Summarizing, DNIT still conducts classificatory vehicle surveys not automated, generating high operational costs, or with invasive equipment, with high installation costs [2]. Therefore, it is mandatory to search for more modern solutions to accomplish an automated non-invasive, and low-cost classificatory survey.

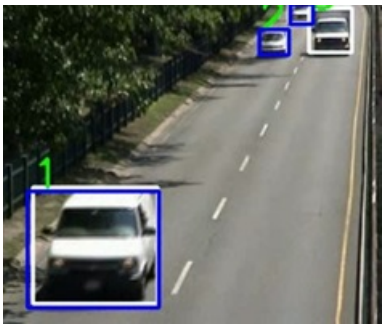


Fig. 1. Application of the proposed model, YOLOv3 and Deep SORT, to perform the global vehicle count in the CD2014 dataset.

In this context, researchers have described methods of engaging computer vision models for the vehicle counting process. For instance, Yang et al. [3] proposed a vehicle detection based on background subtraction. They deployed a motion-based detection method with the low-rank decomposition technique. It presents promising results for static scenarios, but in the situations where the background changes significantly, the accuracy in the detection process diminishes. Also, the vehicle counting process is still challenging, and it is necessary to deal with partial occlusion of the objects and variation of illumination of the images. For this, we need to perform accurate object detection for later tracking.

In the last few years, Deep Learning methods (DL), more precisely Convolutional Neural Networks (CNNs) [4], have demonstrated dramatic improvement over traditional methods for many computer vision tasks, such as image classification, object detection, pattern recognition, and many others. Thus, CNN's are a possible solution for appearance-based vehicle detection models, which already stands out in video and image processing [5]. In the vehicle counting process, CNN helps us improve object detection, besides allowing vehicles' classification. In this regard, Abdelwahab [6] proposed an efficient approach to global vehicle counting employing Regions with Convolutional Neural Networks (R-CNN) and the KLT (Kanade-Lucas-Tomasi) tracker.

The R-CNN model obtained some promising results with KLT or other deep based models [6]. However, the application of deep learning methods for vehicle counting continues to be

a challenging task for the implementation in many real-world applications. The application in real-world platforms is even more challenging when the application is limited by hardware. It can occur in the cases in which robots and smartphones are deployed. In this context, Redmon introduced the YOLOv3 model, which enables real-time detection together with high precision [7].

In this paper, we propose a system to detect, track, and count vehicles composed by the detector YOLOv3 (You One Look Once algorithm version 3) [7] and the tracker Deep SORT (Simple Online Realtime Tracking) [8]. The first goal is to improve the precision regarding the Abdelwahab's [6] and Yang's model [3]. We selected YOLOv3 and Deep SORT since they presented promising results in some related video counting problems with people [9]. In Fig. 1, we present one example of a global volumetric count performed by the model proposed in this work. This proposal reached a precision of 99.15% in the GRAM dataset [10] and the CD2014 dataset [11]. In a DNIT dataset, we have longer videos with a duration of 30 minutes each. In this second scenario, we applied the proposed model with the optimal hyperparameters found in this work, allowing us to reach rates above 90%.

Therefore, this article addresses the problems and solutions that exist in the area of traffic research, especially volumetric surveys of global counts. In this sense, this work aims to validate the YOLOv3 and DeepSORT models for the global vehicle counting problem. The main contributions of this article are: (1) Improve the counting accuracy concerning other works in the area of global vehicle counting; (2) Present a study of the hyperparameters of the proposed model concerning vehicle counting accuracy; (3) Assess the accuracy of YOLOv3 and Deep SORT in a real scenario on the Brazilian federal highways of the DNIT dataset. Finally, in addition to presenting a high precision method in the global vehicle count, this work presents a study of the YOLOv3 and DeepSORT models' performance in the global vehicle counting problem. Thus, in the future, we can apply this proposed model to perform vehicle counting by classes at DNIT, which is still done not automated with expensive non-invasive methods.

This paper is organized as follows. In Section 2, we present related works of vehicle counting, detection, and multiple object tracking. In Section 3, we introduce our proposal. In Section 4, we have experimental methods. In Section 5, we show the analysis and discussion of the results. In Section 6, we give our conclusions and outlook.

II. Background

In this section, we present the vehicle counting-related works based on image background subtraction like low-rank decomposition [3] and deep learning models like R-CNN with KLT tracker [6]. We also present highlights models in the area of object detection with YOLOv3. Besides, we present the algorithm Multiple Object Tracking models with Deep SORT.

A. Vehicle Count

Vehicle counting depends simultaneously on detection and tracking. We can divide the detection models into detection based on appearance and motion. Appearance-based detection uses feature extraction like color, corner points, and edges. On the other hand, we have motion-based methods with frame difference, background subtraction, and other methods [12].

In Yang et al. [3], the video frames are decomposed into the foreground and background using the background subtraction method. The background is separated by the low-rank component while moving objects and noisy stay in the foreground. After that, they track moving objects using a Kalman filter.

Although the works related to vehicle counting based on background subtraction performs well, this counting can be difficult if the background has a lot of movement, which implies more noise [6].

To solve this problem, we used deep learning models. In this regard, Abdelwahab [6] proposed employing Regions with Convolutional Neural Networks (R-CNN) and the KLT tracker. The R-CNN suggest regions and classifying these region proposals with feature extracted from them. In the tracker, the vehicles are counted every "*n*" frames to decrease the time complexity. The authors used "*n*" equal to 15 in the experiments. Besides that, he extracted it by tracking corner points through the observed "*n*" frames. The proposed model then assigns unique vehicle labels to their corresponding trajectories [6]. With his model applied in the public bases GRAM and CD2014, Abdelwahab achieved an accuracy of 96.44%.

B. Object Detection with YOLO

The YOLO (You Only Look Once) is one of the most commonly used computer vision models for real-time object detection and classification. The third version of the algorithm (YOLOv3) is useful for human detection and counting [9]. Thus, we used YOLOv3 in vehicle detection to validate our proposed model for the vehicle counting problem.

Generally, object detectors based on a two-stage approach popularized by R-CNN are more accurate. This Faster R-CNN [13] is one of the best-reported versions [14]. In contrast, single-stage detectors can be faster and more straightforward, such as YOLO, SSD, and RetinaNet. In this sense, one detector that has stood out is the RetinaNet, capable of exceeding the accuracy of all two-stage detectors reported in [14]. Although RetinaNet has good accuracy, it takes 3.8 times longer to process an image than YOLOv3 [7]. YOLOv3 can reach up to 45 frames per second on an Nvidia Titan X, allowing real-time prediction. YOLOv3 has accuracy above the SSD (Single Shot Detector), being up to 3 times faster. The YOLOv3 uses as a backbone of the Darknet-53, which is a network architecture with 53 convolutional layers [7], according to Fig. 2.

The network receives an image as input and divides it into a $S \times S$ dimension grid. If any object of interest is inside one of this matrix's cells, this cell is responsible for the object detection [15]. Each cell in the matrix predicts "*B*" bounding boxes and a YOLO Score. The Confidence Score

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 2. Darknet-53 convolutional network used by YOLOv3 to perform vehicle detection [7].

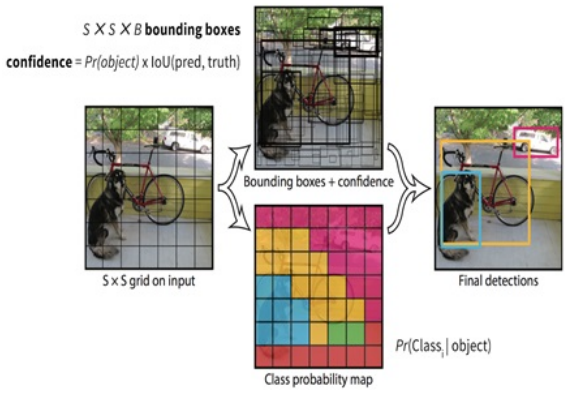


Fig. 3. The YOLO model divides the image into an $S \times S$ grid and for each cell of the grid it provides B bounding boxes and the Confidence Score for those boxes [16].

reflects the confidence of the model to detect a real object within that bounding box. The Confidence Score is defined as the multiplication of Prob (Object) by IoU, where *Prob (Object)* is the probability of an object within the bounding box, and *IoU* is the intersection on the union of the bounding boxes provided with the ground truth [15]. Fig. 3 shows the detection process, and we present the YOLO score in Equation (1).

$$\textit{YOLO Confidence Score} = \textit{Prob (Object)} * \textit{IoU (pred, truth)} \quad (1)$$

Thus, a bounding box without an object must have a Confidence Score equal to zero. Otherwise, this index is the same intersection over the union between the bounding box present in the ground truth and the predicted. As the final output, the network produces a tensor composed of the class, Confidence Score, and the coordinates of the predicted bounding box [15].

C. Multiple Object Tracking

According to Dadhich [17], Multiple Object Tracking (MOT) can be performed in two steps: Detection and Association. One of the usual models to track multiple objects, especially for people, is Deep Simple Online and Real-time Tracking (Deep SORT). The main difference is the addition of feature association metrics to improve performance relative to SORT [18] using a CNN. Because of this extension, the Deep SORT can track objects through more prolonged periods of occlusion, effectively reducing the number of Identity Switches. According to Dadhich [17], one can define the Deep SORT in three stages:

a) CNN based object detection (for example, Faster-RCNN or YOLO), used for initial detection in frames;

b) The intermediate step consists of the association of data of an estimated model. This algorithm uses a vector of eight parameters center (x, y), width (l), height (h) and it is derivatives of velocities (x ', y', l ', h') to perform tracking; the Kalman filter is used to model these states as a dynamic system;

c) From the predicted states of the Kalman filtering, we associate the new detection with old object tracks in the previous frame. This association is calculated using the Hungarian Algorithm with an association metric that measures the bounding box's overlap after "n" frames.

III. The Proposed Approach

Although Yang et al. and Abdelwahab show promising results, we can improve that result using prominent real-time object detection models like YOLOv3 and robust tracking models like Deep SORT. Both models have already been shown to be useful in the problem of counting people [9]. In this work, we validate this model in the vehicle counting problem.

In this context, the preliminary results show that the YOLO Confidence Score influences the level of requirement to detect a vehicle, consequently influencing the counting accuracy. This is an essential hyperparameter for designing vehicle counting solutions.

In the tracking step, we saw that the Deep SORT has several metrics to perform its function. However, the use of features to compare objects between frames is a differential. In this work, our results show that its CNN weights provided in Wojke [8] worked well in vehicle counting, despite having been trained to track people. However, we verified that Deep SORT is sensitive to the hyperparameter "number of frames" for the association, and confirmation of a supposed track is the Association K (aK), which was the object of study in this article. For example, all the detection sent by YOLO is considered a supposed track that needs to be confirmed by a consecutive association between frames. If the supposed track is confirmed after aK frames, the system counts this track.

Therefore, we propose a system that uses the Deep SORT with the YOLOv3 to perform the automatic counting of

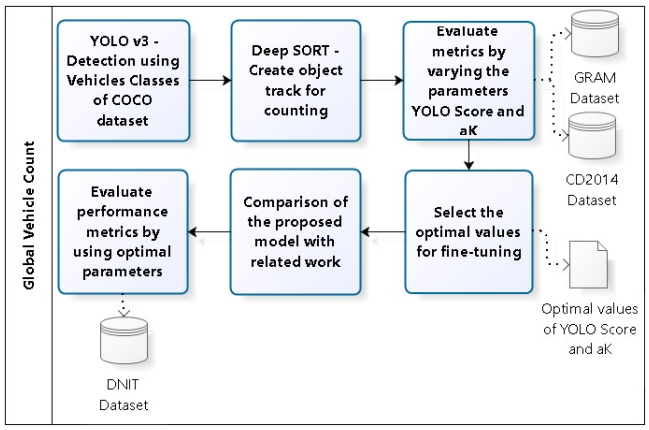


Fig. 4. Method to evaluate the proposed model of global vehicle count through analysis and selection of optimal hyperparameters, comparison with related works and application in the real scenario on Brazilian highways.

vehicles in videos. We evaluate and analyze the values of the hyperparameters YOLO Score and Association *K* in the same dataset as Abdelwahab [6] and Yang [3]. After that, the optimal values of these hyperparameters are selected to compare (Experiment 1). Selecting these hyperparameters, we fine-tuning the model to improve accuracy. Finally, we evaluate the solution's performance in a real-world Brazilian DNIT dataset (Experiment 2). Fig. 4 shows the step-by-step process.

IV. Materials and Methods

This section presents the experimental arrangement, the metrics used in experiments, and the dataset description.

A. Experimental Arrangement

We used the videos of the GRAM dataset [10] and the CD2014 dataset [11] to allow us to compare the proposed model with the Abdelwahab model [6] and the Yang model [3].

Also, certain Regions of Interest (ROI) in the dataset used were stipulated, eliminating the Bounding Boxes generated at 5% of the frame size near the edge. Especially for the M-30 and M-30HD videos, we set 20% of frame size near the top of the image because our ROI needs to be higher than the author dataset's ROI. This ROI prevents the proposed model from creating tracks for parts of the vehicle entering the scene, improving accuracy counting.

In the experiment, we use two hyperparameters: the YOLO Confidence Score in the detection step and the Deep SORT Association *K* (*aK*) to generate a track of a vehicle. In this case, we assessed the YOLO Confidence Score with values from 0.5 to 0.9. As seen in Equation (1), the YOLO Confidence Score is given as the product of *Prob (Object)* and *IoU*. We note that *IoU* is a relevant metric for this purpose, to be weighted by *Prob (Object)*. In the tracking step, we assessed the Deep SORT with the values of 5 to 15 frames to confirm a supposed track and, consequently, the count.

Furthermore, we deployed the YOLOv3 implementation with the weights available on the official Redmon website [7] using the COCO dataset. The main hyperparameters used by Redmon in the YOLOv3 configuration are shown in Table I. As well, only the following vehicle classes were selected: truck, car, motorcycle, and bus. Besides that, we use the default Deep SORT [8] by varying only the Association *K* hyperparameter to confirm a tracking.

TABLE I
Main hyperparameters used by Redmon in the YOLOv3 [7].

Parameter	Value	Description
img_size	416 x 416	Input image size
learning_rate	1 10-3	Initial learning rate
batch	64	Batch size in each training step
max_batches	500200	Maximum number of iterations
optimizer	SGD	Stochastic Gradient Descent
loss	yolo_loss	Loss function to optimize

For the best of our knowledge, the source codes for the Abdelwahab [6] and Yang [3] algorithms are not available for further experiments in any repository. Thus, we compared our results with the ones already published by the authors, but we could not compare our results with these previous approaches in the DNIT dataset.

In our experiments, we used a notebook with the following specifications: Intel Core i5-8300H, NVIDIA GeForce GTX 1050 with 4GB of GDDR5, and 8GB DDR4 RAM.

B. Deployed metrics

The results evaluation was based on some metrics used for multiple object tracking (MOT). However, we made some updates to reflect the counting goal rather than tracking the object throughout a lifetime.

In this sense, the metrics tracked and lost did not have their percentages considered when compared to the metrics used by Wojke [8]. As long as the algorithm counts only one object, it is not a problem if the object is accompanied by 100% or 50% of the life cycle (number of frames in which the object appears). The metrics used in this work were:

- Precision: it is the result of the empirical count (True No. or Ground truth) subtracted from the error calculated in the interval [6] represented in Equations (2) and (3) below.

$$Precision(\%) = 100 - Error(\%) \quad (2)$$

$$Error(\%) = \frac{|Estimated - TrueNo|}{TrueNo} X100 \quad (3)$$

- True Positives or Tracked (TP): the number of vehicles tracked. That is when the object was tracked at some point in its useful life.
- False Positives (FP): number of false detections, this is, "false alarm". In our work, an example would be counting a pedestrian or a bicycle.
- False Negatives or Lost Tracked (FN): the number of trajectories lost. That is, the target is not tracked in its useful life.
- True Negatives (TN): number of false trajectories not tracked.

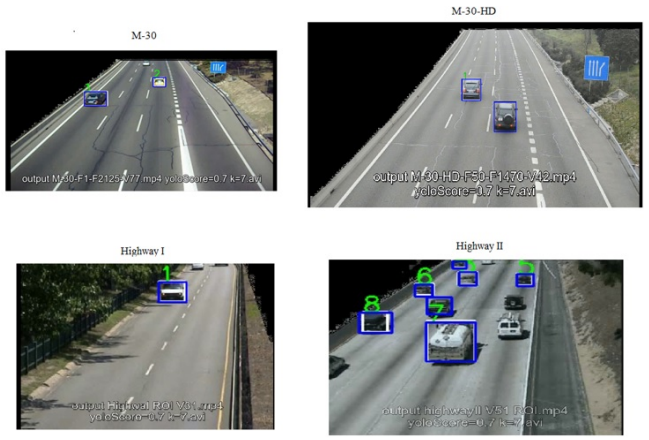


Fig. 5. Examples of frames from public Datasets CD2014 and GRAM show the vehicle count by the proposed model. We can see the different characteristics of the resolution, climate, and scenery of each video.

- ID switches (ID sw): number of times the reported identity of a true path (Ground truth) changes. Therefore, this means a double count if two IDs (tracks) are selected for the same object. Alternatively, a count loss if the same ID (track) is applied to two objects at different times.
- Frames Per Second (FPS): The frequency at which consecutive images are processing.

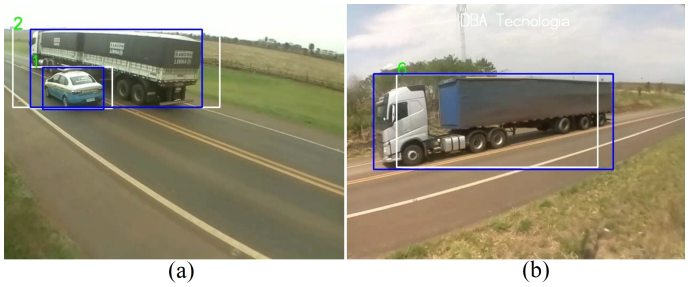
C. Characteristics of the Datasets

In experiment 1, we used Highway I and Highway II, taken from the CD2014 dataset, which has some challenges, such as shaking trees and shadows. Highway II is full of vehicles and has low image quality. In the GRAM dataset, we evaluated the M-30 and M-30 HD. The M-30 represents a sunny video, while the M-30 HD represents a cloudy video, and both are road videos. In Fig. 5, we can see some frames of these videos with the application of the proposed model.

In Experiment 2, we chose two videos with 640x480 resolution in different highway stretches and different camera perspectives, as shown in Figure 6. These videos have 30 minutes, more extensive than those used in the GRAM dataset [10] and the CD2014 dataset [11], that have equal to or less than 5 minutes long. In the DNIT dataset, it is used the lateral perspective instead of the top perspective. The lateral perspective brings more regions of occlusion. All the characteristics of the videos used in experiments 1 and 2 can be summarized in Table II, which includes the True Number of vehicles and video length.

V. Analysis and Discussion

This section presents the evaluation of the metrics by varying the hyperparameters YOLO score and Association *K*. After selecting these hyperparameter values for fine-tuning, we performed the first experiment that compares the results with related works. The second experiment evaluates the model proposed in the DNIT dataset.



(a) (b)

Fig. 6. Brazilian highway, BR60 I in (a) and BR60 II in (b) maintained by DNIT counted by the proposed model.

TABLE II
Information of the videos used in Experiment 1 and 2.

Dataset	Name	Resolution	True No	Length	Description
CD 2014	Highway I	240x320	30	00:01:53	Having trees and shadows.
CD 2014	Highway II	240x320	48	00:00:33	Full of vehicles and poor image quality.
GRAM	M-30	480x800	77	00:02:21	Sunny.
GRAM	M-30 HD	720x1200	42	00:01:34	Cloudy.
DNIT	BR60 I	480x640	226	00:30:01	Cloudy, many occlusions.
DNIT	BR60 II	480x640	74	00:30:01	Sunny, trees and many occlusions.

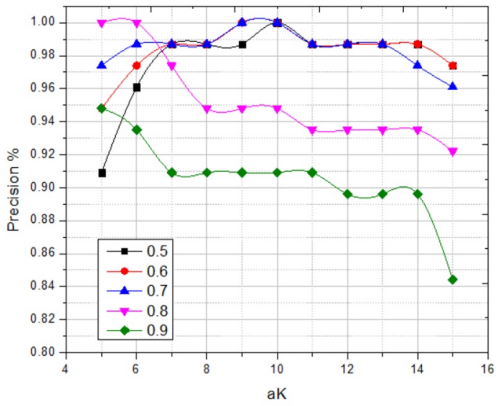


Fig. 7. Accuracy of the proposed model as a function of the YOLO score from 0.5 to 0.9 and the *aK* from 5 to 15 in the **M-30** video.

A. Evaluating the metrics varying the hyperparameters

Figures 7, 8, 9, and 10 show the global vehicle counting accuracy by varying the YOLO Confidence Score and Association *K* (*aK*) hyperparameters of the videos M-30, M-30 HD, Highway I, Highway II, respectively.

Fig. 7 shows the accuracy of the medium resolution M-30 video when using the YOLO Score equal to 0.7. We notice that the obtained accuracy in *aK* equal to 5 is 97%. For *aK* equal to 10, the accuracy increases to 100%. In *aK* equal to 15

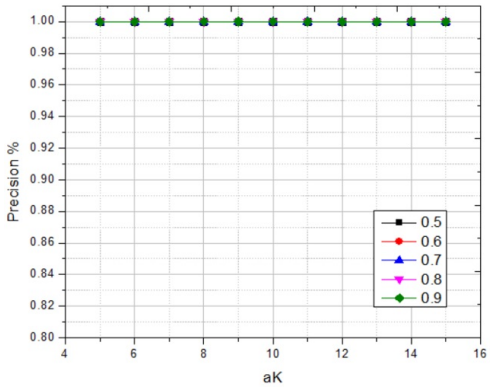


Fig. 8. Accuracy of the proposed model as a function of the YOLO score from 0.5 to 0.9 and the aK from 5 to 15 in the **M-30 HD** video.

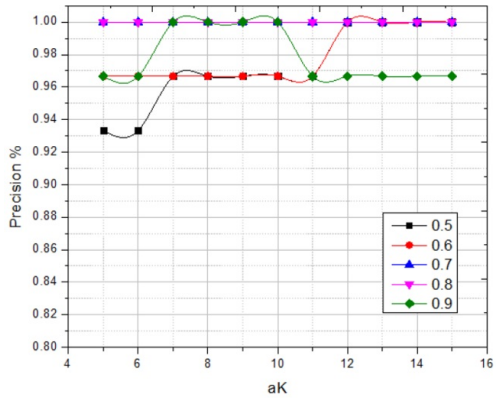


Fig. 9. Accuracy of the proposed model as a function of the YOLO score from 0.5 to 0.9 and the aK from 5 to 15 in the **Highway I** video.

it falls back to 96%. We observed similar behavior for YOLO Score below 0.7 in this video. In this case, the number of ID Switch and False Positives is higher for aK below 10.

Also, in the M-30 video, the accuracy decreases with the increase in aK for YOLO Score equal to or above 0.8. Some vehicles in specific frames do not reach the threshold necessary to activate the detection. If an object is not detected, we stop the tracking. Therefore, the proposed model becomes more demanding to create a track and start counting, and this can generate False Negatives.

Fig. 8 shows the precision of the M-30 HD video with high resolution and noiseless. The proposed model obtained 100% performance in all the measured values of the hyperparameters YOLO Score and aK.

Fig. 9 depicts the precision for the Highway I video. This video presents low resolution and reasonable quality. The proposed model obtained 100% precision for the YOLO Score equal to 0.7 and 0.8. However, for YOLO Score below 0.7, the number of ID Switch and False Positives has increased. For YOLO Score equal to 0.9, the model generated False Negatives.

Fig. 10 shows the precision of the low-resolution, low-quality Highway II video. The proposed model obtained better

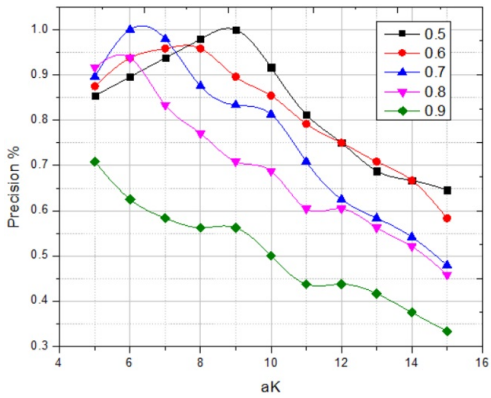


Fig. 10. Accuracy of the proposed model as a function of the YOLO score from 0.5 to 0.9 and the aK from 5 to 15 in the **Highway II** video.

results only for YOLO Score around to 0.7 and aK around 7. Due to the low quality of the images, many vehicles did not reach the required YOLO Score threshold to activate detection.

Therefore, an increase in double vehicle count (ID Switch) and False Positives can be seen by using Association K (aK) and YOLO Confidence Score values lower than the optimal hyperparameters. In case of a decrease of aK, fewer frames are used to generate a track. This makes the model less demanding to count a given vehicle. For the low YOLO Score, more than one Bounding Box may satisfy the established low confidence level. These Bounding Boxes also favors double counting.

On the other hand, when the values of Association K and YOLO Score are higher than the optimal hyperparameters, the model becomes more demanding for the creation of a track and detection of a vehicle. In this case, there may be a loss of vehicles in the count (False Negative). Therefore, it is essential to select the hyperparameters to allow the proposed model to achieve the best counting accuracy results.

B. Selecting the hyperparameters values for fine-tuning

After the graphical analysis of the proposed hyperparameters, we observed promising results in the four videos analyzed for YOLO Score values around 0.7 and Association K around to 7. Therefore, these are the values chosen to set the hyperparameters for comparing the results with other works.

Table III presents a detailed analysis using YOLO Score equal to 0.7 and aK equal to 7. One can verify the best accuracy of the proposed model in counting vehicles in the M-30-HD scenario, already shown in Fig. 8 that has a sharpness and high resolution of the image. However, from Table III, we observed that the Highway II video has high rates of double count (ID Switches) and False Negatives. In this video, the low resolution and a large amount of noise can cause the low result of the FN and ID Sw metrics.

C. Comparison of Results with related works

In Table IV, we show the results of the comparison between the proposed model (YOLOv3 and Deep SORT) with the related works. Our approach achieved 99.15% of precision in the global vehicle count. This value is 2.71% and 9.12%

TABLE III
DETAILED ANALYSIS WITH OPTIMAL HYPERPARAMETERS YOLO SCORE EQUAL TO 0.7 AND *aK* EQUAL TO 7 - EXPERIMENT 1.

Video Name	M-30	M-30 HD	Highway I	Highway II
True No.	77	42	30	48
Count	78	42	30	47
Precision (%) [↑]	98.7	100	100	97.92
TP [↑]	78	43	30	47
FP [↓]	0	0	0	0
FN [↓]	0	0	0	5
TN [↑]	0	0	0	0
ID Sw [↓]	1	0	0	4
FPS [↑]	5.67	5.13	5.91	6.03

TABLE IV
COMPARISON OF THE PROPOSED MODEL WITH ABDELWAHAB [6] AND YANG [3].

	Video Name	M-30	M-30 HD	High way I	High way II	**Average**
Yang et al.	True No.	77	42	16	91	
	Count	71	37	14	84	
	Precision (%)	**92.21**	**88.1**	**87.5**	**92.31**	**90.03**
Abdel-wahab	True No.	77	42	28	48	
	Count	72	42	27	46	
	Precision (%)	**93.51**	**100**	**96.43**	**95.83**	**96.44**
Proposed Model	True No.	77	42	30	48	
	Count	78	42	30	47	
	Precision (%)	**98.7**	**100**	**100**	**97.92**	**99.15**

higher than the approaches proposed by Abdelwahab [6] and Yang [3], respectively. To summarize Table IV, we depict the results in Fig. 11.

D. Evaluating the proposed model in the DNIT dataset

Although the proposed model has reached a promising result regarding other literature approaches, it is necessary to evaluate this model in a more complex scenario. The dataset was the same used in DNIT's daily traffic surveys.

When evaluating our solution in a real-world scenario, it was possible to verify better the behavior of True Negatives and unusual vehicles like the one depicted in Fig. 12. In this case, the YOLO detected two vehicles, but the Deep SORT only generated one track. Therefore, the two vehicles were counted only one. When analyzing the precision in the DNIT dataset with the optimal hyperparameters, YOLO Score at 0.7 and Association *K* at 7, we found rates above 90%, according to Table V.

Another relevant point in the vehicle count process in the DNIT dataset was the occurrence of the ID Switch in two

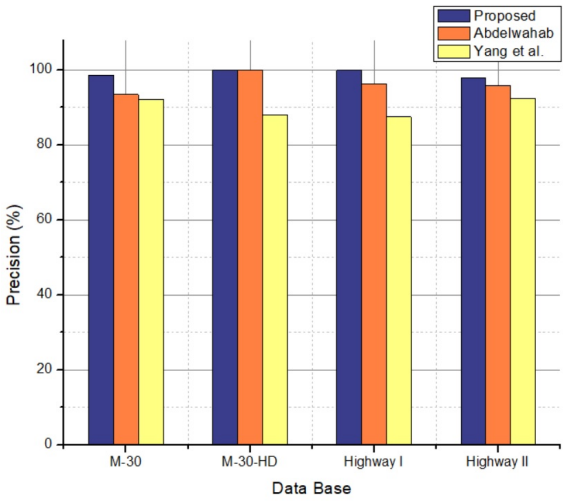


Fig. 11. Video accuracy among comparative models.

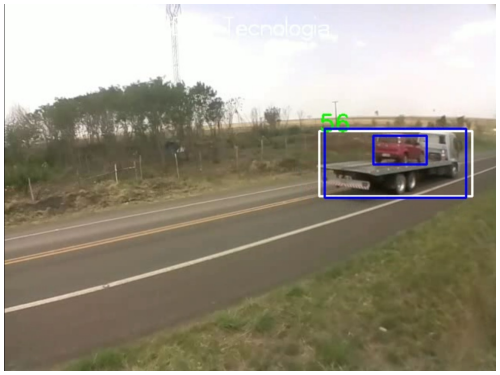


Fig. 12. Unusual vehicles, counted by the model proposed in the BR 60 II.

TABLE V
APPLICATION OF THE PROPOSED MODEL IN THE DNIT DATASET -EXPERIMENT 2.

Video Name	BR60 I	BR60 II
True No.	226	74
Count	209	67
Precision (%)[↑]	92.47	90.54
TP [↑]	226	74
FP [↓]	0	0
FN [↓]	16	9
TN [↑]	3	0
ID Sw [↓]	1	2
FPS [↑]	6.03	6.31

possible types. The BR60 I video ID Switch refers to the missing track, and the BR 60 II ID refers to the double count.

Therefore, we considered the estimations as satisfactory for most of DNIT needs [1]. We can also highlight the robustness of the model that did not include any element other than vehicles. In other words, there were no false positives in the assessed videos.

VI. Conclusions

To allow the traffic researches in Brazilian roads using deep learning, we present a study of the YOLOv3 and Deep SORT models to perform global vehicle detection and counting. The main goal is to automatize volumetric surveys in DNIT in a non-invasive and low-cost manner. To do this, we improved the counting accuracy concerning other works in global vehicle counting. Our proposal allows reaching an average accuracy of 99.15% in the global vehicle count in the GRAM dataset [10] and the CD2014 dataset [11]. This value is 2.71% and 9.12% higher than the approaches proposed by Abdelwahab [6] and Yang [3], respectively.

Besides this, we show an analysis of the hyperparameters of the proposed model concerning vehicle counting accuracy. We observed that accuracy could improve a lot with this fine-tuning of the hyperparameters. In the Deep SORT, we present the hyperparameter "number of frames" for association and creation of a track, Association K (aK). Another hyperparameter that influences the counting accuracy is the YOLO Score, which indicates the probability that the bounding box contains an object. In this sense, we verified that the low Association K and YOLO Score values, for optimal hyperparameters, resulted in less accuracy in the overall score. This result is mainly related to the increase in the ID Switch (e.g., double count). However, for the values of Association K and Confidence YOLO Score high, there is a loss of vehicles in the count (False Negative).

After the graphical analysis of the results, we vary the YOLO Score and the aK in the GRAM dataset [10] and the CD2014 dataset [11]. We found optimal value for the YOLO Score equal to 0.7, i.e., 70% confidence. Moreover, the value of aK equal to 7 was selected for creating a track. Besides this, we validated the accuracy of YOLOv3 and Deep SORT with the selected optimal hyperparameters, which reached an accuracy above 90% in a real scenario of Brazilian federal highways.

Therefore, this work showed an improvement in the count related works, in addition to presenting a study of essential hyperparameters and metrics for those who intend to use YOLOv3 and Deep SORT to perform vehicle counting. This study can be useful for several entities that need to carry out traffic studies in their countries.

In the future, The validation of the proposed model in the global counting opens the way for DNIT to perform the classificatory counting through YOLOv3 training in its DNIT vehicle class table. In this context, we intended to create a dataset to train the proposed model to perform the classificatory count according to DNIT PNCT vehicle classes based on vehicle axles. We believe this system will allow the vehicle count by class automatically, bringing higher speed in obtaining traffic information and lower costs for the federal government.

Acknowledgment

We appreciate the support of DNIT and UPE. Besides that, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] DNIT, *Manual de estudos de tráfego*, 384th ed., DNIT, Rio de Janeiro, 2006.

[2] ——, "Plano nacional de contagem de tráfego (pnct)," available: http://www.dnit.gov.br/pnct. [Accessed: April 13, 2020].

[3] H. Yang and S. Qu, "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 75–85, 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] D. C. Zanotta, M. P. Ferreira, and M. Zortea, *Processamento de imagens de satélite*. Oficina de Textos, 2019.

[6] M. A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*. IEEE, 2019, pp. 1–5.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[9] H. Nakashima, I. Arai, and K. Fujikawa, "Passenger counter based on random forest regressor using drive recorder and sensors in buses," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 561–566.

[10] R. Guerrero-Gómez-Olmedo, R. J. López-Sastre, S. Maldonado-Bascón, and A. Fernández-Caballero, "Vehicle tracking by simultaneous detection and viewpoint estimation," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 2013, pp. 306–316.

[11] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 387–394.

[12] 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.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[15] A. A. dos SANTOS, S. AVILA, and T. T. SANTOS, "Detecção automática de uvas e folhas em viticultura com uma rede neural yolov2." in *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*. In: Congresso Interinstitucional de Iniciação Científica, 12., 2018 , 2018.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[17] A. Dadhich, *Practical Computer Vision: Extract Insightful Information from Images Using TensorFlow, Keras, and OpenCV*. Packt Publishing Ltd, 2018.

[18] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.