ROMAI J., v.20, no.2(2024), 1–16

# MULTIPLE VEHICLE DETECTION AND INFORMATION EXTRACTION FRAMEWORK USING MACHINE AND DEEP LEARNING VIDEO ANALYSIS SOLUTIONS

Tudor Barbu[1], Silviu-Ioan Bejinariu[1], Costică Moroşanu[2], Ramona Luca[1]
[1] *Institute of Computer Science of the Romanian Academy, Iaşi Branch*
[2] *"Alexandru Ioan Cuza" University of Iaşi, Iaşi, Romania*
tudor.barbu@iit.academiaromana-is.ro, silviu.bejinariu@iit.academiaromana-is.ro,
costica.morosanu@uaic.ro, ramona.luca@iit.academiaromana-is.ro

**Abstract** A computer vision technique that detects the vehicles in the traffic videos and extract their main information by using novel machine and deep learning solutions is proposed in this research work. It applies a transfer learning-based multiple vehicle detection scheme that builds train and validate a YOLO deep network on a voluminous vehicle database developed by us. An active contour-based segmentation is introduced to improve the detection process. Then, one identifies the most important information of the detected vehicles: trajectory, type, license plate, number and logo. The trajectory of each detected vehicle is determined by applying a motion-based tracking approach that is based on a Kalman filtering algorithm. The type of each vehicle is identified next by applying a novel convolutional neural network (CNN)-based recognizer that is built by modifying a GoogLeNet-based classifier. An ensemble learningbased vehicle license plate detection technique is then introduced here. It uses a boosting cascade classifier with Haar features that is trained on a traffic dataset prepared by us. Next, the vehicle numbers on these license plates are recognized applying some morphological operations to extract the characters and a template matching process on these obtained alpha-numericals. Then, the vehicle logos are located by training, validating and testing another YOLO-based deep learning model and applying that obtained CNN on the detected vehicles.

**Keywords:** vehicle detection, transfer learning-based detector, active contours, motion-based vehicle tracking, Kalman filter, CNN-based vehicle classification, cascade classifier-based license plate detection, vehicle number extraction, deep learning-based logo detection.
**2020 MSC:** 35Axx, 35Q68, 65D19, 68Txx, 68T07.

## 1. INTRODUCTION

The intelligent vehicle monitoring represents an essential subdomain of the traffic management field [1]. The vehicle monitoring solutions are crucial for numerous applications, like intelligent traffic control, intelligent transportation systems, traffic surveillance, public safety and security, playing an important role in preventing the traffic jams and road accidents, and identifying the traffic rule violations [2, 3]. These vehicle information detection techniques are also very helpful for various traffic-related projects, such as those aiming at the road widening, traffic signal timing and parking space construction.

These monitoring techniques detect multiple vehicles in the frames of the traffic videos and also some important information of them, such as the types, trajectories, license plates, numbers and logos of those vehicles, by applying various artificial intelligence (AI) and computer vision (CV)-based solutions. The existing machine and deep learning-based methods developed for these tasks, which are discussed in the next sections, deal with a variety of challenges such as the camera motion, deformation, motion blur, variations in brightness and illumination and vehicle occlusions.

The research described here is part of a pedestrian and traffic monitoring project that is conducted by us. It introduces an AI-based framework for multiple vehicle detection and information extraction, which uses some novel machine learning algorithms and deep neural networks in combination to mathematical models.

The vehicle detection component of this framework is described in the following section. It applies a novel transfer learning-based YOLO v5 detector that is built by modifying a MobileNet model and is trained and validated on a voluminous annotated vehicle database that has been developed in this project. Its detection results are improved by a nonlinear partial differential equation (PDE)-based geometric active contour segmentation approach that is proposed here. The main information of the detected vehicles are extracted next. Thus, the vehicles tracks are determined by a motion-based vehicle tracking (counting) technique using a Kalman filtering process, which is proposed in the third section.

Next, the detected and tracked vehicles are classified by their type, as cars, buses, trucks, trams or (moto)bikes, by applying another transfer-learning approach that is presented in the fourth section. It builds a new CNN-based vehicle classifier by modifying an existing GoogleNet model. Then, the obtained deep classification network is trained, validated and tested successfully on a vehicle dataset. The classification results are improved by using the previously determined vehicle tracks.

The license plates of the vehicles are detected by using a cascade classifier with Haar features that is based on the Gentle AdaBoost (GAD) meta-algorithm, in the fifth section. It is properly trained on a vehicle plate dataset containing positive and negative samples and applied successfully to the video frames. A vehicle number detection and recognition approach is then applied to each detected license plate in that section. The numbers are detected applying some morphological operations to extract their characters and a template matching process is performed next on the obtained alfa-numerical characters, to recognize them.

The vehicle logos are then located by training, validating and testing a YOLO-v8 deep learning detection model and applying the CNN-based detector on the detected and recognized vehicles, in the sixth section. Some results of the described computer vision techniques are discussed in the following section and the conclusions of the research are drawn in the final section.

## 2. DEEP LEARNING-BASED MULTIPLE VEHICLE DETECTION TECHNIQUE

The considered vehicle detection task, which addresses the problem of localization of multiple vehicle in the traffic videos, has been approached by using various solutions. They include the frame differencing [4], Support Vector Machines (SVM) with HOG features [5] and other local-pattern descriptors [6], cascade classfiers [7], Active Contours [8], genetic algorithms [9], Aggregate Channel Features (ACF) [10], Gaussian Mixture Models (GMM) [10, 11] and Deep Learning frameworks [12, 13, 14].

We also propose here a deep learning-based technique for solving the multiple vehicle detection task. Thus, we have built a You-Only-Look-Once (YOLO) v5 network-based detector [15] by transfering the learning of a MobileNet v2 model to construct its feature extraction sub-network [16]. A MobileNet with 154 layers and the weights trained on ImageNet database has been assembled by us. This pretrained CNN has been modified by replacing the input layer, *input_1*, to a layer with $[300 \times 300 \times 3]$ input size. Next, the addition layer *block_11_add* has been considered for feature extraction and all the layers positioned after it have been removed from this deep network.

Its detection sub-network has been created by using 2 groups of 3 connected layers: convolution, ReLU and batch normalization. Convolution layer filter size has been set at $[3 \times 3]$ and 4 anchor boxes have been considered for prediction: $[8, 8]$, $[32, 48]$, $[40, 24]$, $[72, 48]$. We have built the YOLO v5 Class Convolution layer, YOLO v5 Transform layer with 2 anchors and YOLO v5 Output layer with 2 anchors and added them to the other detection layers. The resulted feature extraction and detection sub-networks have been connected into a final YOLO v5 detection network.

This obtained CNN-based detector has been trained, validated and tested on a voluminous traffic database created and annotated by us in this project [17]. So, we have recorded 20 traffic videos at 30 fps, achieving more than 400K frames. We have selected more than 52K frames and annotated around 400K objects representing land vehicles (cars, buses, trucks, trams and bikes) with 2D bounding boxes. The annotated vehicle dataset has been split into the training set, containing 70% of the data, validation set, with 10% of data and testing set, with the remaining 20% of data.

A data augmentation process has been applied to the training set to increase its variety and the CNN accuracy and to overcome the overfitting effect, by applying random transforms, like rotations. Also the augmented images have been resized to the input size. Next, the pre-processed training images have been fed into the YOLO network. The CNN has been trained successfully by using the following training options: a stochastic gradient descent (SGD) algorithm with momentum, 40 training epochs, the mini-batch size value of 16 and the initial learning rate of 1e-3. We have obtained a training accuracy of 96%, a validation accuracy of 94.5% and a testing accuracy of 92%.

This trained deep learning-based detector locates properly both static and moving vehicles. It is not motion-sensitive, working well for both static- and moving-camera video sequences, and also detects successfully the partially occluded objects. However, this CNN detector does not identify exactly the vehicles, but the sub-images determined by their bounding boxes. Since some of these boxes do not fit perfectly the vehicles, being larger than them, we have introduced a vehicle contour detection solution that determines more exactly the vehicles and improves their bounding boxes.

A level-set based active contour segmentation approach is proposed for this task. We introduced some PDE-based active contours for image segmentation in our past works [18, 19]. Here we consider an active contour model inspired by geodesic active contour (GAC)-based schemes with level-set functions, that is created using the next nonlinear diffusion-based model:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \beta\nabla\cdot(\psi(|\nabla v|)\delta(\|\nabla u\|)\nabla u)\delta(\|\Delta u\|) - \lambda\|\nabla u\|\psi(|\nabla v_\sigma|) = 0, \\ u(0,x,y) = u_0(x,y),\ \forall(x,y)\in\Omega, \\ \dfrac{\partial}{\partial \vec{n}}(t,x,y) = 0,\ \forall(x,y)\in\partial\Omega, \end{cases} \tag{1}$$

where $u$ represents a level-set function, $v$ is the vehicle image function, $\lambda,\beta \in [0.5,2)$, $\Omega \subseteq R^2$, $v_\sigma = v^* G_\sigma$, the Gaussian 2D kernel $G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\,e^{-\frac{x^2+y^2}{2\sigma^2}}$, the stopping function is

$$\psi : [0,\infty) \to [0,\infty) : \psi(s) = e^{-\frac{s^3}{\tau}} \tag{2}$$

with $\tau \geq 3$, and the considered diffusivity function has the next form:

$$\delta : [0,\infty) \to [0,\infty) : \delta(s) = \eta\left(\frac{\tau}{|\xi s^2+\alpha|}\right)^{\frac{1}{3}} \tag{3}$$

where $\eta \in (0,1]$, $\alpha,\xi \in (1,3]$.

The proposed nonlinear PDE-based segmentation model is solved numerically by applying the finite difference method [20-23]. The obtained active contour evolves $u$ to the real boundaries of the vehicle, in approximately 100 iterations. A new and more reliable bounding box, based on these boundaries, is then created for the detected vehicle.

## 3. A MOTION-BASED VEHICLE TRACKING APPROACH

Vehicle tracking, or counting, which represents the process of determining the vehicles trajectories in the video traffic sequence, has been addressed by developing techniques based on the mean-shift approach [24], Kalman filters [25], optical flow estimation [26], Hidden Markov Models (HMM) [27], tracking by detection [28], Particle Swarm Optimization (PSO) [29] and convolutional neural networks (CNN) [30]. Here one applies a motion-based tracking approach to the previously detected vehicles.

Thus, the detections are associated to the same video vehicle object using the motion estimation provided by a Kalman filter [25, 31]. A Kalman filtering algorithm estimates the state of a system using observations or measurements, and represents an effective instrument for many CV tasks [31]. We have modeled the video sequence $V = [F_1, F_2, ..., F_N]$ and used a Kalman filter to predict a vehicle instance in each frame $F_i$ and identify the likelihood of each detection to be assigned to each *vehicle track* labeled by an integer ID value. The proposed Kalman filtering procedure is composed of the next prediction equation

$$X(k) = AX(k-1) + W(k-1) \tag{4}$$

and the update equation

$$Y(k) = HX(k-1) + V_{\text{obs}}(k) \tag{5}$$

where $X(k)$ is the systems state at the moment $k$, $Y(k)$ represents the measurement at the time $k$, the state transition matrix $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and measurement matrix $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ are related to the constant velocity motion model, the process noise $W(k-1) \sim N(0, \mathcal{Q})$, with covariance $\mathcal{Q}$ modeled as a diagonal matrix based on motion noise vector $[100, 25]$, and the observation (measurement) noise $W(k-1) \sim N(0, R)$, where its covariance $R$ is represented as a diagonal matrix based on a measurement noise of 100.

We have considered the initial state estimation for the filter to be the 2D vector $[200, 50]$. Its initial location parameter is a 2D vector representing the centroid of the first detection of $F_1$. Our Kalman filter predicts the centroid of each track in the video frame $F_i$. The assignment of the vehicles detected in $F_i$ to the proper tracks is based on a cost minimization operation. That cost constitutes the negative log-likelihood of a vehicle detection related to a track and the Euclidean distance between the predicted centroid of that track and the detections centroid. Each vehicle detection that is assigned to a track receives the ID value of the respective track.

Each unassociated detected vehicle in the current video frame is treated as the beginning of a new video vehicle. Therefore, it receives a new track label that represents the lowest unassigned integer ID. If a vehicle track remains unassigned for a large enough number of consectutive frames, it ends because it means its vehicle has left the traffic video scene. The vehicle trajectories determined by this motion-based tracking scheme are next used to improve the vehicle classification results.

## 4. CNN-BASED VEHICLE RECOGNITION TECHNIQUE

A vehicle recognition process is performed on the detected and counted vehicles. We have considered the following land vehicle classes: *Car, Bus, Truck, Tram* and *Bike*. A transfer learning-based network that modifies a CNN-based classification model is introduced here to label all vehicle detections with these class names.

The proposed deep learning classifier is based on GoogLeNet, a CNN using the Inception network modules and trained on 1.2 million images of ImageNet [32]. Since the pre-trained GoogLeNet model can classify the objects into 1000 categories, we have altered its 144-layer architecture, in order to work for our 5 vehicle classes only.

First, we have created a new input layer characterized by the $[300 \times 300 \times 3]$ input size. Then, the $142^{\text{th}}$ layer of the convolutional network, which is the 1000 fully connected (FC) layer *loss3-classifier* and works a high-level feature learner, is replaced to the new FC layer named *Vehicle Feature Learne*r. The newly introduced fully connected layer is constructed for the 5 classes, has the learning rate factors for weights and biases equal to 10, and is able to learn the high-level characteristics of the vehicle objects. The final layer of the CNN, called *Classification Output* and designed for 1000 classes, has been also replaced by us. Thus, the classification layer *Vehicle Recognizer* is introduced instead of it in the deep neural network.

The resulted CNN-based classifier has been trained, validated and tested on a vehicle dataset prepared by us and containing 25K vehicle images. It has been divided in 5 sets corresponding to vehicle classes, each of them having 5K images. One has split this dataset into the training dataset, with 80% of the data, validation dataset, with 10% of the images, and the testing dataset, containing the remaining 10% of data. These 3 sets have been pre-processed, by augmenting and resizing at the input size value.

We have applied the optimal training options and started a network training process. Thus, this CNN has been trained for 32 epochs, by shuffling each epoch, with a SGD-based algorithm, the initial learning rate of $3e^{-4}$ and the mini-batch size parameter value 4. We have achieved a 99% training accuracy, a 98.5% validation accuracy and a 92.5% testing accuracy, these accuracy values illustrating the effectiveness of this DL model.

The vehicle class is recognized successfully for the most detected vehicles, but some classification errors may still be generated. We have fixed these errors using the vehicle tracking results. Thus, our classification improving approach assigns to all the instances of a vehicle in a traffic sequence the most common label inside the track of that vehicle.

A vehicle detection, counting and classification example is described in Fig. 1. It displays the results generated by this framework in the $1^{st}$, the $10^{th}$, the $20^{th}$ and the $30^{th}$ frame of a traffic video. The detected vehicles are marked by yellow bounding boxes and are labeled with their track number and class name, which is *Bus* or *Car*. One has counted 5 vehicles in this traffic sequence.

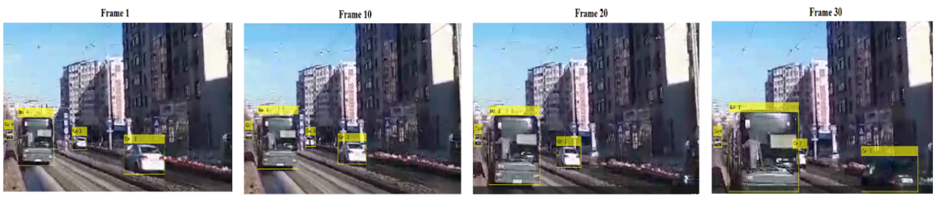


Fig. 1. A vehicle detection, counting and classification example

## 5. VEHICLE LICENSE PLATE DETECTION AND NUMBER RECOGNITION SOLUTIONS

The vehicle license plate detection represents a computer vision task that plays an important role in the intelligent transportation systems. It has been approached by using detection techniques based on the Cluster Run Length Smoothing Algorithm (CRLSA) [33], SVM with morphological operations [34], AdaBoost algorithm [35] and the deep learning networks [36, 37].

Here we propose a cascade classifier-based license plate detection approach. So, a Haar feature-based boosting classifier, which represents an ensemble learning-based scheme that combines several weaker classifiers, has been considered for this purpose.

Thus, we have created a training plate image dataset composed of 4950 images, several hundreds of them recorded by us and the others collected from some free public plate databases [38]. The obtained training dataset contains 2500 *positive samples* representing various license plates of the land vehicles and 2450 *negative samples* that represent some non-license plate images.

One has trained a Haar Cascade Classifier using the Gentle AdaBoost (GAD) meta-algorithm [39], in OpenCV, with these positive and negative sample datasets. The following options have been applied in the training process: a number of 25 cascade stages, the $[24 \times 24]$ sample size, the maximum number of threads = 5, a maximal false alarm rate of 50% and a minimum hit rate of 99.5%.

The trained boosting classifier has been applied to each frame of the video sequence $V$, the license plate subimages being detected, as bounding boxes. A high plate detection rate has been achieved by this cascade classification model, which is illustrated by its high Precision and Recall scores. If a license plate bounding box detected by this algorithm is located *within* a detected vehicle bounding box, then that plate is associated to the respective vehicle. The plates that could not be associated to any vehicles have been discarded, since they may represent false positives.

Next, a vehicle number extraction process has been applied to each detected license plate. The proposed vehicle number detection and recognition technique performs the following tasks:

- Pre-processing
- License plate image segmentation
- Character recognition

In the pre-processing stage, the analyzed plate image has been resized by augmentation and converted from RGB to the gray-level form. Then, a 2D Median filtering process has been applied to the obtained grayscale image.

Then, an automatic threshold-based image segmentation has been applied to extract the license plate foreground. The used threshold has been determined by minimizing the intra-class intensity variance using the Otsu's method [40]. Thus, the binary image coresponding to the analyzed plate has been achieved and some mathematical morphological operations have been performed to improve it [41]. So, one has performed a dilation procedure, followed by an erosion one, both of them using a square structuring element with a $[2 \times 2]$ dimension.

Next, the connected components with inappropriate sizes and shapes have been removed from the binary image. They include the components with very low areas and those with big differences between width and height that may represent long horizontal lines. The remaining connected components have been extracted as the detected characters of the vehicle number.

The vehicle number recognition task is solved by recognizing each of the detected alpha-numeric characters. A template matching-based optical character recognition (OCR) technique has been considered for this task [42]. An alpha-numeric template dataset has been created for this task. It contains template images of the 10 digits $(0, ..., 9)$ and the 26 letters of the English alphabet $(A, B, , Z)$.

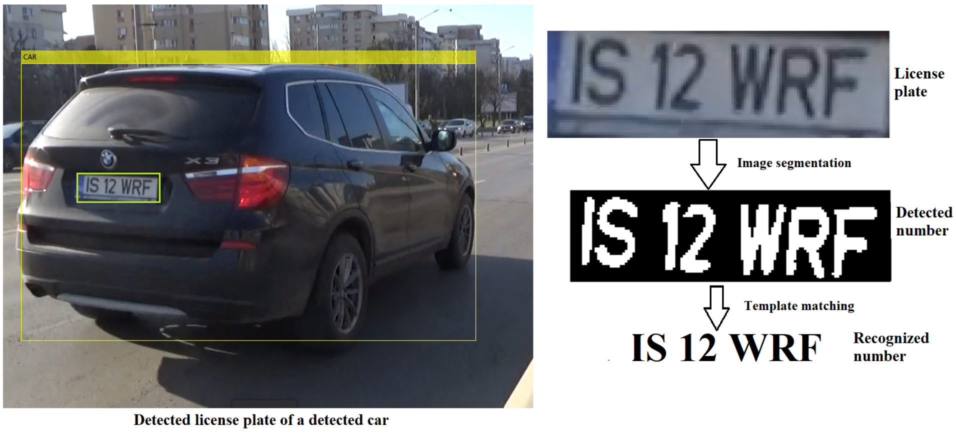


Fig. 2. Example of vehicle, license plate and number detection and recognition

A cross-correlation-based template matching algorithm using these stored templates has been applied to the detected characters. Thus, for each of these characters, one determines the normalized cross-correlation coefficient values between its image and the alpha-numeric templates [43]. The template corresponding to the highest cross-correlation score is selected as the *match* for that character. The alpha-numerical sequence of these determined matches represents the recognized vehicle number.

A vehicle, license plate and number detection and recognition example based on the proposed AI and CV techniques is described in Fig. 2. First, an auto-vehicle is detected and recognized as *car*, then its license plate is also detected. The connected components of this plate are then extracted and recognized as alpha-numericals by the template matching scheme, the vehicle number thus being obtained.

## 6. A CNN-BASED VEHICLE LOGO DETECTION APPROACH

Since the vehicle logo represents a signifiant piece of information about vehicles, its detection constitutes an important computer vision task in the vehicle information extraction domain. Some of the existing vehicle logo detections techniques use cascade classifiers [44], boosting algorithms combined to SVM [45] and deep learning models [46].

We have also proposed a deep learning-based vehicle logo detection technique. Thus, a CNN-based logo detector that uses a YOLO-v8 deep network has been considered for this purpose.

We have prepared a voluminous dataset with 55K vehicle images, recorded by us or collected from the Roboflow Universe [38], which contains over 300K logos annotated with 2D bounding boxes. Then, this logo dataset has been divided into: the training set containing 42354 logos representing 77% of data, validation set with 6166 images representing 11.2% of data, and the testing set of 6480 images representing 11.8% of the data.

The training and validation datasets are then pre-processed by applying some filtering procedures, resizing to the input size and data augmentation processes. Next, this YOLO v8 CNN has been trained successfully on the logo training set by applying the following training parameters: the SGD algorithm with momentum 0.937, 36 training epochs, the mini-batch size value = 16 and the initial learning rate = 0.01. Then it has been validated by using the validation dataset. We have achieved a CNN training accuracy of 85%, a validation accuracy of 82% and a testing accuracy of 80%.

The proposed deep learning detector locates successfully the bounding boxes of the vehicle logos when applied to the video frames of $V$. However, although it has a good logo detection rate, it may generate some false positives and negatives. Therefore, the logos whose bounding boxes are detected *within* a vehicle bounding box and close enough to the license plate bounding box have to be associated to those vehicles, while the others must be discarded, since they represent false positives. An example of a logo detected by the proposed DL-based approach within the bounding box of a detected vehicle that is recognized as a car is displayed in Fig. 3.

The detected vehicle logos can be further recognized by applying a template matching technique based on a logo template dataset, which is quite similar to that applied for vehicle number recognition. Such a logo recognition technique that would identify successfully the vehicle brand represents a focus of our future research.

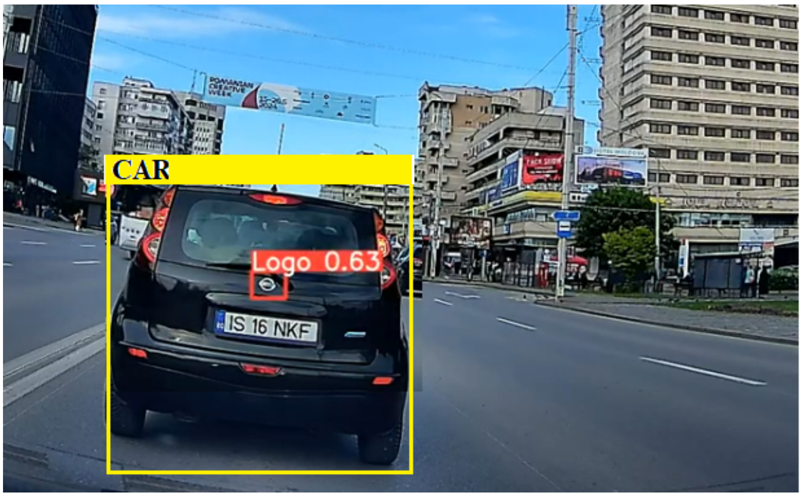


Fig. 3. Example of a detected logo of a detected and recognized vehicle

## 7. DISCUSSIONS AND RESULTS

The AI and CV techniques solving the vehicle information detection tasks, which have been proposed here, have been tested successfully on numerous vehicle-related datasets. The described detection approach locates effectively both static and moving vehicles and works well for both fixed and moving camera videos, since it is motion-insensitive.

Thus, it represents a better solution than motion-sensitive vehicle detection schemes based on frame differences [4], optical flow and GMM [11]. Unlike other approaches, it works efficiently for multiple vehicles of all views, addresses properly the inter-vehicle occlusions, providing good results for crowded traffic scenes. It also generates a low number of false positives and negatives, which means high values of the *Precision, Recall* and $F_1$ performance metrics.

The proposed motion-based vehicle tracking technique is also very effective, outperforming other methods, such as the Intersection over Union (IoU)-based trackers [47], and it is quite fast, having a low computational cost too. Anyway, its execution time depends on the videos length, $N$, its frames size and the number of vehicles in the video frames.

Our combined detection and counting solution achieves also high scores of the performance values and outperforms many other vehicle detection and tracking schemes. For the vehicle detection and tracking algorithms, *Precision* represents the percentage of the current track that is correct while *Recall* computes the percent of the ground-truth trajectory overlapping with the respective track. Some method comparison results are provided in Table 1. As shown by these results, our framework outperforms many other machine learning-based vehicle detection and tracking methods and even some deep learning techniques [12], when compared on the same testing dataset from our vehicle database [17].

Table 1. Vehicle detection and counting method comparison

| **Detection and tracking technique** | **Precision** | **Recall** |
|---|---|---|
| This YOLO - v5 + Kalman filter approach | 0.8957 | 0.8864 |
| Frame Differencing + matching | 0.7504 | 0.7411 |
| GMM + Kalman filtering | 0.6692 | 0.6733 |
| SVM-HOG + feature matching | 0.7618 | 0.7618 |
| Cascade Classifier + Kalman filter | 0.7862 | 0.7862 |
| ACF + IoU | 0.8109 | 0.8231 |
| Faster R-CNN+ IoU | 0.8452 | 0.8336 |
| YOLO v8 + DeepSORT | 0.9017 | 0.8985 |

The transfer learning-based vehicle recognition technique described here has also obtained some high score values for these performance measures: $Precision = 0.874$, $Recall = 0.869$ and $F_1 = 0.871$ respectively. It clearly outperforms the GoogleNet classifier from which it has been derived.

The vehicle license plate detection component of this framework has achieved the following performance metric scores: $Precision = 0.812$ and $Recall = 0.809$. While it outperforms some machine learning-based license plate detection methods, such as those based on SVMs, it is outperformed by some DL-based techniques [37]. A quite high vehicle number detection and recognition rate, of over 80%, has been also obtained by our method.

The proposed CNN-based vehicle logo detection technique has also achieved quite good results based on the experiments performed by us. However, it has obtained some lower performance metric values: $Precision = 0.762$ and $Recall = 0.778$. We intend to improve its performance by performing the YOLO network training on a larger logo image dataset and using new training options, as part of our future research.

## 8. CONCLUSIONS

A vehicle information detection framework has been described in this work. Its AI and CV-based components solve effectively several vehicle information detection tasks, including multiple vehicle detection, tracking and classification, vehicle license plate detection, vehicle number detection and recognition, and vehicle logo detection. The proposed system combines successfully some novel machine learning, deep learning and nonlinear PDE-based models for fulfilling all these tasks.

So, several deep learning-based techniques have been proposed for the vehicle detection and recognition, and vehicle logo detection tasks. Some effective convolutional neural networks have been built, trained, validated and tested for those purposes. Also, we created and annotated some voluminous vehicle and logo datasets for the considered CNN-based detectors and classifiers.

Also, successful machine learning-based techniques have been introduced for solving the multiple vehicle tracking, license plate detection and vehicle number extraction and recognition tasks. The nonlinear PDE-based models have been used to create new active contours that improved the vehicle detection results.

Successful results have been obtained by all the components of the proposed framework, some high detection and recognition rates being achieved. These vehicle information detection results can be applied successfully in the intelligent transportation systems domain. In fact, the described research is already part of a contract-based intelligent traffic monitoring project that is acknowledged below. As part of our future research in this AI project, we intend to further improve the proposed ML and DL-based vehicle information detection techniques, as mentioned in the previous sections, and also to develop novel detection and recognition algorithms for other traffic-related entities.

**Acknowledgments.** This research work was supported by a grant of the Romanian Academy, GAR-2023, Project Code 19.

## References

[1] A. M. De Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, L. A. Villas, *Traffic management systems: A classification, review, challenges, and future perspectives*, International Journal of Distributed Sensor Networks, **13** (4) (2017). 1550147716683612.

[2] K. Abdulrahim, R. A. Salam, *Traffic surveillance: A review of vision based vehicle detection, recognition and tracking*, Int. J. Appl. Eng. Res, **11** (1) (2016), 713-726.

[3] G. Dimitrakopoulos, P. Demestichas, *Intelligent transportation systems* IEEE Vehicular Technology Magazine, **5** (1) (2010), 77-84.

[4] W. Li, J. Yao, T. Dong, H. Li, X. He, *Moving vehicle detection based on an improved interframe difference and a Gaussian model*, 2015 $8^{\text{th}}$ International Congress on Image and Signal Processing (CISP), Shenyang, China, 2015, 969-973.

[5] N. Tomikj, A. Kulakov, *Vehicle detection with HOG and linear SVM*, Journal of Emerging Computer Technologies, **1** (1) (2021), 6-9.

[6] A. Foucart, O. Debeir, *Vehicle detection in traffic videos using unsupervised SVM classifiers and local pattern descriptors*, ISVC 2010, 2010.

[7] X. Zhuang, W. Kang, Q. Wu, *Realtime vehicle detection with foregroundbased cascade classifier*, IET Image Processing, **10** (4) (2016), 289-296.

[8] L. Li, S. Yoon, J. Liu, J. Yi, *Multi-scale car detection and localization using contour fragments*, 2014 IEEE International Conference on Image Processing (ICIP), (2014, October), 1609-1613. IEEE.

[9] V. D. Nguyen, T. T. Nguyen, D. D. Nguyen, S. J. Lee, J. W. Jeon, *A fast evolutionary algorithm for real-time vehicle detection*, IEEE Transactions on Vehicular Technology, **62** (6) (2013), 2453-2468.

[10] A. Chetouane, S. Mabrouk, I. Jemili, M. Mosbah, *Visionbased vehicle detection for road traffic congestion classification*, Concurrency and Computation: Practice and Experience, **34** (7) (2022), e5983.

[11] R. Y. Bakti, I. S. Areni, A. A. Prayogi, *Vehicle detection and tracking using gaussian mixture model and kalman filter*, 2016 International Conference on Computational Intelligence and Cybernetics, (2016, November), 115-119. IEEE.

[12] Q. Fan, L. Brown, J. Smith, *A closer look at Faster R-CNN for vehicle detection,* 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden,2016, 124-129.

[13] Y. Chen, Z. Li, *An effective approach of vehicle detection using deep learning*, Computational Intelligence and Neuroscience, 2022 (1), 2019257.

[14] M. A. B. Zuraimi, F. H. K. Zaman, *Vehicle detection and tracking using YOLO and DeepSORT*, 11$^{\text{th}}$ IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), april 2021, 23-29.

[15] X. Wu, D. Sahoo, S. C. Hoi, *Recent advances in deep learning for object detection*, Neurocomputing, **396** (2020), 39-64.

[16] A. G. Howard, et al. *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, arXiv:1704.04861, 2017.

[17] SIMPATIA project: http://iit.academiaromana-is.ro/simpatia

[18] T. Barbu, *Robust contour tracking model using a variational level-set algorithm*, Numerical Functional Analysis and Optimization, Vol. **35**, Issue 3 (2014), 263-274.

[19] T. Barbu, S. I. Bejinariu, *CNN-based Moving Vehicle Recognition using GMM-based Foreground Modeling, Level-set based Segmentation and Kalman Filter-based Tracking*, 2024 International Conference on Innovations in Intelligent SysTems and Applications (INISTA) (2024, September), 1-6. IEEE.

[20] G. Boole, *Calculus of finite differences*, BoDBooks on Demand, 2022.

[21] T. Barbu, *Novel Diffusion-Based Models for Image Restoration and Interpolation*, Book Series: Signals and Communication Technology, Springer International Publishing, 2019.

[22] T. Barbu, *Nonlinear PDE model for image restoration using second-order hyperbolic equations*, Numerical Functional Analysis and Optimization, **36** (11) (2015), 1375-1387.

[23] T. Barbu, *Mixed Noise Removal Framework using a Nonlinear Fourth-order PDE-based Model*, Applied Mathematics & Optimization, **84**, 18651876, Springer US, August.

[24] L. Wei et al., *A SIFT-based mean shift algorithm for moving vehicle tracking*, 2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014.

[25] K. Mantripragada, F. C. Trigo, F.P. Martins, A. de Toledo Fleury, *Vehicle tracking using feature matching and Kalman filtering*, Proc. international congress of mechanical engineering, 2013, 361-370.

[26] Y. Liu, et al., *Optical flow based urban road vehicle tracking*, 2013 ninth international conference on computational intelligence and security. IEEE, 2013.

[27] J. Kato et al., *An HMM/MRF-based stochastic framework for robust vehicle tracking*, IEEE Transactions on intelligent transportation systems 5.3 (2004), 142-154.

[28] T. Barbu, *Deep Learning-based Multiple Moving Vehicle Detection and Tracking using a Nonlinear Fourth-order Reaction-Diffusion based Multi-scale Video Object Analysis*, Discrete & Continuous Dynamical Systems - Series S, AIMS Journals, **16** (1) (2023), 6-32.

[29] G. Jati, A. A. S. Gunawan, W. Jatmiko, *Dynamic swarm particle for fast motion vehicle tracking*, ETRI Journal, **42** (1) (2020), 54-66.

[30] C. Kwan, B. Chou, A. Echavarren, B. Budavari, J. Li, T. Tran, *Compressive vehicle tracking using deep learning*, 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), (2018, November), 51-56. IEEE.

[31] D. Simon, *Kalman filtering*, Embedded systems programming, **14** (6) (2001), 72-79.

[32] L. Alzubaidi et al., *Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions*, Journal of big Data, **8** (2021), 1-74.

[33] S. N. H. S.Abdullah, M. N. Sudin, A. S. Prabuwono, T. Mantoro, *License plate detection and segmentation using cluster run length smoothing algorithm*, Journal of Information Technology Research (JITR), **5** (3) (2012), 46-70.

[34] S. Miyata, K. Oka, *Automated license plate detection using a support vector machine,* 2016 14$^{\text{th}}$ International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 2016, 1-5.

[35] L. Dlagnekov, *License plate detection using adaboost*, Computer Science and Engineering Department, San Diego, 2004.

[36] O. G. Abdulateef et al., *Vehicle License Plate Detection Using Deep Learning*, 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2022.

[37] A. Selmi, M. B. Halima, A. M. Alimi, *Deep learning system for automatic license plate detection and recognition*, 2017 14th IAPR international conference on document analysis and recognition (ICDAR), Vol. **1**, 2017, November 4, 1132-1138. IEEE.

[38] Roboflow Universe – https://universe.roboflow.com/.

[39] P. Viola, M. Jones, *Rapid object detection using a boosted cascade of simple features,* Proceedings of CVPR 2001, 2001. IEEE

[40] X. Xu, S. Xu, L. Jin, E. Song, E., *Characteristic analysis of Otsu threshold and its applications*, Pattern recognition letters, **32** (7) (2011), 956-961.

[41] P. Soille, *Morphological operators*, Computer Vision and Applications, Academic Press, 2000, 483-515.

[42] A. Chaudhuri, K. Mandaviya, P. K. Badelia, S. Ghosh, *Optical character recognition systems*, Springer International Publishing, 2017, 9-41.

[43] J. C. Yoo, T. H. Han, *Fast normalized cross-correlation*, Circuits, systems and signal processing, **28** (2009), 819-843.

[44] T. Zou, X. Chen, Y. Chen, Q. Sun, *Cascade-LogoNet: Eliminating Classification Ambiguity with Cascaded Logo Detection,* 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 2023, 1-5.

[45] H. Pan, B. Zhang, B., *An integrative approach to accurate vehicle logo detection*, Journal of Electrical and Computer Engineering, **2013** (1) (2013), 391652.

[46] S. Sahel, M. Alsahafi, M. Alghamdi, T. Alsubait, *Logo detection using deep learning with pretrained CNN models*, Engineering, Technology & Applied Science Research, **11** (1) (2021), 6724-6729.

[47] H. Rezatofighi et al., *Generalized intersection over union: A metric and a loss for bounding box regression*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.