UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

DYNA
http://dyna.medellin.unal.edu.co/

EspiNet V2: a region based deep learning model for detecting motorcycles in urban scenarios

Jorge Ernesto Espinosa-Oviedo [a], Sergio A. Velastín [b] & John William Branch-Bedoya [c]

[a] Facultad de Ingeniería, Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia. jeespinosa@elpoli.edu.co
[b] Cortexica Vision Systems Ltd. UK, Universidad Carlos III de Madrid, Spain and Queen Mary University of London, UK. sergio.velastin@ieee.org
[c] Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia. jwbranch@unal.edu.co

Received: August 12th, 2019. Received in revised form: November 8th, 2019. Accepted: November 28th, 2019.

Abstract
This paper presents "EspiNet V2" a Deep Learning model, based on the region-based detector Faster R-CNN. The model is used for the detection of motorcycles in urban environments, where occlusion is likely. For training, two datasets are used: the Urban Motorbike Dataset (UMD-10K) of 10,000 annotated images, and the new SMMD (Secretaría de Movilidad Motorbike Dataset), of 5,000 images captured from the Traffic Control CCTV System in Medellín (Colombia). Results achieved on the UMD-10K dataset reach 88.8% in average precision (AP) even when 60% motorcycles were occluded, and the images were captured from a low angle and a moving camera. Meanwhile, an AP of 79.5% is reached for SSMD. EspiNet V2 outperforms popular models such as YOLO V3 and Faster R-CNN (VGG16 based) trained end-to-end for those datasets.

Keywords: vehicle detection; motorcycle detection; Faster R-CNN; region-based detectors; convolutional neural network; deep learning.

EspiNet V2: un modelo basado en regiones de aprendizaje profundo para detectar motocicletas en escenarios urbanos

Resumen
Este artículo presenta "EspiNet V2", un modelo de aprendizaje profundo, fundamentado en el detector basado regiones Faster R-CNN. El modelo es usado para la detección de motocicletas en entornos urbanos, donde se presenta algún nivel de oclusión. Para el entrenamiento de dicho modelo, se utilizaron dos conjuntos de datos: el conjunto de datos de motocicletas urbanas (UMD-10K) que cuenta con 10,000 imágenes anotadas, y el nuevo conjunto de datos de motos de la Secretaría de Movilidad (SMMD), con 5,000 imágenes capturadas obtenidas del Sistema CCTV de Control de Tráfico de la ciudad de Medellín (Colombia). Los resultados obtenidos en el conjunto de datos UMD-10K alcanzan el 88.8% en precisión promedio (AP), incluso con niveles de oclusión de un 60 %, utilizando imágenes capturadas desde un ángulo bajo y desde una cámara en movimiento. Por otro lado se alcanza un AP de 79.5 % para conjunto de datos de motos de la Secretaría de Movilidad (SMMD). EspiNet V2 supera modelos populares como YOLO V3 y Faster R-CNN (basado en VGG16), siendo estos entrenados de extremo a extremo utilizando los conjuntos de datos mencionados.

Palabras clave: detección de vehículos; detección de motocicletas; Faster R-CNN; detectores basados en regiones; redes neuronales convolucionales; aprendizaje profundo.

1. Introduction

The World Health Organization (WHO) reports in the *Global status report on road safety 2018* that more than half (54%) of the road traffic deaths corresponds to Vulnerable Road Users (pedestrians, cyclists, motorcyclists) [1]. From this rate, 28% corresponds to Motorcycles. The annual report Traffic Accidents of the Andean Community (Bolivia, Colombia, Ecuador and Perú) [2] documented 347,642 traffic accidents, 88% of them occurred in urban areas. In this region, for year 2017, Colombia has 57.35% of the total traffic accidentally rate, reporting 171,571 occurrences with

How to cite: Espinosa-Oviedo, J.E, Velastín-Carroza, S.A. and Branch-Bedoya, J.W, EspiNet V2: a region based deep learning model for detecting motorcycles in urban scenarios. DYNA, 86(211), pp. 317-326, October - December, 2019.

6,479 fatal victims. Although for 2017 deaths due to transport accidents were reduced by 7.23% compared to 2016, these numbers still high compared to world statistics [3]. Motorcyclist are the road users most affected by traffic accidents in Colombia, reporting 49.82% of deaths and 56.36% injured victims [3]. This high accidentallity rate can be partially explained due to that 58% of the 14.880.823 total vehicles registered in Colombia for 2019 2Q corresponds to Motorcycles [4], and 76.64% of these motorcycles belongs to the street sport segment which is used as a regular transport mean.

Air quality is also an issue in the main cities of Colombia. The National Planning Department (DNP) estimated that, during 2015, the effects of air pollution were associated with 10,527 deaths and 67.8 million symptoms and diseases [5]. The contaminant with the greatest potential for affectation is Particulate Material Less than 2.5 microns (PM2.5), which is made up of very small particles, produced mainly by heavy vehicles that use diesel as fuel, and which can carry very dangerous material for human body such as heavy metals, organic compounds and viruses, thus affecting the respiratory tract [6]. In Colombia, 59% of PM2.5 is produced by land transportation, from which 40% corresponds to motorcycles. It is therefore desirable to monitor urban motorcycle traffic to reduce incidents and air pollution on what are becoming very congested roads.

Video analytic techniques for vehicle detection have been used in urban traffic analysis, reporting success for detecting regular vehicles (bus, cars, trucks), but there is scarce literature on the analysis of motorcycles as major users in many urban environments, characterised by frequent occlusion between vehicles in congested traffic conditions.

In this paper, we introduce EspiNet V2 a deep learning model based on the two-stage detector Faster R-CNN [7] (Faster Regions with Convolutional Neural Networks features). The model is used to detect motorcycles in congested urban traffic scenes. The paper is structured as follows; section 2 reviews the literature on motorcycles detection, section 3 gives a brief introduction to deep CNN and Faster R-CNN, section 4 explains the proposed EspiNet V2 model, detailing its architecture and main differences w.r.t Faster R-CNN. Section 5 describes the different experiments done employing the UMD-10K and SMMD datasets, providing a results analysis. The article finishes with section 6 with conclusions and proposed future work.

2. Motorcycle detection

Video analytics supports most of the current urban traffic analysis and vehicle detection systems. Traditional approaches for vehicle detection extract discriminative features for vehicle representation, which later implement classification, usually using classifiers trained on those features. Features are generally extracted from object appearance or derived from motion information [8].

Motorcycle detection works based on appearance features such as edge maps are introduced in [9] using Gabor filters and the Sobel operator [10] to reduce illumination variances. Other approaches use corner detection with Harris corners [11], or even using Haar-like features [12,13], despite the poor correlation under different view angles. Feature descriptors such as Histogram of oriented gradients (HOG), Scale-invariant feature transform (SIFT), and Local binary patterns (LBP) are compared in [14] and [15] for motorcycle detection. For helmet detection in motorcycles riders Speeded up robust features (SURF), Haar-like features (HAAR) and HOG [16] have been used as feature descriptors. Meanwhile, in [17], they use hybrid descriptor based on colour for helmet identification. Appearance features based on computer-generated 3D models are used to discriminate between motorcycles and bicycles in [18], and between car/taxi, bus/lorry, motorbike/bicycle, van, and pedestrian in [19]. Background subtraction uses spatio-temporal information for detecting a moving object in a giving scene. Motorcycle detection [20,21] starts with this technique and uses segmentation to detect and separate motorcycles in the analysis. In some works, a similar approach is used, even to detect motorcycle riders without a helmet [24-29].

The most used algorithm for background subtraction is Gaussian Mixture Models (GMM) [22], used in [23,24]. For dealing with object shadows and for continuous update of parameters, Self Adaptive GMM [25] is used in [26] or adaptive background modelling used in [14] and [15]. Nevertheless, background subtraction may fail in congested scenarios or where the objects overlap each other, difficulting their detection, with camera movements, or when objects tend to become part of the background, after a prolonged static sequence as typical in traffic jams.

Motorcycle detection in [24] uses spatial features in conjunction with motion features obtained from optical flow, this type of features is useful for obstacle detection in a Lane Change Assistant (LCA) system [10].

The most frequently classifiers used for motorcycles classification are Support Vector Machines (SVM), used for classifying and counting motorcycles in [9], where object occlusion is avoided capturing images from a top-view point. For helmet detection, different types of kernels are compared in [14] and [15] using background subtraction for object detection. Head regions described by histograms are also used for helmet detection in [27], which are later classified by a linear SVM. This method may fail with drastic changes of illumination. SVMs are also used for classifying a multi-shape descriptor vehicle [25,26] demanding high computational resources for the descriptor construction and evaluation. There is also a proposed Real-Time on Road Vehicle Detection system [10], which uses a binary SVM classification by hierarchies, boosting its performance thanks to an Integrated Memory Array Processor (IMAP) architecture. Nonetheless, the model can fail in adverse weather conditions with low illumination. SVMs for motorcycles detection are also used in conjunction with Bag of Visual Words (BoVW) [28] with a Radial basis function kernel (RBF) or using HOG as a feature descriptor [29], even with 3D models as appearance features [18,19].

Other classifiers used are decision trees for Overhead Real-Time Motorbike Counting [30], where the method relies on the camera specification for decision tree rule construction. Neural networks (NN) such as the Multilayer Perceptron (MLP) have been proposed for motorcycle detection and classification, even though their architectures require tuning of many parameters and the implemented loss function may not converge to a local optimum. Nevertheless, NN are used for helmet detection in [16,31]. There is also Fuzzy neural network (FNN) [24], but without a significant number of motorcycles to detect in their dataset. Finally, K-Nearest Neighbor (KNN) is also used for Helmet detection [23]; nevertheless, this model relies on the background subtraction accuracy for motorcycle individualisation, which may fail in occluded scenarios.

2.1. Deep learning for motorcycle detection

In recent years deep learning has erupted in the field of computer vision showing impressive results, mainly due to the computing capacity that GPUs (Graphics Processing Units) provide for training models, as well as the creation of vast manually labelled datasets of generic objects.

The work of Vishnu et al. [32] use Convolutional Neural Networks (CNNs) as feature extractors in combination with background subtraction for object detection. Once the object is detected, for instance using GMM, the features extracted using the CNN model (e.g., AlexNet), are used to perform classification [33]. Instead of background subtraction, object localisation uses selective search as in [34]. Nevertheless, the work in [35] proposes a straightforward CNN for detecting and classifying motorcycles. The input image is passed through the feature extraction layers generating a motorcycle score map. This score map is thresholded followed by non-maximal suppression for individual motorcycle detections. Most recent works are oriented to detect helmet violation for motorcycle users. For instance, in [36], motorcycles are detected using HOG+SVM, and later, the riders head area is supplied to a CNN model for helmet presence detection. The work in [32] proposes a similar approach. Meanwhile, in [37], moving objects are detected using motion detection algorithms, a pedestrian CNN model is used to detect humans, later a CNN is used again to detect the presence of helmet and the colour of it.

Unfortunately, the analysed literature lacks a unified metric for reporting results and most of the methods use proprietary datasets which are seldom available for comparison and use by the research community.

3. Deep CNN networks and Faster R-CNN

Convolutional Neural Networks (CNNs) are a type of neural network, whose architecture is based on convolutional filters able to capture spatial patterns and that reduce the computational burden of learning parameters. This approach produces features invariant to scale, shift or rotation as the receptive fields provide the neurons access to primitive features such as oriented edges and corners in the initial convolutional layers, which are then aggregated generating more complex features going deeper in the model. Features derived from CNNs often outperform feature descriptors such as HOG, SIFT, SURF, LBP [38,39].

While features obtained from CNNs are very useful for classification, the problem of object detection not only involves the classification of the objects but their localisation in the image. When Spatio-temporal information is available (video sequences), approaches such as background subtraction, optical flow or motion detection algorithms, help to identify moving objects, extracting features from the detected blobs, which are later classified. This approaches may fail due to camera movement, static objects, or even illumination changes. The lack of Spatio-temporal information as in single or static images (frames) forces the use of approaches that combine sliding window search (which slides a window e.g. from left to right, and from up to down in the image extracting patches later used for classification) with binary classifiers (object vs background). Object proposal algorithms, like Branch & Bound [40], Selective search [41] Spatial Pyramid Pooling [42] and Edge boxes [43] are approaches designed to deal with the large numbers of windows useful to cover different aspect ratios and scales.

Two-stage detectors as R-CNN (Regions with CNN features) [44] use selective search to generate up to 2,000 regions which are provided to a CNN to produce a feature vector later fed into SVM to determine the occurrence of an object and the values necessary to adjust the bounding box to the detected object. Since the number of selective search proposals is fixed and is a time-consuming task, Fast R-CNN [45] feeds the input image to the CNN to generate a feature map, identifying the proposal regions which are later warped and fed into a fully connected layer using a Region of Interest (RoI) pooling layer. This model reduces computational time due to the use of only one convolution operation per image instead of 2000 of the R-CNN model. Nevertheless, the Region Proposal is still the bottleneck during testing time.

Faster R-CNN [7] speeds up the detection process, eliminating the use of selective search and using a CNN model which simultaneously learns region proposal and perform object detection. As in Fast R-CNN, the input image is passed through the CNN model generating a feature map, over this feature map the Region Proposal Network (RPN) deploys a sliding window to generate *n* bounding boxes with their associated scores per window. These *n* boxes are called

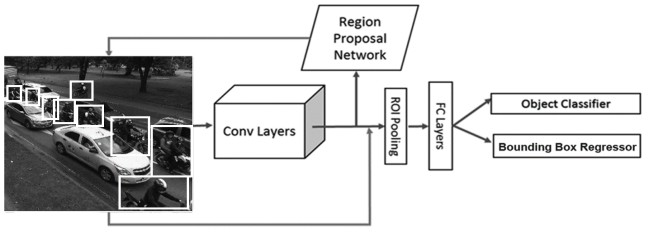

Figure 1. The components of Faster R-CNN.
Source: Image modified from [46]

anchor boxes and represent common sizes and aspect ratios that objects can have. A RoI pooling layer is used to reshape predicted region proposals, classifying the image inside the proposed region and generating the offset values for bounding boxes using regression (Fig. 1).

4. EspiNet V2

EspiNet V2 (Fig. 2) is a deep learning model proposed here that is based on the region based detector Faster R-CNN. This model is used to detect motorcycles in congested urban traffic scenes. Occluded scenarios are frequent on urban traffic analysis (Fig. 3). General vehicle detection in urban conditions has been studied by many authors. Occluded situations has been analysed using the KITTI dataset [47], which unluckily lacks a motorcycle category. EspiNet V2 is an improved version of the one presented in [48]. This new model increases the number of convolutional layers, pursuing to capture more aggregate features that contribute to identify motorcycles in the given images.

EspiNet V2 is publicly available for download (https://github.com/muratayoshio/EspiNet). The model can detect motorcycles in congested urban scenarios and, as in Faster R-CNN, unifies two networks: a Region proposal network (RPN) and a Fast R-CNN [45] detector, sharing the convolutional layers between the two architectures. The main difference between EspiNet V2 and Faster R-CNN lies in the CNN implemented. The best results of Faster R-CNN are obtained working with quite deep models such as VGG-16 [49] having 16 weight layers, 13 of them convolutional and ~ 138 million parameters to be learned. EspiNet V2 uses a more concise CNN network with only six layers (4 convolutional)

reducing the number of parameters to learn (~2 million), still outperforming Faster R-CNN in the chosen task (see section V).

Table 1 shows in detail the configuration and parameters of the EspiNet V2 model.

The input size for classification is the size of the training images. Meanwhile, for detection task, the input layer is a tensor of 32x32x3 (32x32 pixels, 3 channels), considering that in UMD-10K and SMMD datasets the smallest annotated object has a size of 25 pixels. This input layer is zero-center normalised, and its size is determined according to the processing time and the spatial detail the CNN model has to resolve. The first convolutional layer has 64 filters of size 3x3. The same filter size is used for all the convolutional layers to produce a small receptive field, to capture smaller and complex features in the image and optimise the weight sharing process. Each convolutional layer is followed by a ReLU (rectified linear unit) layer, making the learning process computationally efficient, speeding up convergence and reducing the vanishing gradient effect.

Table 1.
Architecture and learnable parameters of EspiNet V2.

Name	Type	Activations	Learn-enables	Total of Learneables
Imageinput 32x32x3 images with zero center normalization	Image Input	32x32x3	-	0
Conv_1 64 3x3x3 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	32x32x64	Weights 3x3x3x64 Bias 1x1x64	1792
Relu_1 ReLU	ReLU	32x32x64	-	0
Conv_2 32 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	32x32x32	Weights 3x3x3x32 Bias 1x1x32	18464
Relu_2 ReLU	ReLU	32x32x32	-	0
Conv_3 64 3x3x32 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	32x32x64	Weights 3x3 x32x64 Bias 1x1x64	18496
Relu_3 ReLU	ReLU	32x32x64	-	0
Conv_4 128 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	32x32x128	Weights 3x3 x64x128 Bias 1x1x128	73856
Relu_4 ReLU	ReLU	32x32x128	-	0
maxpool 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	15x15x128	-	0
Fc_1 64 fully connected layer	Fully Connected	1x1x64	Weights 64x28800 Bias 64x1	1843264
Relu_5 ReLU	ReLU	1x1x64	-	0
Fc_2 2 fully connected layer	Fully Connected	1x1x2	Weights 2x64 Bias 2x1	130
softmax softmax	Soft-max	1x1x2	-	0
classoutput crossentropyex	Classification Output	.	-	0

Source: The Authors.

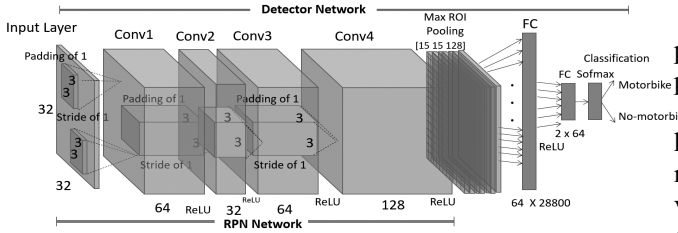

Figure 2. EspiNet V2, the Proposed CNN Model. The same model implements RPN and classification.
Source: The authors.

Figure 3. Example image of the Urban Motorbike Dataset. The smallest object size is 25 px. Occlusions are frequent between motorcycles and other vehicles.
Source: The Authors.

The last two convolutional layers duplicate the number of filters, capturing more complex features, later used for motorcycle recognition due to its enriched image representation [50]. As in Faster R-CNN Faster R-CNN [7] architecture, a max RoI pooling layer is used after the four convolutional filters for detection purposes, it removes redundant spatial information, reduces and fixes the feature map spatial size.

This layer is set to a 15x15 pixels grid covering the smallest detected object. It is the only max-pooling layer in the model since prematurely down-sampling data can lead to loss of important information necessary for learning [51]. After the first fully connected (FC) layer (64 neurons) combines all features extracted in the previous layers, which is corrected next by a ReLU layer, finally combined in the second fully connected layer. The last layer of the model is a softmax layer, which normalises the output of the previous FC layer, providing a confidence measure and computing the loss of the model. Fig. 2 shows the schematic model of EspiNet V2 network.

The multi-task loss function defined for one image is:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\,(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\,(t_i, t_i^*) \quad (1)$$

In eq. (1) i is the anchor index in a mini-batch (with positives and negatives examples anchors), p_i is the predicted probability that the anchor i is an object. The ground truth (gt) p_i^* has label 1 if the anchor is positive, 0 is the anchor is negative. t_i represents the predicted bounding box using a vector of 4 parametrised coordinates, where t_i^* is the gt box coordinates vector related to a positive anchor.

The classification loss L_{cls} part uses a logistic regression cost function. Meanwhile for the bounding box regression loss part $L_{reg}(t_i, t_i^*)$, the robust loss function (smooth L_1) is used.

$$L_{reg}(t_i, t_i^*) = \sum_{i\in\{x,y,w,h\}} smoothL_1(t_i - t_i^*), \quad (2)$$

in which

$$smoothL_1(x) = \begin{cases} 0.5\,x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise, \end{cases} \quad (3)$$

In this bounding box regression, each coordinate is parameterized as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, & t_y &= (y - y_a)/h_a, \\ t_w &= log(w/w_a), & t_h &= log(h/h_a) \\ t_x^* &= (x^* - x_a)/w_a, & t_y^* &= (y^* - y_a)/h_a, \\ t_w^* &= log(w^*/w_a), & t_h^* &= log(h^*/h_a) \end{aligned} \quad (4)$$

where *x, y,* corresponds to the boxs center coordinates, *w,* and *h* its width and height. Variables *x* corresponds to predicted box, x_a anchor box and x^* ground-truth box, (similarly for *y, w* and *h* variables). This can be assumed as a bounding-box regression from an anchor box to the closest ground truth box. The coordinates of the bounding box are values [0,1] which are relative to a specific anchor. For example, t_y denotes the coefficient for *y* (box center *x*,*y*). If t_y is multiplied by h_a and then add y_a we get the predicted *y*. The rest of parameters can be calculated in the same way.

Training comprises four steps using an alternating optimisation. The first two steps train the RPN and the detector network separately. For these first two steps, EspiNet V2 uses a learning rate of *1e-5* trying to obtain a fast convergence, as it is trained from scratch, and no pre-trained models are used for the shared convolutional layers [7]. Once the shared convolutional layers are trained and fixed, the last two steps fine-tuning the unique layers of the RPN and Fast R-CNN detector, using a learning rate of *1e-6* for a smoother process.

The optimisation training algorithm used in all the described steps is Stochastic Gradient Descent with Momentum (SGDM) (eq. (5)).

$$\theta_{\ell+1} = \theta_\ell - \alpha\nabla E(\theta_\ell) + \gamma(\theta_\ell - \theta_{\ell-1}) \quad (5)$$

where ℓ is the iteration number, the learning rate is defined as $\alpha > 0$, weights and biases define the parameter vector θ and $E(\theta)$ is the loss function. The algorithm is stochastic since it uses a subset of the training set (minibatch) to evaluate and update the parameter vector. One iteration corresponds to each evaluation of the gradient using the mini-batch. At each iteration, the algorithm takes one step towards minimising the loss function. One epoch encompasses the full pass of the training algorithm over the entire training set using mini-batches. For EspiNet v2, the number of epochs is defined after training analysis [48]. The momentum term γ regulates the contribution of the previous gradient step to the current iteration and is used to avoid oscillation along steepest descent to the optimum.

5. Experiments and results

5.1. Motorbikes datasets

To train and evaluate the proposed model, two datasets are used: The UMD-10K dataset, which is an extension of [48], with 10,000 annotated images including 317 motorcycles with 56,975 individual annotations (bounding boxes). 60% of the annotated data corresponds to occluded motorcycles (See Fig. 3). Moreover, the Secretaría de Movilidad de Medellín created the Sistema Inteligente de Movilidad de Medellín (Intelligent Mobility System of Medellín) [52], which includes a CCTV with 80 cameras to monitoring urban traffic conditions in this Colombian city. From this network of cameras, we selected six strategic surveillance located cameras (Fig. 4) to create the SSMD dataset with 5,000 images, containing 21,625 annotated motorcycles (817 different motorcycles). (Fig. 5). These dataset are available from http://videodatasets.org/UrbanMotorbike.

5.2. Results on the UMD-10K dataset

The performance of previous experiments in [48] achieved a 75.23% of Average Precision (AP) [53], training and evaluated on the UMD-7.5k, with 7,500 examples.

EspiNet is now compared with two models: YOLO V.3 [54] as a single-stage detector and for a two-stage detectors, the original Faster R-CNN [49] (VGG16 based). We selected these models since they have been extensively used to compare new proposals, and because of their good performance and their availability in the public domain. All these models were trained end to end from scratch, using the challenging UMD-10k dataset.

As is recommended by [55] an due to the large number of examples needed to train, the three models use 90% (9,000 images) of the UMD-10k dataset for training data, while the remaining 10% (1,000 images) are used for validation. The selection of training and test set is done randomly to avoid any bias in the distribution.

The proposed EspiNet V2 model obtain of 88.8% of AP

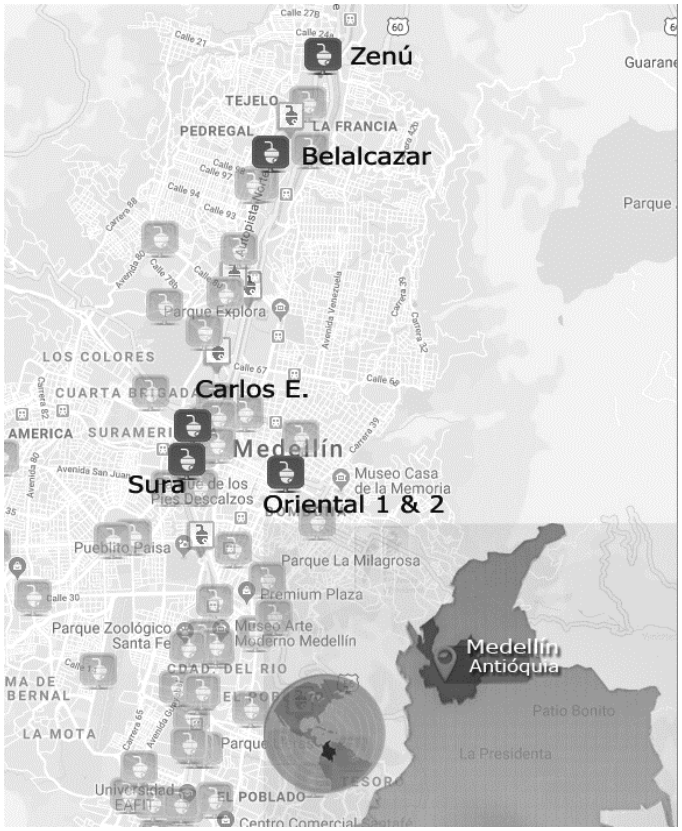

Figure 4. Localisation map of the 80 cameras of the CCTV Secretaría de Movilidad de Medellín [52]. Six cameras are selected for this research. Source: The Authors.

Figure 5. Images examples of the six selected cameras. Each camera covers an important urban zone; from left to right: Belalcazar, Carlos E., Oriental 1 and 2, Zenú and finally Sura. Note the rather poor quality of the images. Source: The Authors.

Table 2.
EspiNet model against Faster-RCNN (VGG16 based) [49] and YOLO V3 [54], comparative results - Results on UMD dataset.

Metrics	EspiNet	Faster RCNN	YOLO V3
Precision (%)	**93.7**	57.3	93.0
Recall (%)	**90.0**	76.3	81.0
F1-score (%)	**91.8**	65.4	86.6
AP (Average Precision)	**88.84**	68.75	80.75

Source: The Authors.

and 91.8% of F1-score [56], which outperforms results for YOLO and Faster R-CNN (VGG16 based). Table 2 shows the comparative results. Fig. 6 presents a graphic comparison of the three models Average Precision (AP).

In all metrics, EspiNet obtains better results than the other two detectors, being YOLO V3 the closest performance. YOLO achieved almost equal precision but a reduced recall since the single stage detector architecture has not Region Proposal Network (RPN), failing to detect too small objects or that appear too close each other.

The results of the detectors applied to the UMD-10K dataset can be seen on https://goo.gl/bJM3HF

5.3. Results on SMMD dataset

On the Secretaría de Movilidad de Medellín dataset (SMMD), we train EspiNet, Faster R-CNN (VGG based) and YOLO V3 end to end using the same proportion of training and evaluating sets of UMD-10k.

Table 3 shows that EspiNet V2 over-perform YOLO V3 and Faster R-CNN in the terms of AP, reaching 79.52 and with a Recall of 83.39. This can be explained again by the absence of RPN in YOLO V3, which fails to detect objects that appear too close or too small. Nevertheless, YOLO V3 can deal better with false detections, outperforming the region based detectors (EspiNet V2 and Faster R-CNN) in terms of Precision, consequently improving the final F1 score. Fig. 7 shows the comparative performance of the three detector in terms of Average Precision (AP).

EspiNet V2 and the Faster R-CNN (VGG 16 based) models were trained on a Windows 10 Machine with a CPU core i7 7th generation 4.7 GHz, with 32 GB of RAM using a NVIDIA Titan X (Pascal) 1531Mhz GPU.

On UMD-10k dataset, the training process of EspiNet V2 model took 32 hours and 47 hours for training Faster R-CNN (VGG 16) model. A Linux machine running Ubuntu 16.04.3, with a Xeon E5-2683 v4 2.10GHz CPU, 64 GB of RAM and a NVIDIA Titan Xp 1582 Mhz GPU was used for training YOLO V3. This model took 18 hours for training on UMD-10k dataset. All models were trained end to end from scratch.

Table 3.
Comparative detection results - Results for the SMMD dataset

Metrics	EspiNet	Faster RCNN	YOLO V3
Precision (%)	65.6	54.6	**85.8**
Recall (%)	**83.3**	82.5	77.6
F1-score (%)	73.43	65.7	**81.5**
AP (Average Precision)	**79.52**	74.96	76.65

Source: The Authors.

The time employed for training the model for the SMMD dataset were 24 hours for EspiNet V2, 35 hours for Faster R-CNN (VGG 16) and 14 hours for YOLO, using the same environments described previously.

6. Conclusions and future work

This paper has introduced EspiNet V2, a model derived from Faster R-CNN, for motorcycle detection in urban scenarios. The model can deal with occluded objects achieving an Average Precision of nearly 90% for UMD-10K, as far as we know the most challenging urban motorbike detection dataset at present. It achieves almost 80% AP in the new SMMD, also a challenging dataset made public for other researchers to improve on these baseline results.

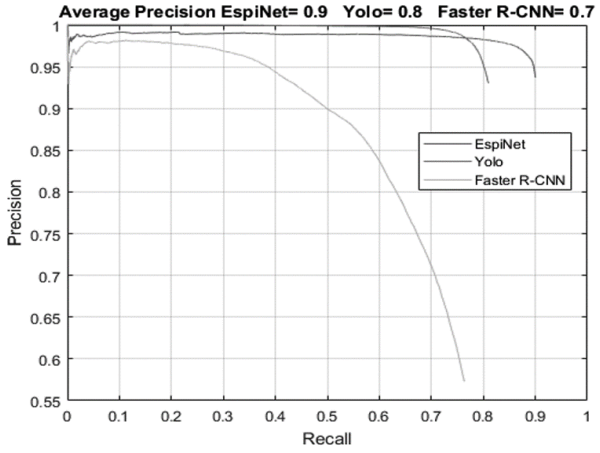

Figure 6 Average Precision (AP) of the model compared with YOLO V3 and Faster R-CNN (VGG16 based). Results on UMD-10K dataset.
Source: The Authors

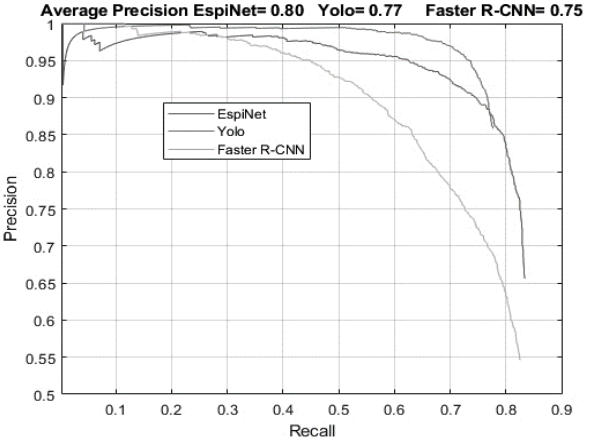

Figure 7 Average Precision (AP) of the model compared with YOLO V3 and Faster R-CNN (VGG16 based). Results on Secretaría de Movilidad de Medellín Dataset (SMMD).
Source: The Authors.

EspiNet V2 and the deep learning detectors models as YOLO V3 and Faster R-CNN (VGG16 based) are compared in this study. The models were trained in the UMD-10k and SMMD datasets, and EspiNet V2 was found to outperform the others in terms of Average Precision (AP).

As per most deep learning architectures, and is also evaluated in [57], the model obtains better results as the number of training examples increases. It is important to have enough representative data for each distribution of examples used for train a deep learning model. The amount and distribution of examples used in these two datasets explain the quality of the classification obtained.

The use of spatio-temporal information could be integrated to the model to improve detection capabilities. EspiNet V2 could be used as a neural network layer that incorporates not only the current time step input information (frame) but also the activation values of previous time steps (previous frames). This architecture corresponds to Recurrent Neural Networks (RNNs) such as Gated Recurrent Units (GRUs) or Long short-term memory (LSTM) which apply sequence modelling for predicting next stages after initial detection according to historical information. This improvement could lead to detection by tracking, where the models can spread their detection class scope to include other urban road users like pedestrians, cyclists, trucks, buses, etc.

Acknowledgements

Sergio A. Velastin is grateful for funding received from the Universidad Carlos III de Madrid, the European Unions Seventh Framework Programme for Research, Technological Development and demonstration under grant agreement N. 600371, el Ministerio de Economia, Industria y Competitividad (COFUND2013-51509) el Ministerio de Educación, Cultura y Deporte (CEI-15-17) and Banco Santander.

This work was partially supported by COLCIENCIAS project: Reduccion de Emisiones Vehiculares Mediante el Modelado y Gestion Optima de Trafico en Areas Metropolitanas - Caso Medellin - Area Metropolitana del Valle de Aburra, codigo 111874558167, CT 049-2017. Universidad Nacional de Colombia- Politécnico Colombiano Jaime Isaza Cadavid. Proyecto HERMES 25374.

The authors also gratefully acknowledge the support of NVIDIA Corporation with the donation of the two GPUs used for this research.

The datasets and code used in this work are available upon request from the authors.

References

[1] WHO, Global status report on road safety, [Online]. 2018, WHO. [Accessed: June 10th of 2019]. Available at: http://www.who.int/violence_injury_prevention/road_safety_status/2018/en/.

[2] Accidentes de tránsito en la Comunidad Andina, 2007-2016, 48 P.

[3] Así Vamos en Salud., Mortalidad por accidentes de tránsito, [Online]. 2018. [Accessed: 2August 23th of 2018]. Available at: https://www.asivamosensalud.org/salud-para-ciudadanos/mortalidad-por-accidentes-de-transito.

[4] RUNT. Estadísticas del RUNT, [Online]. Accessed: August 09th of 2019]. Available at: https://www.runt.com.co/cifras

[5] IDEAM. Calidad del aire, [Online]. [Accessed: August 09th of 2019]. Available at: http://www.ideam.gov.co/web/contaminacion-y-calidad-ambiental/calidad-del-aire.

[6] Walsh, M.P., PM 2.5: global progress in controlling the motor vehicle contribution, Front. Environ. Sci. Eng., 8(1), pp. 1-17, 2014. DOI: 10.1007/s11783-014-0634-4

[7] Ren, S., He, K., Girshick, R. and Sun, J., Faster r-cnn: towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, [online]. 2015, pp. 91-99. Available at: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks

[8] Tian, B. et al., Hierarchical and networked vehicle surveillance in ITS: a survey, IEEE Trans. Intell. Transp. Syst., 18(1), pp. 25-48, 2017. DOI: 10.1109/TITS.2016.2552778

[9] Le, T.S. and Huynh, C.K., An unified framework for motorbike counting and detecting in traffic videos, in: 2015 International Conference on Advanced Computing and Applications (ACOMP), 2015, pp. 162-168. DOI: 10.1109/ACOMP.2015.32

[10] Duan B., Liu W., Fu P., Yang C., Wen X., and Yuan H., Real-time on-road vehicle and motorcycle detection using a single camera, in Industrial Technology, 2009. ICIT 2009. IEEE International Conference on, 2009, pp. 1-6. DOI: 10.1109/ICIT.2009.4939585

[11] Muzammel, M., Yusoff, M.Z. and Meriaudeau, F., Rear-end vision-based collision detection system for motorcyclists, J. Electron. Imaging, 26(3), pp. 033002, 2017. DOI: 10.1117/1.JEI.26.3.033002

[12] Shuo, Y. and Choi, E.-J., A driving support system base on traffic environment analysis, Indian J. Sci. Technol., 9(47), 2016. DOI: 10.17485/ijst/2016/v9i47/108374

[13] Wonghabut, P., Kumphong, J., Satiennam,, T., Ung-arunyawee R. and Leelapatra, W., Automatic helmet-wearing detection for law enforcement using CCTV cameras, in: IOP Conference Series: Earth and Environmental Science, 2018, 143, pp. 012063. DOI: 10.1088/1755-1315/143/1/012063

[14] Dahiya, K., Singh, D. and Mohan, C.K., Automatic detection of bike-riders without helmet using surveillance videos in real-time, in: 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 3046-3051. DOI: 10.1109/IJCNN.2016.7727586

[15] Singh, D., Vishnu, C. and Mohan, C.K., Visual big data analytics for traffic monitoring in smart city, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 886-891. DOI: 10.1109/ICMLA.2016.0159

[16] e Silva, R.R., Aires, K.R. and Veras, R. de MS, Detection of helmets on motorcyclists, Multimed. Tools Appl., 77(5), pp. 5659-5683, 2017. DOI: 10.1007/s11042-017-4482-7

[17] Wu, H. and Zhao, J., An intelligent vision-based approach for helmet identification for work safety, Comput. Ind., 100, pp. 267-277, 2018. DOI: 10.1016/j.compind.2018.03.037

[18] Messelodi, S., Modena C.M. and Cattoni, G., Vision-based bicycle/motorcycle classification, Pattern Recognit. Lett., 28(13), pp. 1719-1726, 2007. DOI: 10.1016/j.patrec.2007.04.014

[19] Buch, N., Orwell, J. and Velastin, S.A., Urban road user detection and classification using 3D wire frame models, IET Comput. Vis., 4(2), pp. 105-116, 2010. DOI: 10.1049/iet-cvi.2008.0089

[20] Chiu, C.-C., Ku, M.-Y. and Chen, H.-T., Motorcycle detection and tracking system with occlusion segmentation, in: Image Analysis for Multimedia Interactive Services, 2007. WIAMIS07. Eighth International Workshop on, 2007, pp. 32-32. DOI: 10.1109/WIAMIS.2007.60

[21] Ku, M.-Y., Chiu, C.-C., Chen, H.-T. and Hong, S.-H., Visual motorcycle detection and tracking algorithms, WSEAS Trans. Electron., [online]. pp. 121-131, 2008. Available at: http://www.wseas.us/e-library/transactions/electronics/2008/30-863.pdf

[22] Stauffer, C. and Grimson, W.E.L., Adaptive background mixture models for real-time tracking, in: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., 1999, pp. 246-252. DOI: 10.1109/CVPR.1999.784637

[23] Waranusast, R., Bundon, N., Timtong, V., Tangnoi, C. and Pattanathaburt, P., Machine vision techniques for motorcycle safety helmet detection, in: 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), 2013, pp. 35-40. DOI: 10.1109/IVCNZ.2013.6726989

[24] Rashidan, M.A., Mustafah, Y.M., Shafie, A.A., Zainuddin, N.A., Aziz, N.N.A. and Azman, A.W., Moving object detection and classification using Neuro-Fuzzy approach, Int. J. Multimed. Ubiquitous Eng., 11(4), pp. 253-266, 2016. DOI: 10.14257/ijmue.2016.11.4.26

[25] Chen, Z. and Ellis, T., Self-adaptive Gaussian mixture model for urban traffic monitoring system, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1769-1776. DOI: 10.1109/ICCVW.2011.6130463

[26] Chen, Z., Ellis, T. and Velastin, S.A., Vehicle detection, tracking and classification in urban traffic, in: 15th International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 951-956. DOI: 10.1109/ITSC.2012.6338852

[27] Chiverton, J., Helmet presence classification with motorcycle detection and tracking, Intell. Transp. Syst. IET, 6(3), pp. 259-269, 2012. DOI: 10.1049/iet-its.2011.0138

[28] Thai, N.D., Le, T.S., Thoai, N. and Hamamoto, K., Learning bag of visual words for motorbike detection, in: 13th International Conference on Control Automation Robotics Vision (ICARCV), 2014, pp. 1045-1050. DOI: 10.1109/ICARCV.2014.7064450

[29] Mukhtar, A. and Tang, T.B., Vision based motorcycle detection using HOG features, in: IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2015, pp. 452-456. DOI: 10.1109/ICSIPA.2015.7412234

[30] Dupuis, Y., Subirats, P. and Vasseur, P., Robust image segmentation for overhead real time motorbike counting, in: IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 3070-3075. DOI: 10.1109/ITSC.2014.6958183

[31] Sutikno, S., Waspada, I., Bahtiar, N. and Sasongko, P.S., Classification of motorcyclists not wear helmet on digital image with backpropagation Neural Network, TELKOMNIKA Telecommun. Comput. Electron. Control, 14(3), pp. 1128-1133, 2016. DOI: 10.12928/telkomnika.v14i3.3486

[32] Vishnu, C., Singh, D., Mohan, C.K. and Babu, S., Detection of motorcyclists without helmet in videos using convolutional neural network, in: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 3036-3041. DOI: 10.1109/IJCNN.2017.7966233

[33] Espinosa, J.E., Velastin, S.A. and Branch, J.W., Vehicle detection using Alex Net and Faster R-CNN deep learning models: a comparative study, in: International Visual Informatics Conference, 2017, pp. 3-15. DOI: 10.1007/978-3-319-70010-6_1

[34] Adu-Gyamfi, Y.O., Asare, S.K., Sharma, A. and Titus, T., Automated vehicle recognition with deep convolutional Neural Networks, Transportation Research Record: Journal of the Transportation Research Board 2645(1), pp. 113-122, 2017. DOI: 10.3141/2645-13

[35] Huynh, C.K., Le, T.S. and Hamamoto, K., Convolutional neural network for motorbike detection in dense traffic, in: IEEE Sixth International Conference on Communications and Electronics (ICCE), 2016, pp. 369-374. DOI: 10.1109/CCE.2016.7562664

[36] Ra,j K.C.D., Chairat, A., Timtong, V., Dailey, M.N. and Ekpanyapong, M., Helmet violation processing using deep learning, in: International Workshop on Advanced Image Technology (IWAIT), 2018, pp. 1-4. DOI: 10.1109/IWAIT.2018.8369734

[37] Wu, H. and Zhao, J., Automated visual helmet identification based on deep convolutional neural networks, in: Computer Aided Chemical Engineering, 44, Elsevier, 2018, pp. 2299-2304. DOI: 10.1016/B978-0-444-64241-7.50378-5

[38] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., ImageNet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009 - CVPR 2009, 2009, pp. 248-255. DOI: 10.1109/CVPR.2009.5206848

[39] Zeiler, M.D. and Fergus, R., Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818-833. DOI: 10.1007/978-3-319-10590-1_53

[40] Lampert, C.H., Blaschko, M.B. and Hofmann, T., Efficient subwindow search: a branch and bound framework for object localization, IEEE Trans. Pattern Anal. Mach. Intell., 31(12), pp. 2129-2142, 2009. DOI: 10.1109/TPAMI.2009.144

[41] Uijlings, J.R., Van De Sande, K.E., Gevers, T. and Smeulders, A.W., Selective search for object recognition, Int. J. Comput. Vis., 104(2), pp. 154-171, 2013. DOI: 10.1007/s11263-013-0620-5

[42] He, K., Zhang, X., Ren, S. and Sun, J., Spatial pyramid pooling in deep convolutional Networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell., 37(9), pp. 1904-1916, 2015. DOI: 10.1109/TPAMI.2015.2389824

[43] Zitnick, C.L. and Dollár, P., Edge boxes: locating object proposals from edges, in: European Conference on Computer Vision, 2014, pp. 391-405. DOI: 10.1007/978-3-319-10602-1_26

[44] Girshick, R., Donahue, J., Darrell, T. and Malik, J., Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587. DOI: 10.1109/CVPR.2014.81

[45] Girshick, R., Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440-1448. DOI: 10.1109/ICCV.2015.169

[46] Fan, Q., Brown, L. and Smith, J., A closer look at Faster R-CNN for vehicle detection, in: IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 124-129. DOI: 10.1109/IVS.2016.7535375

[47] Geiger, A., Lenz, P. and Urtasun, R., Are we ready for autonomous driving?. The kitti vision benchmark suite, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 3354-3361. DOI: 10.1109/CVPR.2012.6248074

[48] Espinosa, J.E., Velastin, S.A. and Branch, J.W., Motorcycle detection and classification in urban Scenarios using a model based on Faster R-CNN, in: 9th International Conference on Pattern Recognition Systems (ICPRS 2018), 2018, 6 P., ArXiv180802299 Cs, 2018. DOI: 10.1049/cp.2018.1292

[49] Huang, J. et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. ArXiv161110012 Cs, 2017. DOI: 10.1109/CVPR.2017.351

[50] Donahue, J. et al., DeCAF: A deep convolutional activation feature for generic visual recognition, in: ICML, [online]. 2014, pp. 647-655. Available at http://www.jmlr.org/proceedings/papers/v32/donahue14.pdf

[51] Romanuke, V.V., Appropriate number of standard 2 X 2 max pooling layers and their allocation in convolutional neural networks for diverse and heterogeneous datasets, Inf. Technol. Manag. Sci., 20(1), pp. 12-19, 2017. DOI: 10.1515/itms-2017-0002

[52] SIMM. Cámaras de CCTV. [Online]. [Accessed: October 31st of 2018]. Available at: https://www.medellin.gov.co/simm/camaras-de-circuito-cerrado.

[53] Everingham, M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A., The pascal visual object classes (voc) challenge, Int. J. Comput. Vis., 88(2), pp. 303-338, 2010. DOI: 10.1007/s11263-009-0275-4

[54] Redmon, J. and Farhadi, A., YOLOv3: an incremental improvement, Tech. Report, in: Computer Vision and Pattern Recognition (cs.CV), [online]. 2018, 6 P. ArXiv180402767 Cs, Available at: http://arxiv.org/abs/1804.02767

[55] Ng, A., Machine learning yearning, URL Httpwww Mlyearning Org96, 2017.

[56] Yin, F., Makris, D. and Velastin, S.A., Performance evaluation of object tracking algorithms, in: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil, [online]. 2007. Available at: https://pdfs.semanticscholar.org/ad76/bdc7d06a7ec496ac788d667c6ad5fcc0fe41.pdf

[57] Espinosa-Oviedo, J.E., Detection and tracking of motorcycles in urban environments by using video sequences with high level of oclussion, PhD Thesis, Universidad Nacional de Colombia, Medellín campus, Medellín, Colombia, 2019.

J.E. Espinosa-Oviedo, was born in Bogotá D.C., Colombia, in 1973. He received the BSc. in System Engineering in 2001, from the Universidad Los Libertadores de Colombia, and MSc. in Artificial Intelligence in 2003, from the Katholieke Universiteit Leuven, Belgium. Currently, he is candidate to PhD. degree in Systems Engineering from Universidad Nacional de Colombia. From 2010, he has been a full-time professor at the Politécnico

Colombiano Jaime Isaza Cadavid, Medellín Colombia. He is teaching in areas as programming and artificial intelligence. In the last nine years as a professor, he has scientific publications in national and international journals and congresses, mostly related to his research theme: optimization, artificial intelligence, applied computer vision and related areas.
ORCID: 0000-0002-0494-1276

S.A. Velastin-Carroza, (M90, SM12) received the BSc. and MSc. (Research) in Electronics and the PhD. in 1978, 1979, and 1982, respectively, from the University of Manchester, Manchester, U.K., for research on vision systems for pedestrian tracking and road-traffic analysis. He worked in industrial R&D before joining Kings College London, University of London (UK) in 1991 and then Kingston University London where he became director of its Digital Imaging Research Centre and full professor of applied computer vision. In 2013 he became research professor at the University of Santiago Chile, and in 2015 he moved to the University Carlos III of Madrid, Spain where he was a Marie Curie Professor. He has worked in many EU-funded projects and is also a Fellow of the IET.
ORCID: 0000-0001-6775-1737

J.W. Branch-Bedoya, received the BSc. in Mining Engineering, his MSc. in System Engineering and his PhD. in Engineering in 1995, 1997 and 2007 respectively, all of tehm from the Universidad Nacional de Colombia, Campus Medellin, Currently, he is a full professor in the Department of Computer Science at Universidad Nacional de Colombia, Campus Medellin. His main research interests encompass computer vision, image processing and their applications to the industry field and applications of pattern recognition
ORCID: 0000-0002-0378-028X