TEC2017-88169-R MobiNetVideo (2018-2020-2021)

Visual Analysis for Practical Deployment of Cooperative Mobile Camera Networks

D3 v2

Technologies for Mobile Camera Networks

Video Processing and Understanding Lab

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Supported by

AUTHORS LIST

José M. Martínez	josem.martinez@uam.es
Álvaro García Martín	alvaro.garcia@uam.es
Marcos Escudero Viñolo	marcos.escudero@uam.es
Pablo Carballeira López	pablo.carballeira@uam.es
Juan C. San Miguel Avedillo	juancarlos.sanmiguel@uam.es
Elena Luna García	elena.luna@uam.es
Alejandro Lopez Cifuentes	alejandro.lopezc@uam.es
Paula Moral de Eusebio	paula.moral@uam.es
Jesús Bescós Cano	j.bescos@uam.es
Miguel Ángel García García	miguelangel.garcia@uam.es

HISTORY

Version	**Date**	**Editor**	**Description**
0.1	31/07/2019	José M. Martínez	Initial draft version
0.2	23/08/2019	Álvaro García-Martín	People/Car re-identification approach
0.3	04/05/2019	Elena Luna	Cooperative detection and tracking
0.4	05/05/2019	Juan Carlos San Miguel	Review of Cooperative detection and tracking
0.5	13/09/2019	Alejandro López	Scene Recognition and Semantic Segmentation
0.6	27/09/2019	Marcos Escudero	Scene Recognition and Semantic Segmentation
0.7	30/09/2019	José M. Martínez	Editorial checking
1.0	30/09/2019	José M. Martínez	First version
1.1	10/10/2020	José M. Martínez	Structure for Contributions and draft for several sections
1.2	18/11/2020	Marcos Escudero Alejandro López Jesús Bescós	Scene Recognition and Semantic Segmentation
1.3	04/12/2020	Álvaro García-Martín	Multi-view matching: Aicity 2020 / attributes

		Paula Moral	
1.4	15/01/2021	Elena Luna	Extensions for cooperative multi-target detection and tracking (section 5.2)
1.5	18/01/2021	Juan Carlos San Miguel	New section 3.3 "Early experiments for semantic segmentation on synthetic data‟ and Review of Cooperative detection and tracking (section 5)
1.6	24/01/2021	Pablo Carballeira	New section 5.3 "People detection in omnidirectional cameras"
1.7	28/01/2021	Miguel Ángel García	New scene recognition sections 2.3, 2.4 and 2.5
1.8	01/02/2021	José M.Martínez	Editorial checking
2.0	01/02/2021	José M.Martínez	Second version version

CONTENTS:

1. Introduction

1.1. Motivation

Work package 3 (WP3) aims at proposing new technologies for applications related to heterogeneous camera networks where camera mobility plays a key role. Such proposals will be performed on public datasets. If required, small scenarios will be recorded.

This deliverable describes the work related with tasks T.3.1 Scene Recognition, T3.2 Semantic Segmentation, T3.3 Multi-view matching and T.3.4 Cooperative detection and tracking

1.2. Document structure

This document contains the following chapters:

- Chapter 1: Introduction to this document
- Chapter 2: Scene Recognition
- Chapter 3: Semantic Segmentation
- Chapter 4: Multi-view matching
- Chapter 5: Cooperative detection and tracking
- Chapter 6: Conclusions

2. Scene Recognition

2.1. Semantic-Aware Scene Recognition Approach [1]

2.1.1. Design

Scene recognition is currently one of the top-challenging research fields in computer vision. This may be due to the ambiguity between classes: images of several scene classes may share similar objects, which causes confusion among them. The problem is aggravated when images of a scene class are notably different. Convolutional Neural Networks (CNNs) have significantly boosted performance in scene recognition, albeit it is still far below from other recognition tasks (e.g., object or image recognition). In this paper, we describe a novel approach for scene recognition based on an end-to-end multi-modal CNN that combines image and context information by means of an attention module. Context information, in the shape of a semantic segmentation, is used to gate features extracted from the RGB image by leveraging on information encoded in the semantic representation: the set of scene objects and stuff, and their relative locations. This gating process reinforces the learning of indicative scene content and enhances scene disambiguation by refocusing the receptive fields of the CNN towards them. Experimental results on four publicly available datasets show that the proposed approach outperforms every other state-of-the-art method while significantly reducing the number of network parameters.

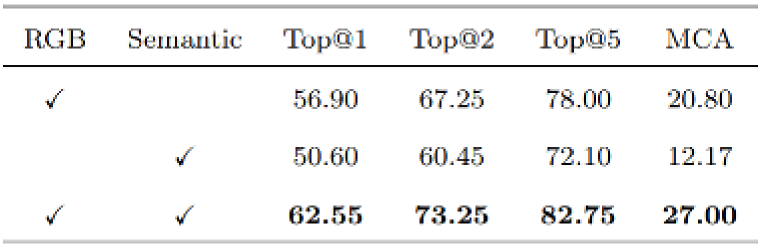
2.1.1. Experimental results

The proposed solution is validated by an extensive comparison with the state-of-the art using four publicly available datasets described in [2]. The following Tables illustrate this comparison. A brief discussion is included for each dataset. See full details in [1].

Results on the ADE20K Dataset from Table 1 indicate the effectiveness of the proposed architecture when compared to the solely use of either the RGB features or the Semantic features. When using both RGB and Semantic features, increments of a 9.9% and a 29.80% in terms of Top@1 accuracy and Mean Class Accuracy are obtained whit respect to the RGB baseline.

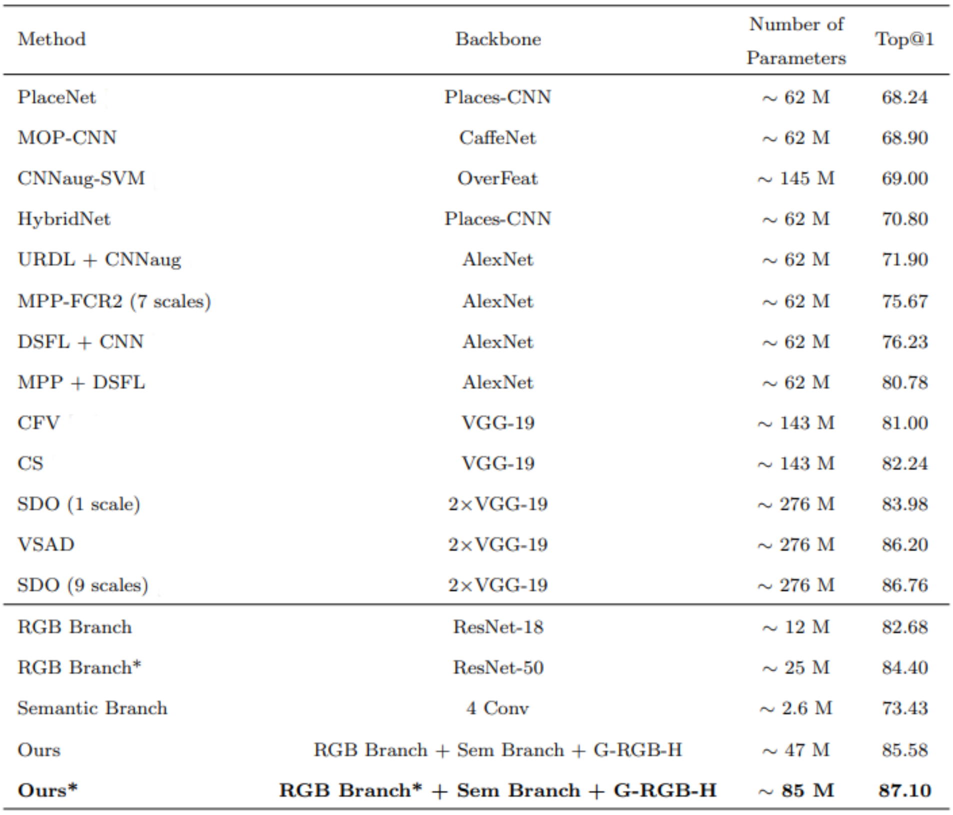
Results from Table 2 and Table 3 indicate that the proposed method outperforms every other scene recognition state-of-the-art algorithm. Specifically, the proposed algorithm using ResNet-

50 as backbone (Ours*) outperforms SDO [4], an algorithm similar in spirit, in a 0.39% and a 0.85% for MIT Indoor 67 [5] and SUN 397 [6] respectively.



RGB	Semantic	Top@1	Top@2	Top@5	MCA
✓		56.90	67.25	78.00	20.80
	✓	50.60	60.45	72.10	12.17
✓	✓	**62.55**	**73.25**	**82.75**	**27.00**

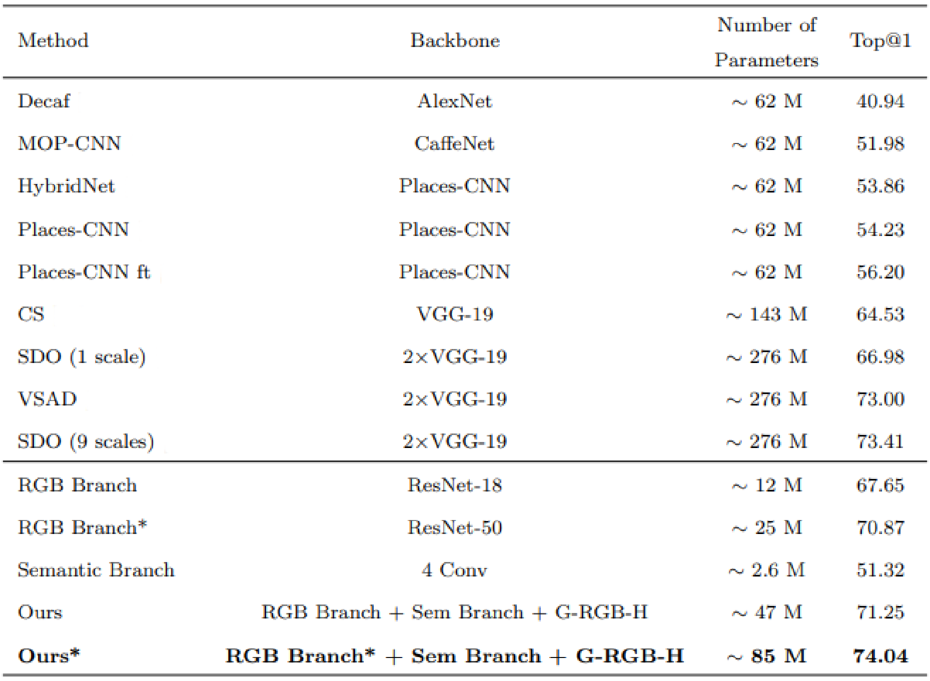
Table 1. Scene recognition results on ADE20K



Method	Backbone	Number of Parameters	Top@1
PlaceNet	Places-CNN	~ 62 M	68.24
MOP-CNN	CaffeNet	~ 62 M	68.90
CNNaug-SVM	OverFeat	~ 145 M	69.00
HybridNet	Places-CNN	~ 62 M	70.80
URDL + CNNaug	AlexNet	~ 62 M	71.90
MPP-FCR2 (7 scales)	AlexNet	~ 62 M	75.67
DSFL + CNN	AlexNet	~ 62 M	76.23
MPP + DSFL	AlexNet	~ 62 M	80.78
CFV	VGG-19	~ 143 M	81.00
CS	VGG-19	~ 143 M	82.24
SDO (1 scale)	2×VGG-19	~ 276 M	83.98
VSAD	2×VGG-19	~ 276 M	86.20
SDO (9 scales)	2×VGG-19	~ 276 M	86.76
RGB Branch	ResNet-18	~ 12 M	82.68
RGB Branch*	ResNet-50	~ 25 M	84.40
Semantic Branch	4 Conv	~ 2.6 M	73.43
Ours	RGB Branch + Sem Branch + G-RGB-H	~ 47 M	85.58
Ours*	**RGB Branch* + Sem Branch + G-RGB-H**	**~ 85 M**	**87.10**

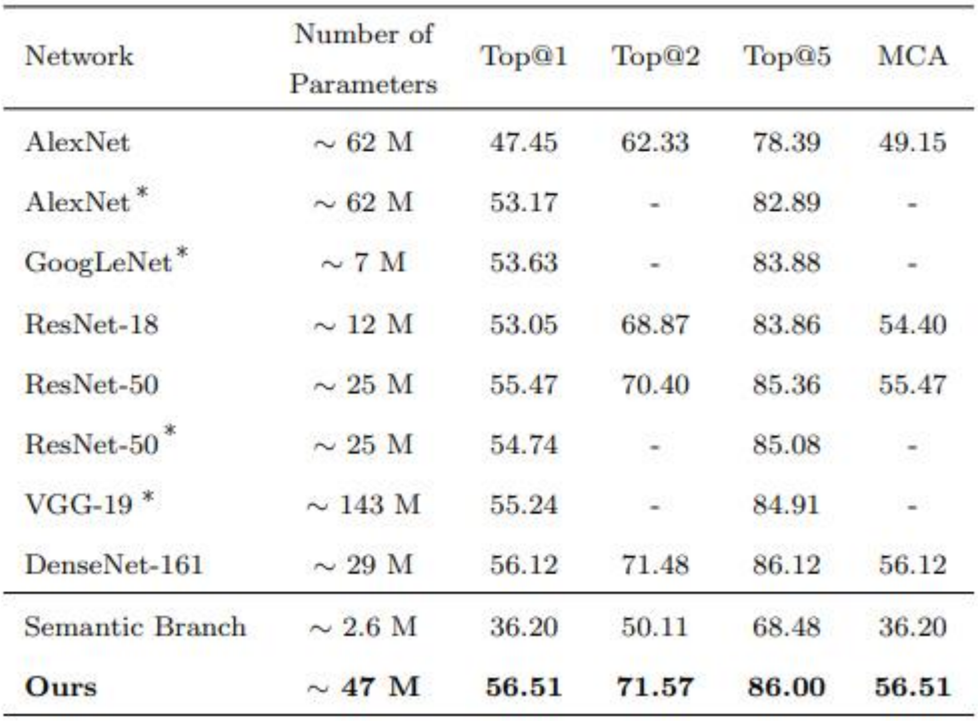
Table 2. State-of-the-art results on MIT Indoor 67 dataset. Methods using objects to drive scene recognition include: [13, 14], Semantic Branch, Ours and Ours*.

Results from Table 4 compare the proposed algorithm with respect to state-of-the-art Convolutional Neural Networks on Places Dataset [7]. “Ours” obtains the best results from the table while maintaining relatively low complexity. Its performance improves those of the deepest network, DenseNet-161, by a 0.73% in terms of Top@1 accuracy and it surpasses the most complex network, VGG-19, by a 2.29% reducing the number of parameters a 67.13%.



Method	Backbone	Number of Parameters	Top@1
Decaf	AlexNet	~ 62 M	40.94
MOP-CNN	CaffeNet	~ 62 M	51.98
HybridNet	Places-CNN	~ 62 M	53.86
Places-CNN	Places-CNN	~ 62 M	54.23
Places-CNN ft	Places-CNN	~ 62 M	56.20
CS	VGG-19	~ 143 M	64.53
SDO (1 scale)	2×VGG-19	~ 276 M	66.98
VSAD	2×VGG-19	~ 276 M	73.00
SDO (9 scales)	2×VGG-19	~ 276 M	73.41
RGB Branch	ResNet-18	~ 12 M	67.65
RGB Branch*	ResNet-50	~ 25 M	70.87
Semantic Branch	4 Conv	~ 2.6 M	51.32
Ours	RGB Branch + Sem Branch + G-RGB-H	~ 47 M	71.25
Ours*	**RGB Branch* + Sem Branch + G-RGB-H**	**~ 85 M**	**74.04**

Table 3. State-of-the-art results on SUN 397 dataset. Methods using objects to drive scene recognition include: [13, 14], Semantic Branch, Ours and Ours*.



Network	Number of Parameters	Top@1	Top@2	Top@5	MCA
AlexNet	~ 62 M	47.45	62.33	78.39	49.15
AlexNet *	~ 62 M	53.17	-	82.89	-
GoogLeNet *	~ 7 M	53.63	-	83.88	-
ResNet-18	~ 12 M	53.05	68.87	83.86	54.40
ResNet-50	~ 25 M	55.47	70.40	85.36	55.47
ResNet-50 *	~ 25 M	54.74	-	85.08	-
VGG-19 *	~ 143 M	55.24	-	84.91	-
DenseNet-161	~ 29 M	56.12	71.48	86.12	56.12
Semantic Branch	~ 2.6 M	36.20	50.11	68.48	36.20
Ours	**~ 47 M**	**56.51**	**71.57**	**86.00**	**56.51**

Table 4. State-of-the-art results on Places-365 Dataset (%). (* stands for performance metrics reported in the dataset).

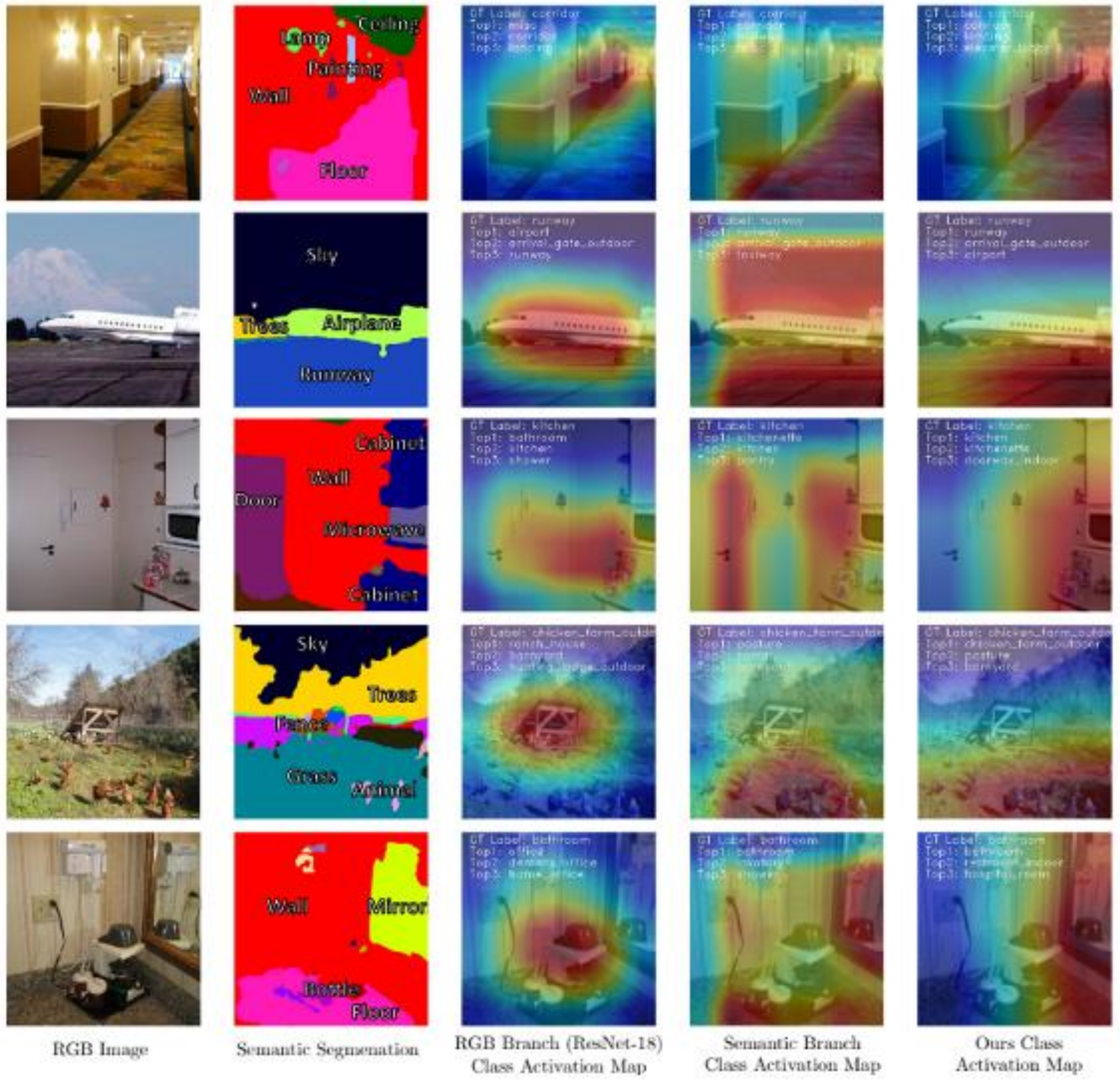


Figure 1. Qualitative results.

First and second column represent the RGB and semantic segmentation images from the ADE20K, the SUN 397 and the Places 365 validation sets. The third, fourth and fifth columns depict the Class Activation Map (CAM) obtained by using features extracted from: the RGB Branch used baseline (ResNet-18), the Semantic Branch and the proposed method (Ours). The CAM represents the image areas that produce a greater activation of the network. CAM images also indicate the ground-truth label and the Top 3 predictions. It can be observed how the proposed method changes the attention towards human-accountable concepts that can be indicative of the scene class, e.g., the microwave for the kitchen, the animals for the chicken farm or the mirror for the bathroom.

2.2. Egocentric Scene Recognition combining depth, color and semantic information [71]

2.2.1. Design

A system for RGB-D scene recognition is designed. We show that using depth maps can further improve the results, since the depth possesses additional cues, not very likely to be learnt from colour data. We define a two-stage learning architecture consisting of three branches—colour, depth and semantic, fused in the end using attention mechanisms. Each branch is firstly maximized in terms of precision on its own. In this case, we show that the proper encoding is crucial for the depth branch and that HHA (Horizontal disparity, Height, Angle) representation leads to the best results. Moreover, we show that proper pre-training makes a great difference when fine-tuning to small datasets. After all branches have been optimized, weights inside them are frozen and different attention modules are trained and evaluated. In the end, using Hadamard combination proved to be the most prolific. Finally, we reached performances comparable to the current state of the art methods, resulting in a 60.0% Top@1 precision in the SUN RGB-D dataset. We also provide an extensive quantitative and qualitative evaluation of our model.

2.2.2. Experimental results

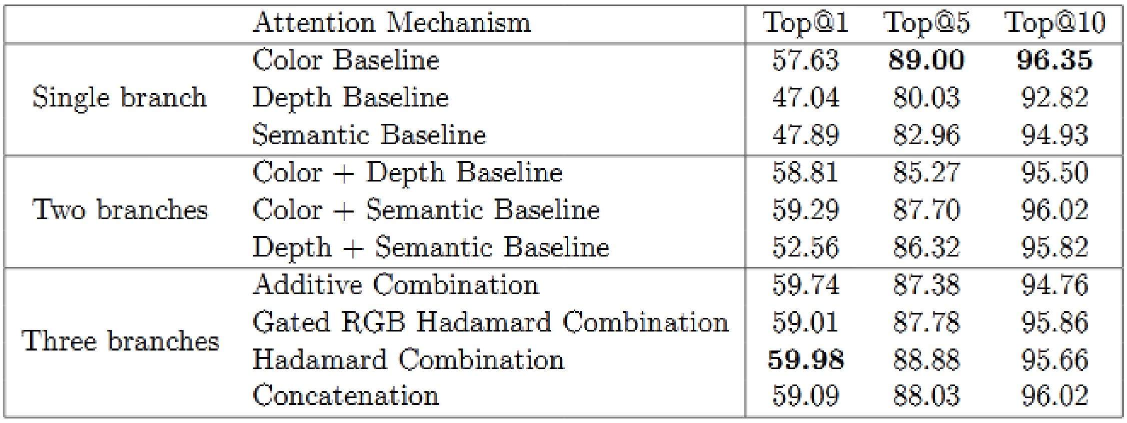
The proposed solution is validated by an extensive comparison with the state-of-the art using the SUN RGB-D dataset [72]. The following Figures and Tables illustrate this comparison. We here include results for the fusion mechanisms when the complete architecture (three branches is used). See full details, including ablation studies and experiments on the effect of the hyperparameters in [71].

Fusion

Fusion between branches is achieved using different attention modules. As aforementioned, all three branches are firstly pretrained separately. Afterwards, their weights are frozen, and the classifiers are discarded from the branches. Two convolutional blocks are appended to each of the branches, in order to extract features relevant for fusion. In the end, fusion is realized as either an element-wise function or a feature concatenation.

Extensive study on the impact of attention mechanisms on results is presented in Table 5. In order to better understand the results that attention mechanisms achieve, the results of two branch architectures are also presented. Fusing colour and semantic branches is carried out by using Gated RGB Hadamard Combination, as noted in [1], while fusion of colour and depth and depth and semantic information is done using Hadamard Combination. The results show that all attention mechanisms manage to improve on the Top@1 results achieved by colour baseline. Yet, not all of them manage to improve on the baseline set by the Semantic-Aware Scene Parsing Network [1], combining RGB and semantic information.

In the end, the best result was achieved when Hadamard Combination attention mechanism was used. Hadamard Combination mechanism computes element-wise product between the features extracted from all-three branches and feeds it further. Even though it managed to improve on Top@1 result, colour baseline remained having better scores on Top@5 and Top@10 metrics.

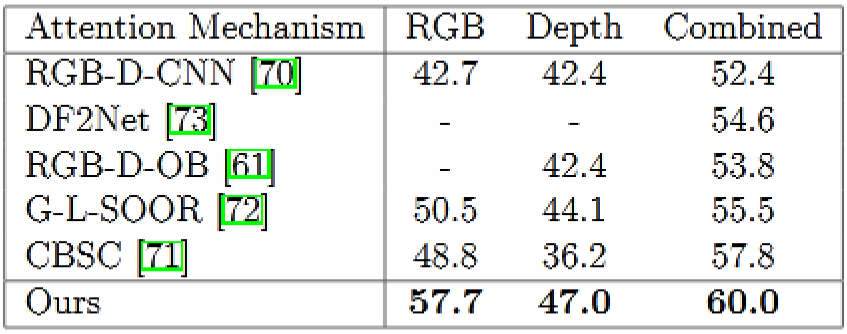


	Attention Mechanism	Top@1	Top@5	Top@10
Single branch	Color Baseline	57.63	**89.00**	**96.35**
	Depth Baseline	47.04	80.03	92.82
	Semantic Baseline	47.89	82.96	94.93
Two branches	Color + Depth Baseline	58.81	85.27	95.50
	Color + Semantic Baseline	59.29	87.70	96.02
	Depth + Semantic Baseline	52.56	86.32	95.82
Three branches	Additive Combination	59.74	87.38	94.76
	Gated RGB Hadamard Combination	59.01	87.78	95.86
	Hadamard Combination	**59.98**	88.88	95.66
	Concatenation	59.09	88.03	96.02

Table 5. **Comparison of proposed attention mechanisms in terms of accuracy on SUN RGB-D dataset**

Comparison against the State-of-the-art

By obtaining 60.0% accuracy, our network surpassed the previous state of the art in the task of RGB-D Scene recognition on SUN RGB-D dataset. The comparison between the previous proposals and our results is shown in Table 6.



Attention Mechanism	RGB	Depth	Combined
RGB-D-CNN [70]	42.7	42.4	52.4
DF2Net [73]	-	-	54.6
RGB-D-OB [61]	-	42.4	53.8
G-L-SOOR [72]	50.5	44.1	55.5
CBSC [71]	48.8	36.2	57.8
Ours	**57.7**	**47.0**	**60.0**

Table 6. **Comparison of proposed architecture with the state of the art approachesin terms of accuracy on SUN RGB-D dataset**

Per-class improvement.

The bar plot representing the per-class percentage Top@1 improvement obtained by incorporating additional branches to the colour baseline model is shown in Figure 2. It can be noticed that in 11 classes incorporating depth and semantic branches resulted in higher precision rate, while the results decreased for only 4 classes. The highest improvement occurs in corridor class. The explanation for this comes from the fact that corridors contain a very specific depth pattern. In contrary to the other rooms, that most often have square shape, corridors are usually very narrow and long. Hence, having an information about a long and continuous patch of increasing depth helps the network to classify it as a corridor. Three examples in which using additional branches helped to produce better predictions in corridor images are presented in Figure 3.

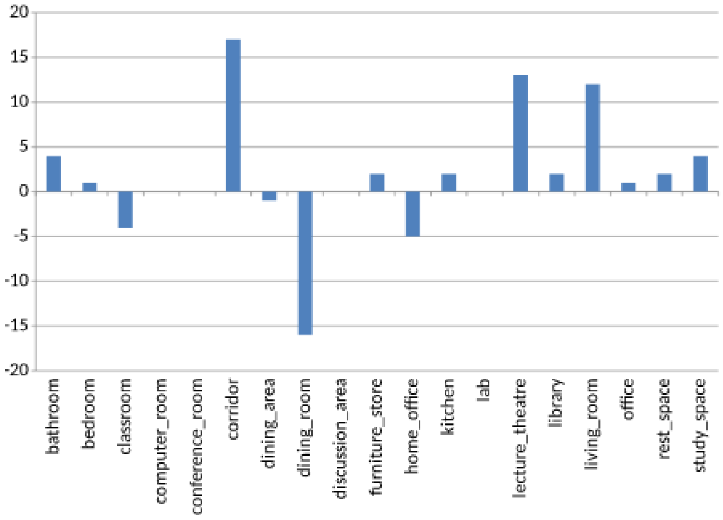


Figure 2. Per-class percentage Top@1 improvement by incorporating depth and semantic information.

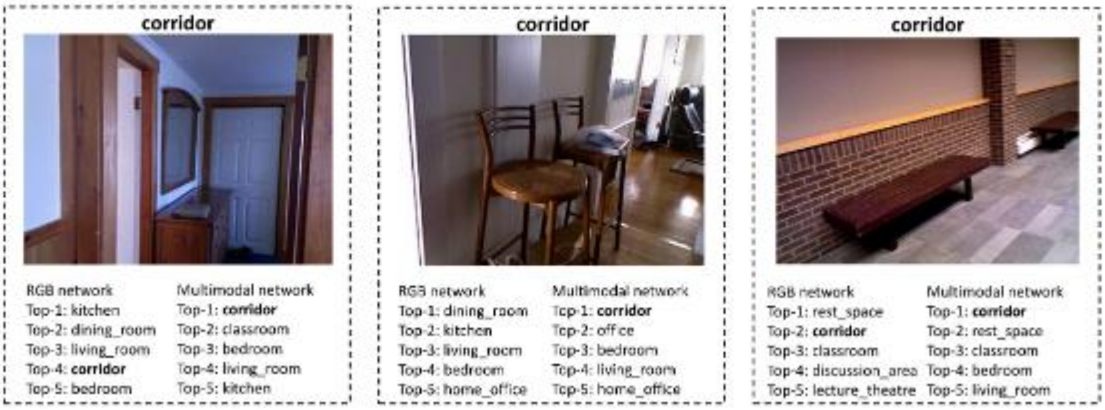


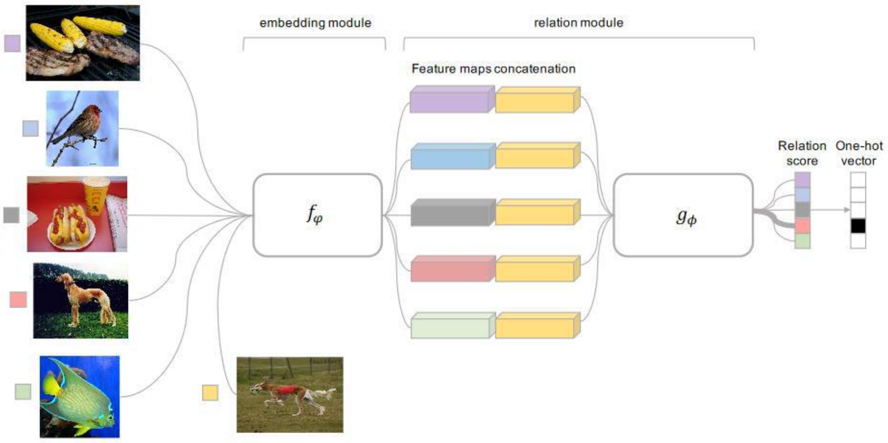
Figure 3. Examples showing the improvement made in images of corridors by incorporating depth and semantic information

2.3. Image classification through reduced training sets and “few-shot” learning [99]

2.3.1. Design

Conventional training of deep convolutional neural networks typically relies on the availability of millions of labelled images. Having access to such huge image repositories is not realistic for many applications in which reduced datasets are only available for training. “Few-shot” learning aims at training deep neural networks with reduced training sets.

This work has analysed both the “one-shot” (one training image per class) and “few-shot” (few training images per class) learning paradigms by implementing and evaluating the “relation network”, a deep neural network described in [100] (CVPR 2018). This network allows the supervised classification of images through reduced training sets. The network consists of two convolutional modules shown in the next figure for the “one-shot” instance and considering the particular example of 5 classes (5 ways):



The “embedding module” extracts a feature vector from an input image. In the previous example, it is applied to 6 images: the test image (bottom right) and the single training image (one shot) associated with each of the 5 classes (left column). The feature vector corresponding to each training image is concatenated with the feature vector obtained for the test image. The subsequent “relation module” generates a numerical score for each pair of feature vectors that amounts for the similarity between them. The test image is classified as belonging to the class with the largest similarity score. In “N-shot” learning, the feature vectors corresponding to the N training images per class are simply added and the result concatenated to the feature vector of the test image.

2.3.2. Experimental results

The aforementioned relation network has been implemented and tested on the particular problem of classifying images of both the public dataset *miniImageNet* used in [100] and of a proprietary dataset with outdoor images of buildings from the UAM campus. In both cases we have considered 5 classes (5 ways) and 20 images per class: either 1 or 5 images for training (sample images) and the others for testing (query images). For example, the figure below shows an example of the 20 images considered for one of the classes of the UAM dataset in the 5-shot experiment: the 5 images in the first row were used for training and the other 15 for testing:

The analyzed network has achieved results comparable to the ones reported in [100] for *miniImageNet* (around 45% classification accuracy for 1-shot and 60% for 5-shot). In turn, the classification accuracy for the scene dataset has been around 58% for 1-shot and 78% for 5-shot. Although these efficiencies are still far away from those yielded by complex state-of-the-art networks trained with millions of images, the results obtained especially for the 5-shot case prove that “few-shot” learning is a promising technology that can be advantageous for many applications.

2.4. Scene recognition using deep neural networks trained with the PLACES database [101]

2.4.1. Design

Deep convolutional neural networks are extensively used in computer vision nowadays. Most well-known backbone network models are already implemented in public frameworks (e.g., PyTorch, TensorFlow, Keras) and pre-trained with millions of images belonging to public datasets (e.g., ImageNet, COCO, PLACES). PLACES365 is a dataset with millions of outdoor images belonging to 365 different scenes. It is specifically targeted to scene recognition rather than to object recognition.

This work has evaluated the performance of two state-of-the-art networks (ResNet18 and ResNet50) pre-trained with PLACES365 when applied to the supervised classification of outdoor images belonging to a particular collection of scenes (classes) of interest.

In order to apply a pre-trained neural network to classifying images belonging to a set of classes different from the ones used for its original training, it is necessary to extend that network with a new block of fully connected layers that generates as many outputs as classes of interest, and then to retrain the extended network with the new training examples. This is known as “transfer learning.”

2.4.2. Experimental results

This work has evaluated the performance of public implementations of both the ResNet18 and ResNet50 state-of-the-art networks pre-trained with PLACES365 when applied to the classification of a proprietary dataset of scene images belonging to the UAM campus.

In particular, the ResNet18 and ResNet50 implementations provided in the PyTorch framework and pre-trained with the PLACES365 dataset were extended in order to classify scenes belonging to 10 different categories from the UAM campus, including outside views of different campus buildings and sport utilities. The extended networks were retrained with 25 different training images per class (data augmentation was applied to those images) and evaluated with other 10 test images per class. Some examples of those images are shown below:

The average classification accuracy obtained in those experiments was 85%. However, although the classification performance for 7 classes was above 80% (with 3 classes scoring more than 90%), there were 3 classes with relatively low accuracies of 60% and 70%. In most cases, the reason for that poor behaviour was the overall visual resemblance of images belonging to different classes.

As a conclusion of this study, the use of properly pre-trained state-of-the-art networks is not sufficient for guaranteeing the proper recognition of specific scenes, even in a simple problem such as the one targeted in this work. Indeed, classification based on the analysis of images as a whole is not reliable enough and should also take into account details present in the images (objects, parts of objects, …) for disambiguation purposes. This opens interesting new research lines in scene recognition.

2.5. Unsupervised scene recognition using features extracted from pre-trained neural networks [102]

2.5.1. Design

Deep convolutional neural networks are extensively used in computer vision for supervised image classification and segmentation problems. However, the need for supervision implies the availability of huge datasets with millions of labelled training images. This is not feasible for many applications. As a consequence, the application of deep neural networks in an unsupervised manner that avoids the use of huge annotated datasets is a very active research area.

This work has evaluated the performance of an unsupervised scene recognition technique proposed in [103]. This technique applies classical clustering to the features extracted by some internal layer of a deep convolutional neural network, such as AlexNet and VGG16, after feeding the network with a set of training images corresponding to the frames of a video sequence recorded while traversing a given path. The different clusters obtained in an unsupervised manner represent the different “places” in the video sequence. Each place/cluster is represented by the cluster centroid. The different places that have been automatically identified are then clustered by applying k-means. The obtained k classes are assumed to represent different scenes appearing in the video (e.g., indoor vs. outdoor).

2.5.2. Experimental results

The technique described in [103] has been implemented upon the AlexNet and VGG16 network models provided in the PyTorch framework, both pre-trained with ImageNet. The two unsupervised clustering algorithms involved in the process (determination of centroids and clustering of centroids) were implemented in PyTorch with extensive use of GPU primitives.

The technique has been evaluated on a proprietary dataset with 200 training images and 200 test images belonging to two classes: indoor and outdoor. The indoor images correspond to different views of the rooms of a house, whereas the outdoor images were captured during a walk on foot over a city’s neighbourhood. Some examples of those images are shown below:

The features extracted from 8 different layers of the VGG16 and AlexNet pre-trained networks were used in the different experiments, considering both their convolutional and fully-connected layers. The final goal was to classify the test images into the two scenes of interest (indoor or outdoor), having characterized those scenes in an unsupervised manner by applying the clustering techniques described above to the extracted features.

The obtained results show that the best scene classification performance was obtained by clustering the output of the last fully-connected layers of both networks instead of the convolutional layers that were only tested in [103], as shown in the table below for VGG16:

Dataset Interior			Dataset Exterior			VGG16
Precisión	Recall	Medida F	Precisión	Recall	Medida F	
1	0.88	0.936	0.893	1	0.945	Fully Connected (Linear 38)
1	0.78	0.876	0.812	1	0.896	Fully Connected (Linear 35)
1	0.54	0.7012	0.685	1	0.813	Fully Connected (Linear 32)
1	0.62	0.765	0.725	1	0.841	conv. (Conv 5_3)
1	0.28	0.437	0.581	1	0.187	Conv. (Conv 4_3)
0.623	0.76	0.68	0.692	0.54	0.609	conv. (Conv 3_3)
1	0.56	0.717	0.694	1	0.8213	conv. (Conv 2_2)
0	0	0	0.5	1	0.167	conv. (Conv 1_2)

The aforementioned classification results for the two considered scenes imply that the output of fully-connected layers can be useful as high-level features that characterize the visual content of images in a far more compact way than features directly extracted from the convolutional layers. The results for VGG16 (max f-measure = 0.94) were slightly superior to those of AlexNet (max f-measure = 0.89). This is consistent with the higher complexity of VGG16.

3. Semantic Segmentation

3.1. Semantic Driven Multi-Camera Pedestrian Detection Approach [3]

3.1.1. Design

Nowadays, pedestrian detection is one of the pivotal fields in computer vision, especially when performed over video surveillance scenarios. People detection methods are highly sensitive to occlusions among pedestrians, which dramatically degrades performance in crowded scenarios. The cutback in camera prices has allowed generalizing multi-camera set-ups, which can better confront occlusions by using different points of view to disambiguate detections. In this paper we present an approach to improve the performance of these multi-camera systems and to make them independent of the considered scenario, via an automatic understanding of the scene content. This semantic information, obtained from a semantic segmentation, is used 1) to automatically generate a common Area of Interest for all cameras, instead of the usual manual definition of this area; and 2) to improve the 2D detections of each camera via an optimization technique which maximizes coherence of every detection both in all 2D views and in the 3D world, obtaining best-fitted bounding boxes and a consensus height for every pedestrian. Experimental results on five publicly available datasets show that the proposed approach, which does not require any training stage, outperforms state-of-the-art multi-camera pedestrian detectors nonspecifically trained for these datasets, which demonstrates the expected semantic-based robustness to different scenarios.

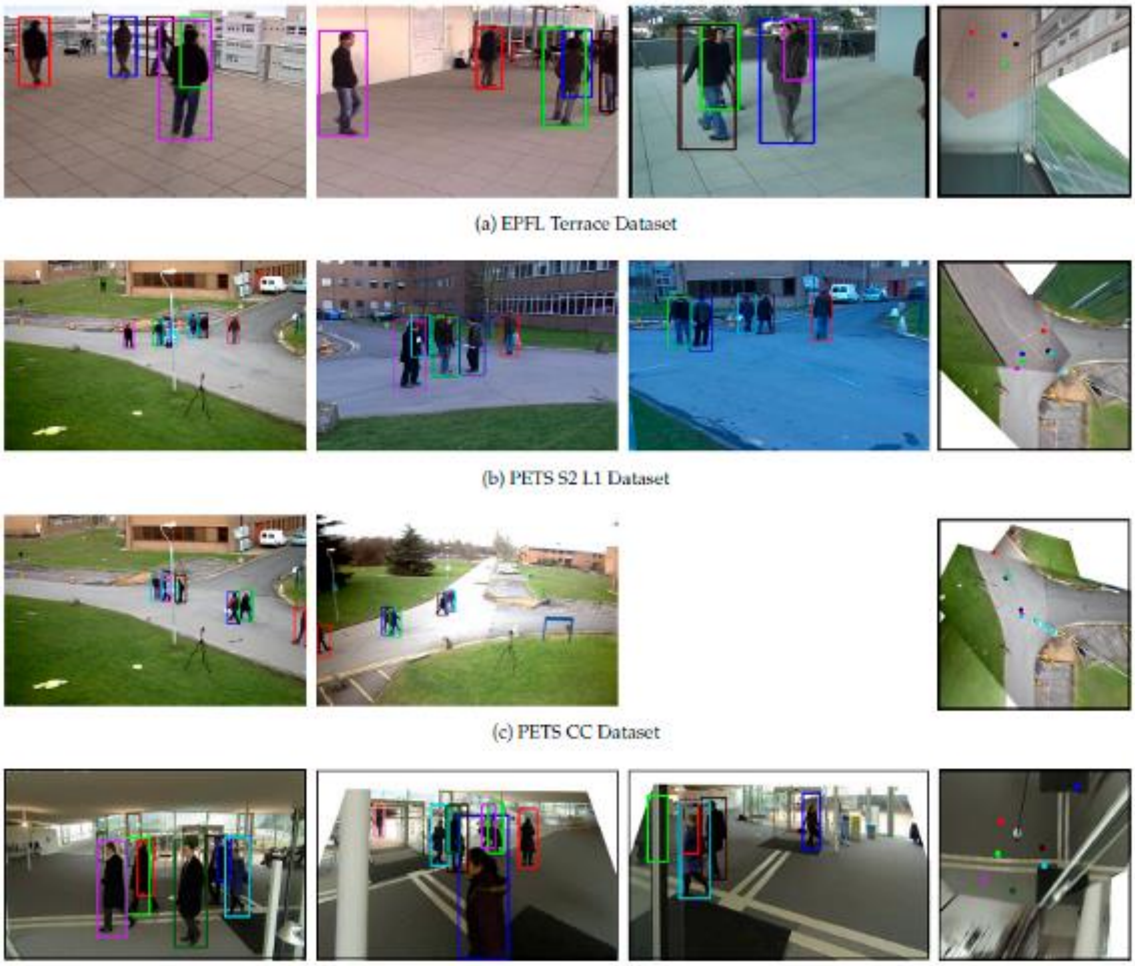
3.1.2. Experimental results

The proposed solution is validated by an extensive comparison with the state-of-the art using five publicly available datasets described in [2]. The following Tables illustrate this comparison. A brief discussion is included for each dataset. See full details in [3].

Results from Table 7 shows that the proposed method outperforms both used baselines (Faster-RCNN [8] and YOLOv3 [7]) when both stages (Pedestrian Semantic Filtering and Semantic-driven Back-projection) of the proposed method are used. Faster-RCNN, in terms of N-MODA is outperformed by an 8.45%, a 4.70%, a 3.52% and a 20.68% for EPFL Terrace, PETS 2009 S2L1, PETS 2009 CC and EPFL RLC respectively. On the other hand, YOLO is outperformed by a 11.84%, a 1.14%, and a 15.25% for EPFL Terrace, PETS 2009 S2L1 and EPFL RLC.

			Dataset															
			EPFL Terrace				PETS 2009 S2 L1				PETS 2009 CC				EPFL RLC			
	Filt	Fus & BP	AUC	F-S	NA	NP	AUC	F-S	NA	NP	AUC	F-S	NA	NP	AUC	F-S	NA	NP
Faster-RCNN			0.82	0.84	0.71	0.74	0.90	0.91	0.85	0.76	0.90	0.91	0.85	0.76	0.77	0.78	0.58	0.69
	✓		0.84	0.85	0.73	0.74	0.90	0.91	0.85	0.76	0.90	0.91	0.85	0.76	0.80	0.82	0.68	0.70
	✓	✓	**0.87**	**0.90**	**0.83**	**0.77**	**0.92**	**0.93**	**0.89**	**0.79**	**0.94**	**0.94**	**0.88**	**0.79**	**0.81**	**0.82**	**0.70**	**0.70**
YOLOv3			0.83	0.87	0.76	0.73	**0.96**	**0.96**	**0.92**	**0.67**	0.92	0.92	0.87	0.79	0.80	0.78	0.59	**0.72**
	✓		0.84	0.87	0.77	0.73	0.96	0.96	0.92	0.67	0.92	0.92	0.87	0.79	0.85	0.83	0.66	0.72
	✓	✓	**0.86**	**0.89**	**0.85**	**0.76**	0.93	0.92	0.89	0.67	**0.94**	**0.94**	**0.88**	**0.79**	**0.85**	**0.83**	**0.68**	0.72

Table 7. Stage-wise performance of the proposed method when Faster-RCNNN [9] and YOLOv3 [8] are used as baselines. Indicators are Area Under the Curve (AUC), F-Score (F-S), N-MODA (N-A) and N-MODP (N-P). *Filt* stands for "Pedestrian Semantic Filtering" stage and *Fus & BP* stands for "Fusion of Multi-Camera Detections (*Fus*) and Semantic-driven Back-projection (*BP*)" stages.



(a) EPFL Terrace Dataset

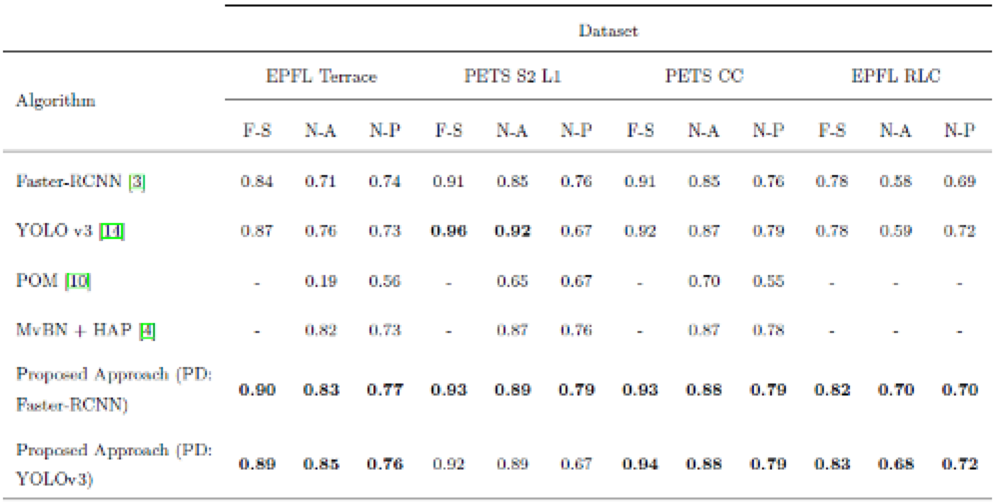
(b) PETS S2 L1 Dataset

(c) PETS CC Dataset

(d) EPFL RLC Dataset

Figure 4. Proposed method qualitative results on selected frames of the EPFL Terrace, PETS S2 L1, PETS CC and EPFL RLC datasets.

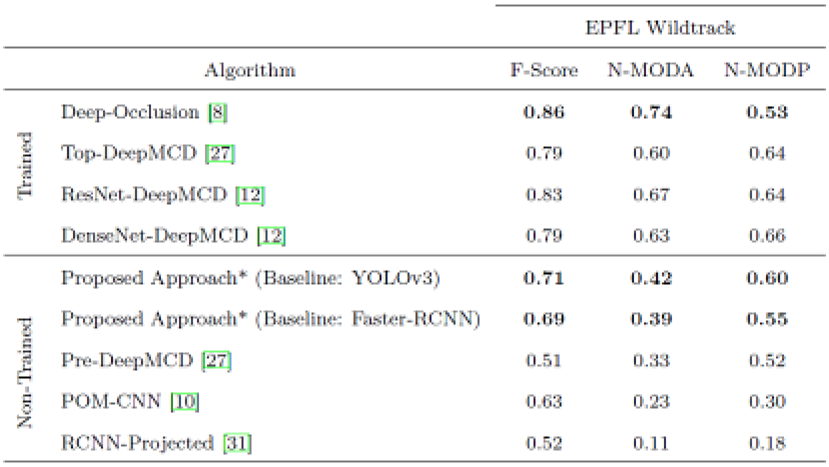
Qualitative results from Figure 4 represent bounding-boxes obtained by the proposed algorithm on first to third columns. Most-right column represents detections on the ground plane. (Faster-RCNN baseline is used for this qualitative example)



Algorithm	Dataset											
	EPFL Terrace			PETS S2 L1			PETS CC			EPFL RLC		
	F-S	N-A	N-P	F-S	N-A	N-P	F-S	N-A	N-P	F-S	N-A	N-P
Faster-RCNN [3]	0.84	0.71	0.74	0.91	0.85	0.76	0.91	0.85	0.76	0.78	0.58	0.69
YOLO v3 [14]	0.87	0.76	0.73	**0.96**	**0.92**	0.67	0.92	0.87	0.79	0.78	0.59	0.72
POM [10]	-	0.19	0.56	-	0.65	0.67	-	0.70	0.55	-	-	-
MvBN + HAP [4]	-	0.82	0.73	-	0.87	0.76	-	0.87	0.78	-	-	-
Proposed Approach (PD: Faster-RCNN)	**0.90**	**0.83**	**0.77**	**0.93**	**0.89**	**0.79**	**0.93**	**0.88**	**0.79**	**0.82**	**0.70**	**0.70**
Proposed Approach (PD: YOLOv3)	**0.89**	**0.85**	**0.76**	0.92	0.89	0.67	**0.94**	**0.88**	**0.79**	**0.83**	**0.68**	**0.72**

Table 8. Comparison with respect to both baselines (Faster-RCNN [3] and YOLOv3 [14]), and multi-camera state-of-the-art methods non based on deep-learning (POM [10] and MvBN + HAP [4]).

Results from Table 8 compare the proposed approach, using Faster-RCNN and YOLOv3 as baselines, with respect to multi-camera pedestrian algorithms. It can be observed that the proposed method yields a higher recall, i.e. increases the number of correct detections by coping with occlusions and pedestrian detector errors, while keeping similar precision, i.e. without increasing the number of false positives. With respect to POM [10] and MvBN + HAP [4], the proposed method also obtains better results in terms of N-MODA which, precisely, measures detection accuracy along the whole sequences.



	Algorithm	EPFL Wildtrack		
		F-Score	N-MODA	N-MODP
Trained	Deep-Occlusion [8]	**0.86**	**0.74**	**0.53**
	Top-DeepMCD [27]	0.79	0.60	0.64
	ResNet-DeepMCD [12]	0.83	0.67	0.64
	DenseNet-DeepMCD [12]	0.79	0.63	0.66
Non-Trained	Proposed Approach* (Baseline: YOLOv3)	**0.71**	**0.42**	**0.60**
	Proposed Approach* (Baseline: Faster-RCNN)	**0.69**	**0.39**	**0.55**
	Pre-DeepMCD [27]	0.51	0.33	0.52
	POM-CNN [10]	0.63	0.23	0.30
	RCNN-Projected [31]	0.52	0.11	0.18

Table 9. Wildtrack Dataset Comparison Results. All the stated methods (except both baselines) are multi-camera deep-learning based algorithms.

Table 9 summarizes state-of-the-art results on Wildtrack Dataset [10]. "Trained" denotes that the algorithm has been explicitly trained on Wildtrack dataset, while "Non-Trained" denotes that the algorithm has not been trained on it. The proposed method, either with Faster-RCNN or YOLOv3 baseline, is also able to outperform all deep-learning approaches that have not been specifically adapted to the Wildtrack dataset. Our method improves 18.18% respect to Pre-DeepMCD [11]—the second ranked—, which is an end-to-end deep learning architecture trained on the PETS dataset.

3.2. Measuring the increase of diversity provided by the Unified Semantic Segmentation Dataset

We have designed a python framework for the training of a semantic segmentation algorithm that jointly considers five of the principal semantic segmentation benchmarks publicly available. The idea is to leverage on different appearances of the defined semantic classes to enhance the generality and scalability of semantic segmentation. To this aim, we have collected and aligned the semantic classes of five semantic segmentation dataset into a Unified Semantic Segmentation Dataset (see [2]).

We want to quantify the increase in diversity obtained in the shared classes by merging the datasets. Diversity refers to the variety that exists within a dataset, species, cultures, etc. We refer to the differences that exist within the same semantic class, that are directly related to the richness of the semantic class and the dominance of one dataset over another. This analysis is performed on those semantic classes that appear in more than one dataset of which the unified dataset is composed. The analysis is performed on the 57 semantic classes that fulfill this requirement.

In order to quantify the added diversity, we leverage on an existing framework for analysing and comparing distributions [73]. Our hypothesis is that if we compare the distribution in terms of a given set of features for samples of a given semantic class of the Unified Dataset with that obtained by using only the samples of that class for one of the individual datasets that are merged, these distributions will be more different if the diversity is increased.

3.2.1. Feature extraction

For each image, we use the binary mask image resulting from selecting a given semantic class to isolate the RGB information for that specific class. Specifically, both images are introduced in a pretrained CNN and the activations at a particular layer are extracted. In this case, activations at the eighth fully connected layer of an Alexnet network trained on ImageNet were extracted, yielding a 1x1000 feature vector for each combination of image and semantic class (see Figure 5).

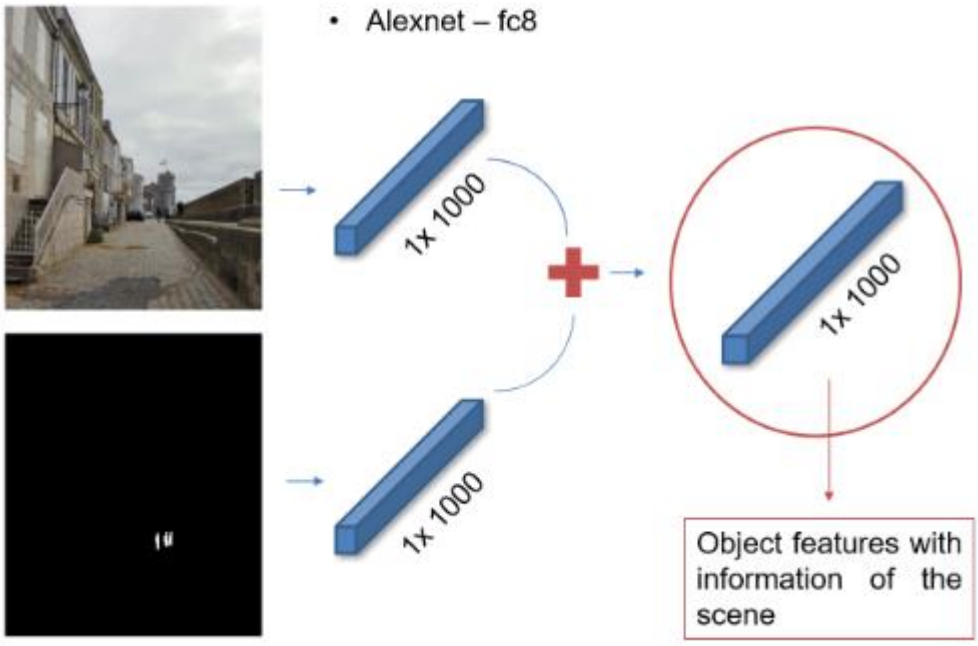


Figure 5. Feature extraction process

3.2.1. Comparison of feature distributions.

For a given semantic class, we randomly select 4000 samples (or the maximum available number of samples, if it is smaller than 4000) of that class from the Unified Dataset (two times) and one of the individual datasets that also contains that class. Therefore, three sets of samples are created, one for the individual dataset and two for the unified dataset. For these three sets of samples, we perform the feature extraction process described in the previous section.

With the three sets of features (U1, U2 and V) we perform two Maximum Mean Discrepancy Tests [73]. First U1 and U2 are compared to ensure that the sampling is meaningful, which is tested by assessing that both sets shape similar distributions; hence resulting in an acceptance test. Then we perform a Maximum Mean Discrepancy Test on U1 and V, if the Unified dataset enlarges the diversity, this may result in a rejected test, as the distributions may be different.

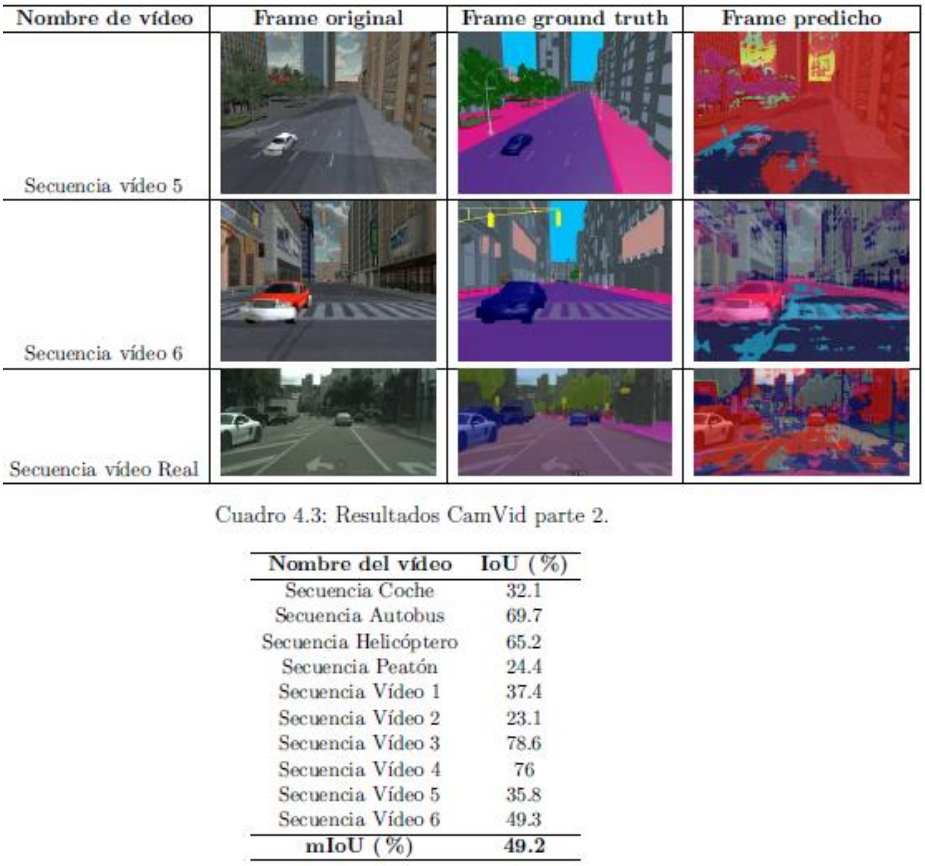
We repeat this process several times to reduce the effect of the random selection. In our preliminary results, the test is rejected 100% of the times for 55 out of the 57 evaluated classes.

3.3. Early experiments for semantic segmentation using synthetic data

This section describes exploratory work focusing on the application of deep learning algorithms to semantic segmentation. To be more precise, the goal is to test different semantic segmentation algorithms with simulated videos instead of real ones.

We have tested two popular approaches: ENET [74] and DeepLabV1[75]. As generator of synthetic data, we used the MSS simulator [76] to obtain several test sequences. Moreover, we have compared different models from the CityScapes dataset [77], Cambridge-Driving dataset [78] and Mapillary dataset [79].

The following figure shows some example results for the experiments performed. Full details of the experiments can be found at the Undegraduate Thesis “Análisis automático de video simulado con sistemas multicámara basados en UNITY”.



Nombre de vídeo	Frame original	Frame ground truth	Frame predicho
Secuencia vídeo 5			
Secuencia vídeo 6			
Secuencia vídeo Real			

Cuadro 4.3: Resultados CamVid parte 2.

Nombre del vídeo	IoU (%)
Secuencia Coche	32.1
Secuencia Autobus	69.7
Secuencia Helicóptero	65.2
Secuencia Peatón	24.4
Secuencia Vídeo 1	37.4
Secuencia Vídeo 2	23.1
Secuencia Vídeo 3	78.6
Secuencia Vídeo 4	76
Secuencia Vídeo 5	35.8
Secuencia Vídeo 6	49.3
mIoU (%)	**49.2**

Figure 6. Semantic segmentation results for ENET using synthetic data

4. Multi-view matching

4.1. People/Car re-identification approach

4.1.1. Description of the algorithm

The proposed re-identification system [15] is based on the combination of adapted deep learning feature embedding representations and a distance metric learning process.

This section includes the summary of the techniques used to develop the proposed multi-camera person/vehicle re-identification approach. In **Figure 7** we have the flow diagram of the approach, first we obtain the features embedding representation using the query, train and test sets. Then, we learn the metric in order to get the projection matrix with the features map. The objective of using metric learning is to learn a feature space where features metrics that belongs to the same object are closer than those of different ones. Finally, we obtain the distances between each query and all the test set.

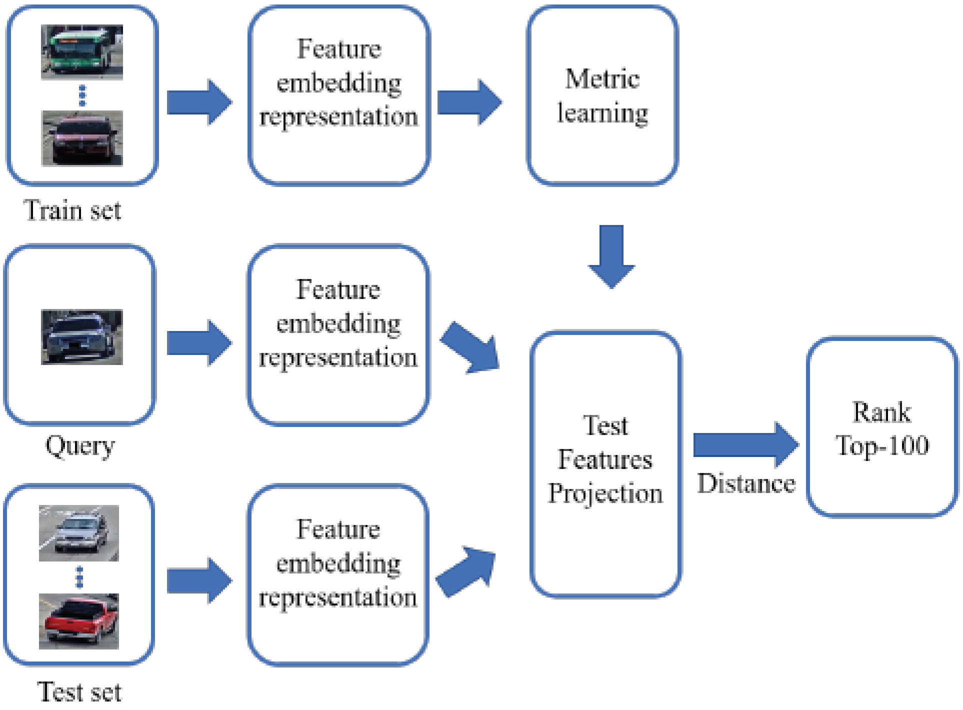

Feature embedding representation
Metric learning
Train set
Feature embedding representation
Query
Test Features Projection
Distance
Rank Top-100
Feature embedding representation
Test set

Figure 7. Flow diagram of the vehicle ReID system approach.

4.1.2. Feature representation

In order to extract the feature representations, we use the networks AlexNet [34], ResNet-18 [35], ResNet-50 [35], ResNet-101 [35], Densenet-201 [36] and Inception-ResNet-v2 [37]. We choose these networks because of their relevance in scene and object classification.

Feature extraction module models the appearance of each detected box via deep learning features by considering the different networks architectures, all of them pre-trained on the ImageNet database [18]. Since ImageNet covers 1000 classes and we need to adapt the model to our target, i.e. vehicles, we train some layers of the network while leaving others frozen. We have based on [38] to decide the frozen parts of the networks. We freeze before the CNN block3 except for AlexNet that we freeze before the pool1 layer. All the remaining parts of the networks that are not frozen adapt their weights when we retrain on the vehicle images.

The input images of the CNNs are resize to 227x227. The parameters used for the transfer learning of the non-frozen layers are a learning rate of 3e-4 and a batch size of 10. We have trained for 6 epochs and use Stochastic Gradient Descent with Momentum optimizer [39].

4.1.3. Metric learning

Instead of using the feature embedding representation and the Euclidean distance to rank the test candidates, we improve the performance of the system introducing a supervision decision using the training data. In particular, the metric learning allows learning a feature space where the feature vectors of the same object ID are closer than the features from different objects. After the evaluation of the three most common metrics from the literature (XQDA [40], NFST [41] and KISSME[42]), we had chosen for the final evaluation the one with the best performance, the XQDA.

4.2. Improvement proposals for the 2019 AI City Challenge

All the improvements included are explained in detail in this section in order to obtain better results than those obtained with the baseline method in the 2019 AI City Challenge [47].

4.2.1. Feature combination at distance level

To increase the performance of our system, we develop a decision combination at distance level. As we can see in **Figure 8**, we first extract the feature representations and learn the metric learning space. Then we compute the distances between the input query and all the images in the gallery. At this point, the distances are normalized between 0 and 1. The final re-identification decision is based in the averaged distance.

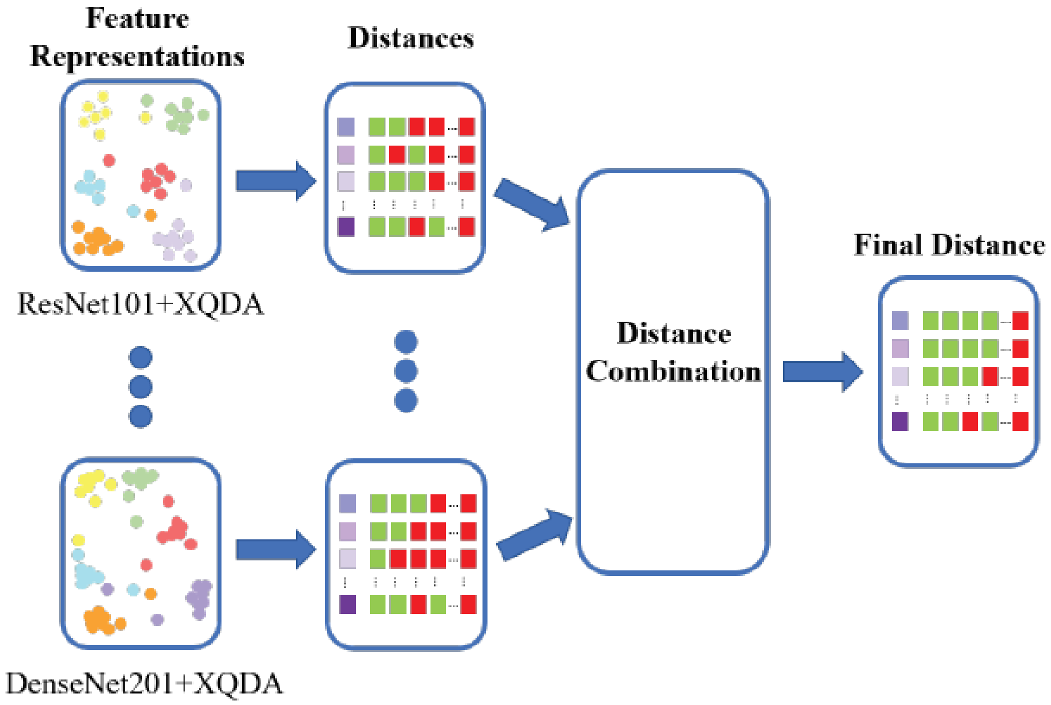


Figure 8. Feature combination at distance level.

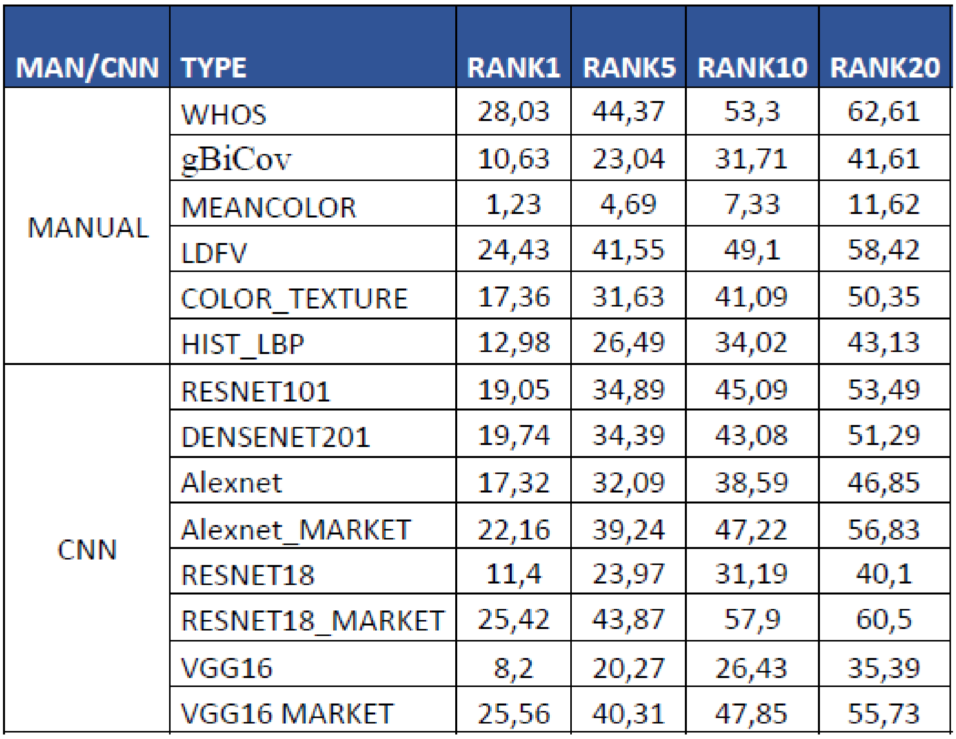
4.2.2. Vehicle trajectory information

Each test track for the CityFlow-ReID dataset [47] contains multiple images of the same vehicle captured by one camera. According to the ranked distance between the query and the test gallery, we can assume that if there are some images of the same test track with small distances, i.e., high confidence of being the same vehicle, the rest of the test track should be also included in the ReID decision.

Therefore, we sort the test tracks that appear in each query (top-100 matches) according to their first occurrence in the top-100 rank. We include progressively in ascending distance order, all the images of the sorted test tracks until we complete the output list of 100 matches.

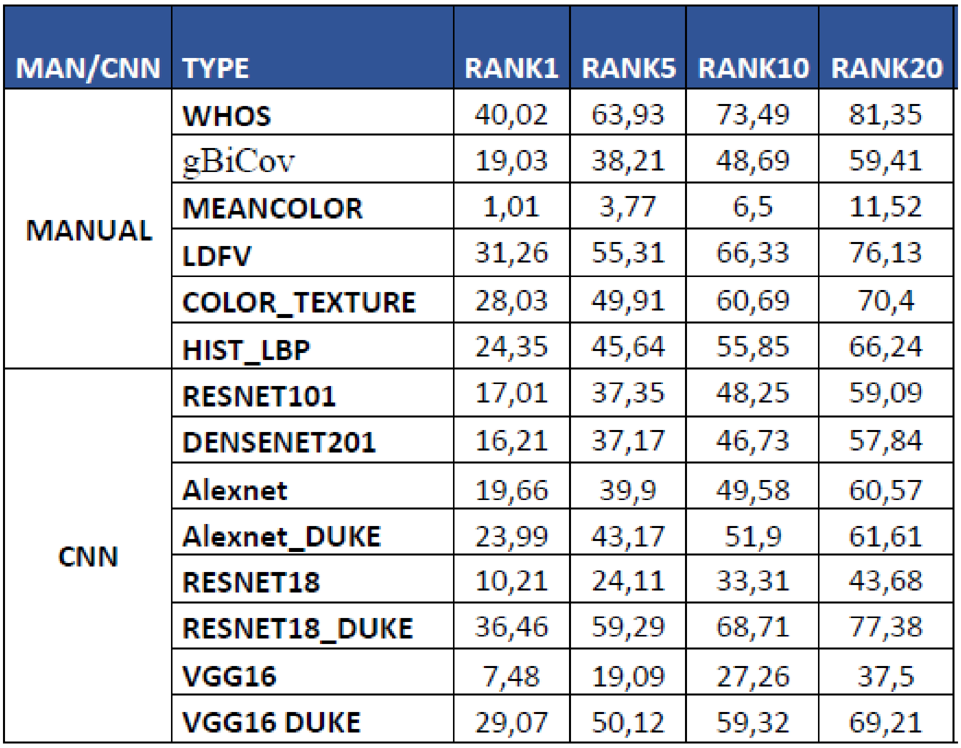
4.2.3. People re-identification results

The basic or the preliminary results were described in the deliverable “D2v1 Feasibility studies algorithms and findings”. This section describes the obtained people re-identification results [17]. We compare the results using hand-crafted (manual) features and Deep-learning-based features (CNN). **Table 10** shows the people re-identification results obtained in dataset DuleMTMC4ReID [45] using Market1501 [44] as training dataset. **Table 11** shows the people re-identification results obtained in dataset Market1501 [44] using DuleMTMC4ReID [45] as training dataset. **Table 12** shows the people re-identification results obtained in dataset ViPER [43] using both DuleMTMC4ReID [45] and Market1501 [44] as training dataset. In general, the results show clearly that the re-training process improve significantly the CNN based features. However, the traditional features or hand-crafted have been tuned during many year in the state of the art of people re-identification and still gets better results.



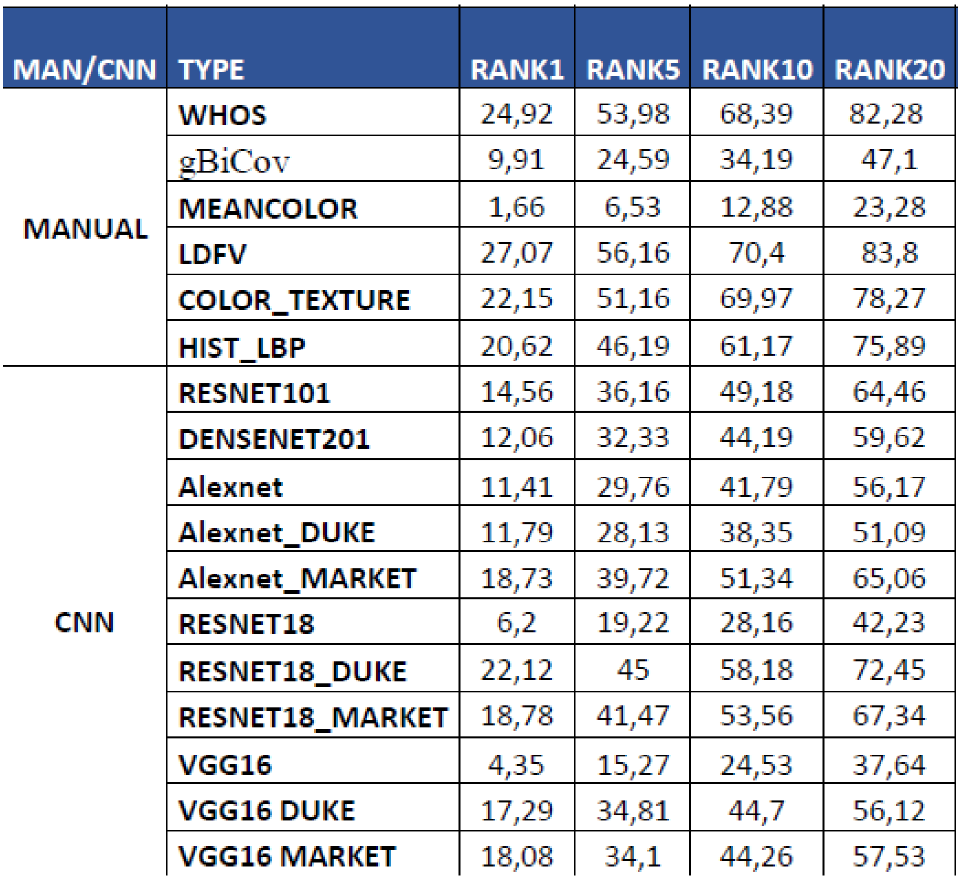
MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	WHOS	28,03	44,37	53,3	62,61
	gBiCov	10,63	23,04	31,71	41,61
	MEANCOLOR	1,23	4,69	7,33	11,62
	LDFV	24,43	41,55	49,1	58,42
	COLOR_TEXTURE	17,36	31,63	41,09	50,35
	HIST_LBP	12,98	26,49	34,02	43,13
CNN	RESNET101	19,05	34,89	45,09	53,49
	DENSENET201	19,74	34,39	43,08	51,29
	Alexnet	17,32	32,09	38,59	46,85
	Alexnet_MARKET	22,16	39,24	47,22	56,83
	RESNET18	11,4	23,97	31,19	40,1
	RESNET18_MARKET	25,42	43,87	57,9	60,5
	VGG16	8,2	20,27	26,43	35,39
	VGG16 MARKET	25,56	40,31	47,85	55,73

Table 10 People re-identification results obtained in dataset DuleMTMC4ReID [45].



MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	WHOS	40,02	63,93	73,49	81,35
	gBiCov	19,03	38,21	48,69	59,41
	MEANCOLOR	1,01	3,77	6,5	11,52
	LDFV	31,26	55,31	66,33	76,13
	COLOR_TEXTURE	28,03	49,91	60,69	70,4
	HIST_LBP	24,35	45,64	55,85	66,24
CNN	RESNET101	17,01	37,35	48,25	59,09
	DENSENET201	16,21	37,17	46,73	57,84
	Alexnet	19,66	39,9	49,58	60,57
	Alexnet_DUKE	23,99	43,17	51,9	61,61
	RESNET18	10,21	24,11	33,31	43,68
	RESNET18_DUKE	36,46	59,29	68,71	77,38
	VGG16	7,48	19,09	27,26	37,5
	VGG16 DUKE	29,07	50,12	59,32	69,21

Table 11 People re-identification results obtained in dataset Market1501 [44].

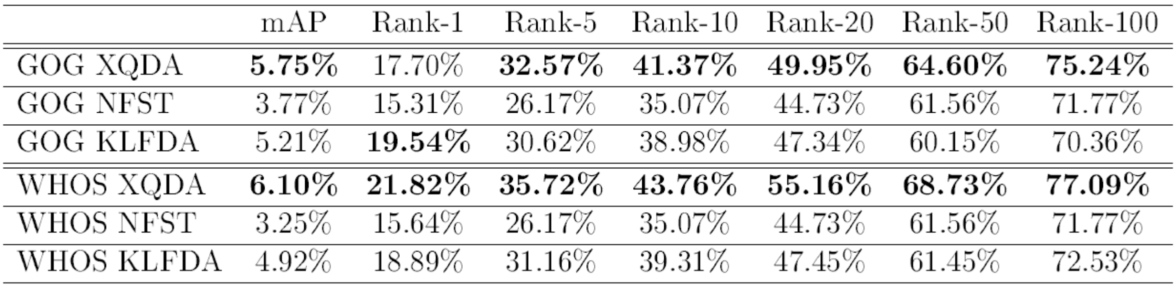


MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	WHOS	24,92	53,98	68,39	82,28
	gBiCov	9,91	24,59	34,19	47,1
	MEANCOLOR	1,66	6,53	12,88	23,28
	LDFV	27,07	56,16	70,4	83,8
	COLOR_TEXTURE	22,15	51,16	69,97	78,27
	HIST_LBP	20,62	46,19	61,17	75,89
CNN	RESNET101	14,56	36,16	49,18	64,46
	DENSENET201	12,06	32,33	44,19	59,62
	Alexnet	11,41	29,76	41,79	56,17
	Alexnet_DUKE	11,79	28,13	38,35	51,09
	Alexnet_MARKET	18,73	39,72	51,34	65,06
	RESNET18	6,2	19,22	28,16	42,23
	RESNET18_DUKE	22,12	45	58,18	72,45
	RESNET18_MARKET	18,78	41,47	53,56	67,34
	VGG16	4,35	15,27	24,53	37,64
	VGG16 DUKE	17,29	34,81	44,7	56,12
	VGG16 MARKET	18,08	34,1	44,26	57,53

Table 12 People re-identification results obtained in dataset ViPER [43].

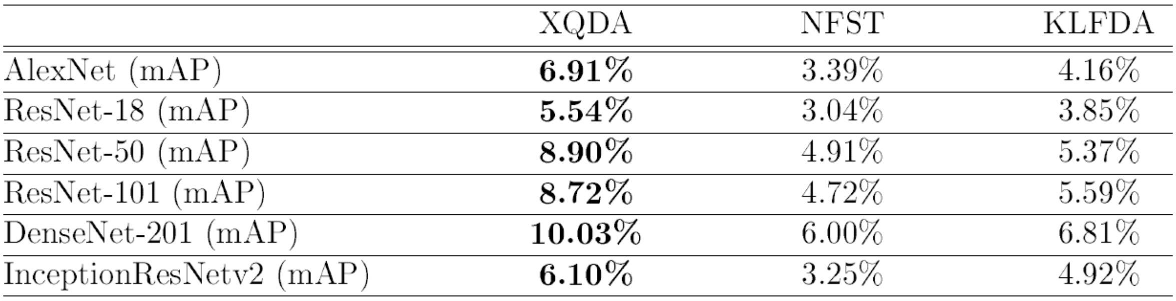
4.2.4. Car re-identification results

The basic or the preliminary results were described in the deliverable "D2v1 Feasibility studies algorithms and findings". This section describes the obtained car re-identification results [15][16] over the car re-identification dataset CityFlow-ReID [46]. We first compare the results using the three most common metrics from the literature (XQDA [29], NFST [30] and KISSME [31]) using the baseline algorithms in **Table 13** and **Table 14**. The results show clearly a better performance using the metric XQDA.



	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
GOG XQDA	**5.75%**	17.70%	**32.57%**	**41.37%**	**49.95%**	**64.60%**	**75.24%**
GOG NFST	3.77%	15.31%	26.17%	35.07%	44.73%	61.56%	71.77%
GOG KLFDA	5.21%	**19.54%**	30.62%	38.98%	47.34%	60.15%	70.36%
WHOS XQDA	**6.10%**	**21.82%**	**35.72%**	**43.76%**	**55.16%**	**68.73%**	**77.09%**
WHOS NFST	3.25%	15.64%	26.17%	35.07%	44.73%	61.56%	71.77%
WHOS KLFDA	4.92%	18.89%	31.16%	39.31%	47.45%	61.45%	72.53%

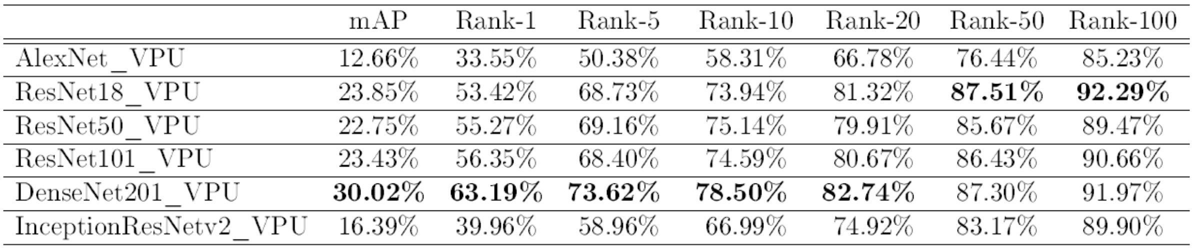
Table 13 GOG and WHOS comparison with XQDA, NFST and KLFDA.



	XQDA	NFST	KLFDA
AlexNet (mAP)	**6.91%**	3.39%	4.16%
ResNet-18 (mAP)	**5.54%**	3.04%	3.85%
ResNet-50 (mAP)	**8.90%**	4.91%	5.37%
ResNet-101 (mAP)	**8.72%**	4.72%	5.59%
DenseNet-201 (mAP)	**10.03%**	6.00%	6.81%
InceptionResNetv2 (mAP)	**6.10%**	3.25%	4.92%

Table 14 Metric Learning comparison with baseline CNNs. In bold is the XQDA result with the best performance for all the networks.

The, we present the obtained results after re-tanning the CNN architectures (XNet_VPU version) in **Table 15**. We realize that using the fine-tuned architectures we obtain more than the double of mAP. For instance, in case of DenseNet-201 (architecture trained in ImageNet) and DenseNet-201_VPU (architecture fine-tuned in CityFlow-ReID-subset) the mAP obtained is 10.03% and 30.02% respectively. Also, the rank list is significantly higher in case of fine-tuned architectures.



	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet_VPU	12.66%	33.55%	50.38%	58.31%	66.78%	76.44%	85.23%
ResNet18_VPU	23.85%	53.42%	68.73%	73.94%	81.32%	**87.51%**	**92.29%**
ResNet50_VPU	22.75%	55.27%	69.16%	75.14%	79.91%	85.67%	89.47%
ResNet101_VPU	23.43%	56.35%	68.40%	74.59%	80.67%	86.43%	90.66%
DenseNet201_VPU	**30.02%**	**63.19%**	**73.62%**	**78.50%**	**82.74%**	87.30%	91.97%
InceptionResNetv2_VPU	16.39%	39.96%	58.96%	66.99%	74.92%	83.17%	89.90%

Table 15 Results of the fine-tuned deep learning feature methods obtained in the CityFlow-ReID-subset, all of them with the metric learning XQDA. In bold are the results with the best performance, in particular for DenseNet201_VPU and ResNet18_VPU.

4.2.5. 2019 AI City Challenge re-identification results

The results of the AI City Challenge have been published on May of 2019. There were three tracks with different issues to solve. Fist track was City-scale multi-camera vehicle tracking, second one was the City-scale multi-camera vehicle re-identification (our participation track) and the last one was Traffic anomaly detection. The number of participants to each track were 22, 84 and 23 respectively, being our track the one with more participants. We published our work in [16].

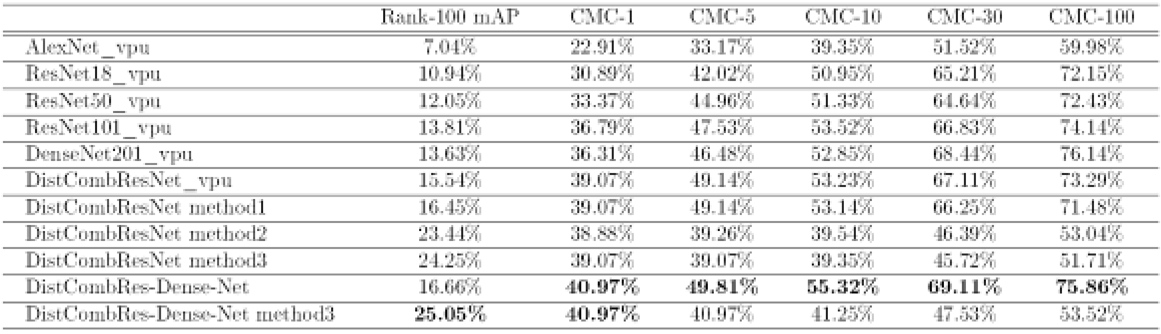
The environment given by 2019 NVIDIA AI City Challenge has allowed to submit up 5 results per day, with a total of 20 submissions. The results that have returned the server until the competition deadline were computed on a 50% subset of the test data. The online server also has provided a leader board with the top 3 results of all the competition and the own best result (in case not to be on the top-3). Once the deadline has been reached, the server shows all the submissions evaluated with all the test set and the entire leader board with all the participants' best result.

In **Table 16** we can see the results given at the end of the challenge of the different methods that we have developed. First of all, we have the features embedding representation with XQDA as metric learning and the CNNs AlexNet, ResNet18, ResNet50, ResNet101 and DenseNet201, given ResNet101 and DensNet201 the best results in mAP and in Rank-1, and Rank-100 for the case of DenseNet201. Then, we develop the distance combinations with the distance of ResNet101, ResNet50 and ResNet18 (DisCombResNet) and ResNet101, DenseNet201 and ResNet50 (DistCombRes-Dense-Net), obtaining similar ranks values and a higher mAP than with each network separately.

When we include the information of the tracks files provided in the CityFlow-ReID [47] explained in section 3.6.2, we improve the mAP with the inconvenient that we loss precision. DistCombResNet method1 ,DistCombResNet method2 ,DistCombResNet method3 are the first, second and third method respectively. The best result is given by the third method of the distance combination of ResNet101, DenseNet201 and ResNet50 (DistCombRes-Dense-Net method3) with a mAP value of 25.05%.

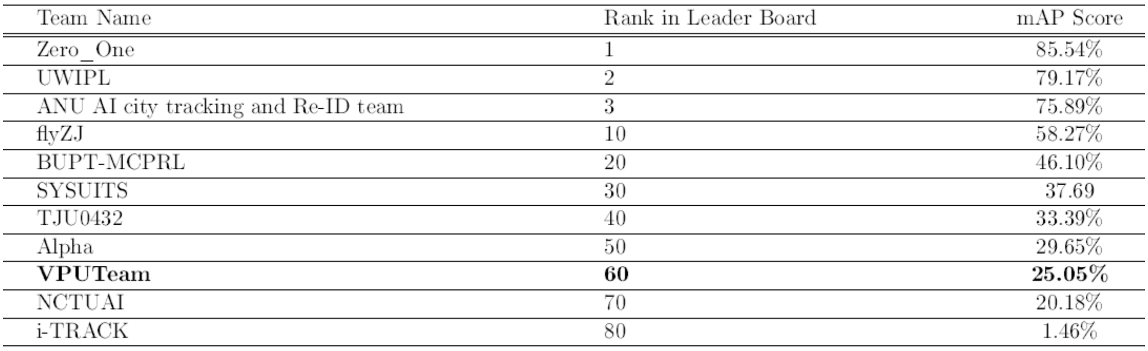
We compare the results obtained with our experimental setup included in **Table 15** with the ones obtained in the AI City server in **Table 16**. For instance, the value of AlexNet_VPU in our evaluation gives a mAP value of 12.66% while in the AI City evaluation is 7.04%. The same thing happens with the results of the other feature embedding representations. In our evaluation the results are around double than for the AI City server. That could be because, our evaluation is done in a reduce subset of the CityFlow-ReID dataset given, and furthermore, the challenge does not provide the entire data in order to make its own evaluation.

The method proposed in this paper has finished the 60 out of the 84 participating teams on the challenge City-Scale Multi-Camera Vehicle Re-Identification. In order to compare our performance in the challenge with the other teams, we show in **Table 17** the participants that are in the multiples of ten positions in the rank. We can see that the team in position 40th (TJU0432) that is in the middle of the ranked results of the challenge has a mAP score equal to 33.39%, which is only 8.34% more than our mAP result (25.05%). Best mAP result achieved in the challenge is equal to 85.54%. The teams with the best performance use as baseline the networks trained using triplet loss or cross entropy loss. They also include in the classification step the information of vehicle models and the vehicle orientation.



	Rank-100 mAP	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
AlexNet_vpu	7.04%	22.91%	33.17%	39.35%	51.52%	59.98%
ResNet18_vpu	10.94%	30.89%	42.02%	50.95%	65.21%	72.15%
ResNet50_vpu	12.05%	33.37%	44.96%	51.33%	64.64%	72.43%
ResNet101_vpu	13.81%	36.79%	47.53%	53.52%	66.83%	74.14%
DenseNet201_vpu	13.63%	36.31%	46.48%	52.85%	68.44%	76.14%
DistCombResNet_vpu	15.54%	39.07%	49.14%	53.23%	67.11%	73.29%
DistCombResNet method1	16.45%	39.07%	49.14%	53.14%	66.25%	71.48%
DistCombResNet method2	23.44%	38.88%	39.26%	39.54%	46.39%	53.04%
DistCombResNet method3	24.25%	39.07%	39.07%	39.35%	45.72%	51.71%
DistCombRes-Dense-Net	16.66%	**40.97%**	**49.81%**	**55.32%**	**69.11%**	**75.86%**
DistCombRes-Dense-Net method3	**25.05%**	**40.97%**	40.97%	41.25%	47.53%	53.52%

Table 16 Results obtained in the online evaluation AI City Challenge [47] server for our different methods, all of them with the metric learning XQDA.



Team Name	Rank in Leader Board	mAP Score
Zero_One	1	85.54%
UWIPL	2	79.17%
ANU AI city tracking and Re-ID team	3	75.89%
flyZJ	10	58.27%
BUPT-MCPRL	20	46.10%
SYSUITS	30	37.69
TJU0432	40	33.39%
Alpha	50	29.65%
VPUTeam	**60**	**25.05%**
NCTUAI	70	20.18%
i-TRACK	80	1.46%

Table 17 Results of the leader board in [47].

4.3. Proposal for the 2020 AI City Challenge

All the improvements included are explained in detail in this section in order to obtain better results than those obtained with the previous proposal in the 2019 AI City Challenge [47].

This section describes the details of the techniques used to develop the proposed multi-camera vehicle ReID approach (see **Figure 9**). On the top of the figure we have the input of the system, on one hand it is image-based in case of feature with the different combination of losses and, on the other hand it is video-based for the keypoint and visibility estimation. The train step adjust the weights of each pre-trained CNN modules to the CityFlow-ReID dataset. Then, the test step infers the gallery and query images in order to obtain all the features. These features are assembled to have a unique feature representation for each image. After that, a query expansion and a temporal pooling for the gallery are applied in order to refine the feature representation and to obtain more accurate results. Once the distances between the gallery and the query images are calculated, the post-processing steps, re-ranking and the inclusion of trajectory information methods proposed in this work, are performed to improve the final ReID results.

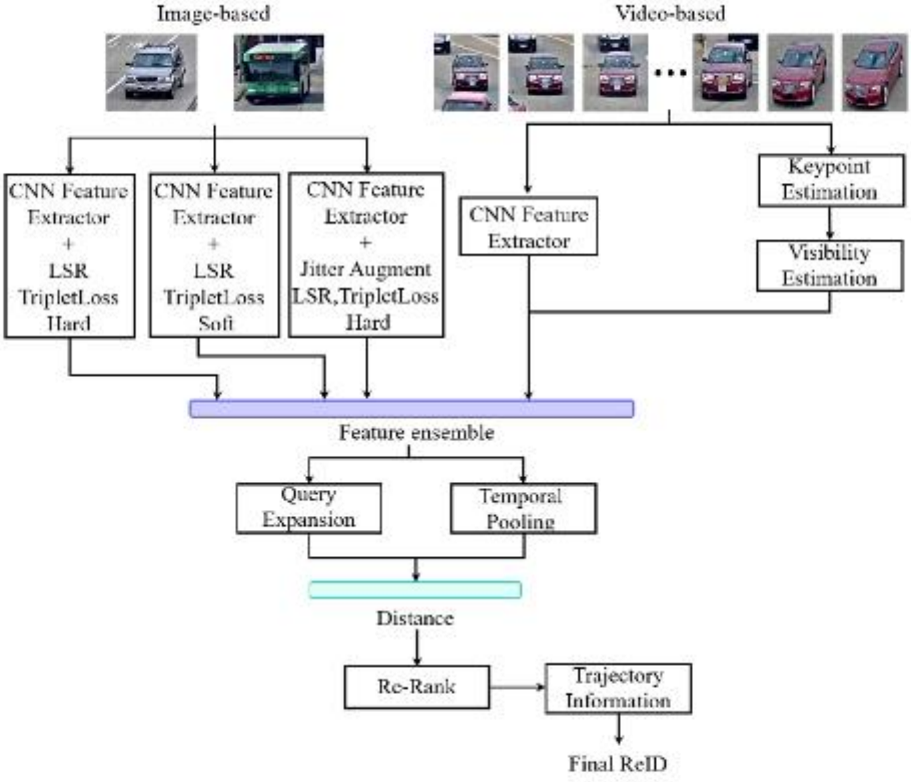


Figure 9. Proposed system overview.

4.3.1. Feature Extraction

Image-based features extractors. This part of the sys-tem uses images as input and the architecture chosen to obtain the feature representation is DenseNet121 [36] pre-trained on ImageNet [66], based on Lv et al. [82]. To train this convolutional neural network, a cross-entropy loss and a triplet loss trained with batch-hard sampling method are used. According to the different variations on loss functions, it could be divided in the feature extractors: The first uses label smooth regularization (LSR) and triplet loss with hard margin, The second network training conditions also use LSR and triplet loss, but in this case it is trained using softmargin [83]. In the last module, the training loss variation combines LSR, triplet loss with hard margin and Jitter Augmen-tation.

Video-based features extractor. The input to this part of the system are a set of images (bounding boxes), consecutive in time and location, of the same vehicle. The features extractor convolutional neural network is ResNet50 [35] pretrained on ImageNet [66] that obtains the features related to appearance of the identity. Following [80] and [81], the orientation of the vehicle is obtained locating the 36 vehicle keypoints that define 18 vehicle orientation surfaces.

The surfaces determine the visible areas of the vehicle, giving the orientation. This structure features are concatenated to the previous appearance features and a triplet loss hard margin and a cross-entropy functions are included in the training.

4.3.2. Feature Ensemble

Once the three features from image-based part and the appearance and structure feature form the video-based are extracted, in this module of the system they are concatenated in order to obtain a more robust representation feature. To perform this combination, the four different features must be normalized by L2 normalization.

4.3.3. Query expansion and Temporal pooling

In order to obtain a more discriminative feature representation, a query expansion [84][85] and a temporal pooling for gallery are applied. The proposed query expansion performs a sum-aggregation and re-normalization of the features that belong to a specific query and the top-k gallery features that are retrieved as the sorted ReID list. The resulting feature will be the new query feature. Then, for the gallery features, it takes into account the trajectory information and performs an average pooling for the T−1 consecutive images. In this work, T is fixed to 6 (as proposed in [81]).

4.3.4. Post-processing: Re-ranking and Trajectory information inclusion

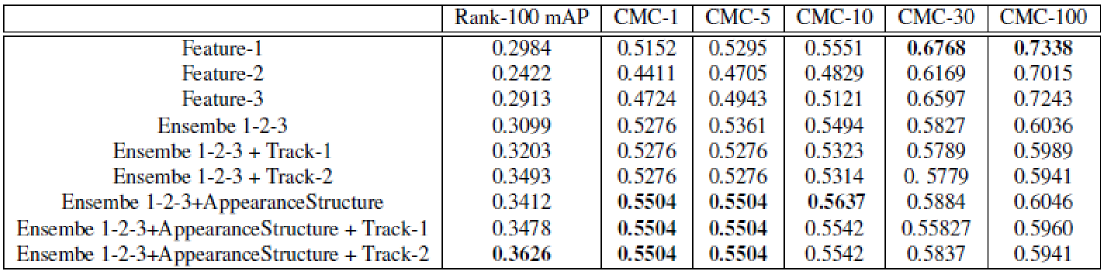
Re-ranking with k-reciprocal encoding. Following [86] we include a post processing step that exploits the hypothesis that if a gallery image is close in the retrieval result of a probe in the k-reciprocal nearest neighbors, its chance of being a true match is higher. For this task, the k-reciprocal nearest neighbors features are encoded into a single feature which will be used for the re-ranking using Jaccard distance.

Trajectory information. Already described in previous section 4.2.2.

4.3.5. 2020 AI City Challenge re-identification results

All the experiments developed to analyze the performance of the proposed method are collected in this section. The two metrics used to evaluate the performance are mean Average Precision (mAP) [87] of the top-100 matches, that is the mean of all the queries' average precision (area un-der the Precision-Recall curve), and the other metric is therank-100 hit rate (additionally, rank-1, rank-5, rank-10, and rank-30 hit rates are shown).

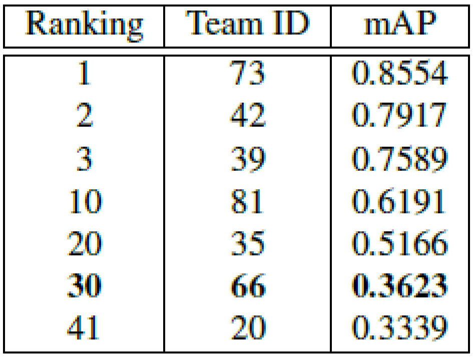
Table 18 shows the different proposed system configurations results obtained on the online evaluation server. Feature-1 is the feature extractor block that we can see in **Figure 9**, which applies Densenet121 network with LSR and triplet loss with hard margin. Feature-2 is Densenet121 with LSR and triplet loss with soft margin and, finally, Feature-3 is the same CNN architecture with jitter augmentation, LSR and triplet loss with hard margin. After assembling these three methods (Ensemble 1-2-3) the result (mAP= 0.3099) overcomes in 3.71% to previous result of the best feature. In the last step, the trajectory information is included using method 1 and method 2 described in3.4. Method 1 improves the previous ensemble result in a 3.24%, whilst method 2 in an 11.27%.



	Rank-100 mAP	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
Feature-1	0.2984	0.5152	0.5295	0.5551	**0.6768**	**0.7338**
Feature-2	0.2422	0.4411	0.4705	0.4829	0.6169	0.7015
Feature-3	0.2913	0.4724	0.4943	0.5121	0.6597	0.7243
Ensembe 1-2-3	0.3099	0.5276	0.5361	0.5494	0.5827	0.6036
Ensembe 1-2-3 + Track-1	0.3203	0.5276	0.5276	0.5323	0.5789	0.5989
Ensembe 1-2-3 + Track-2	0.3493	0.5276	0.5276	0.5314	0. 5779	0.5941
Ensembe 1-2-3+AppearanceStructure	0.3412	**0.5504**	**0.5504**	**0.5637**	0.5884	0.6046
Ensembe 1-2-3+AppearanceStructure + Track-1	0.3478	**0.5504**	**0.5504**	0.5542	0.55827	0.5960
Ensembe 1-2-3+AppearanceStructure + Track-2	**0.3626**	**0.5504**	**0.5504**	0.5542	0.5837	0.5941

Table 18 Table of results obtained in Evaluation server for our different proposals. Bold indicates best performance per metric.

Moreover, the module of appearance and structure feature extraction is included. As we can see in **Table 18**, it supposes an increase in terms of mAP with respect to the feature 1, 2 or 3 due to the introduction of the video-based feature. If we compare the ensemble of the three appearance features with the ensemble with the three features and the appearance and structure video-based one, this last one provides an improvement of 8.96%. As earlier noted, including method 2 of the trajectory information gives an improvement, in this case of 5.9%. **Figure 10** shows the visual result of two specific queries for Feature-1 compare with the assembling of the three features and fourth one (appearance and structure). In case of using only feature-1, it returns more false matches. Then, **Figure 11** shows the ReID result of two different queries. The upper row for each query belongs to the results of ensemble the Features 1-2-3 and the appearance and structure feature, and the lower row corresponds to the same feature ensemble(at all are true positives, but when we move in the rank list, we can see that the trajectory information provides more true positives. In addition, **Table 19** shows the results of the leader board in the AI City Challenge 2020, where the system proposed in this work achieved the30thrank on the list with a (mAP= 0.3626) using the feature ensemble method of the four features and the trajectory method 2.



Ranking	Team ID	mAP
1	73	0.8554
2	42	0.7917
3	39	0.7589
10	81	0.6191
20	35	0.5166
30	**66**	**0.3623**
41	20	0.3339

Table 19 Table of track 2 leader board: City-Scale Multi-Camera Vehicle Re-Identification. Bold indicates this system approach.

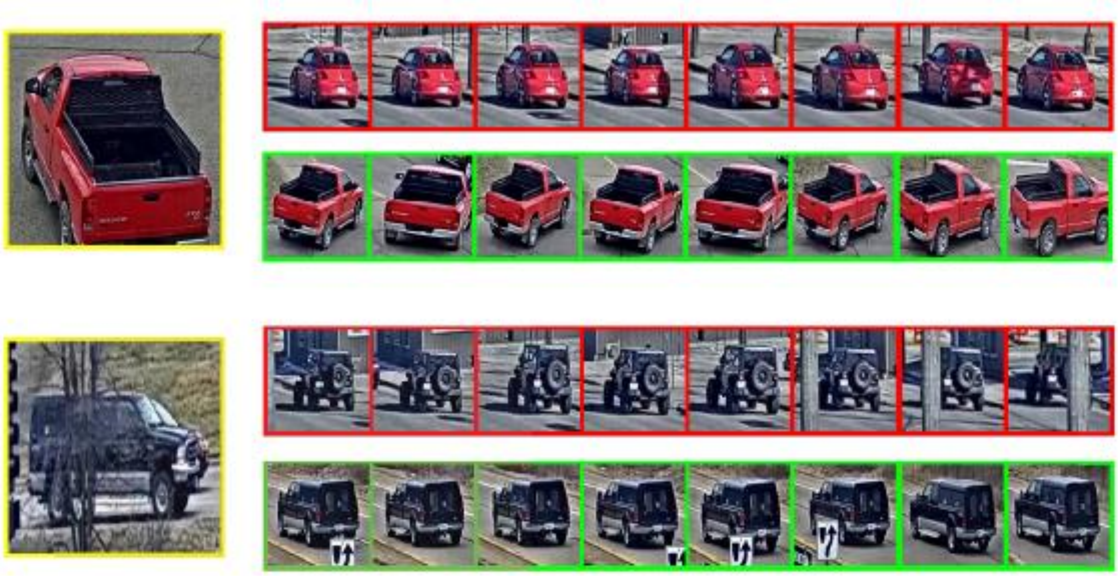


Figure 10. Example 1 of the visual results for the proposed ReID system. It shows two queries (in yellow), the upper rows of each query is the result for only use Feature-1, and lower rows is the result of ensemble the four-feature representation. Green blobs represent true matches and red blobs false matches.

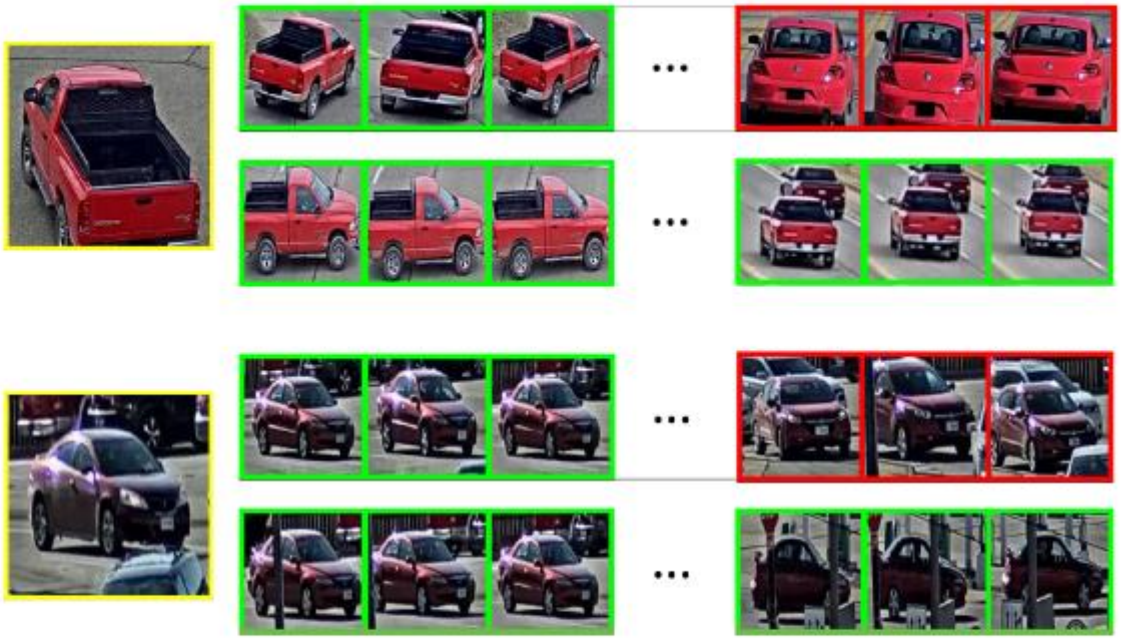
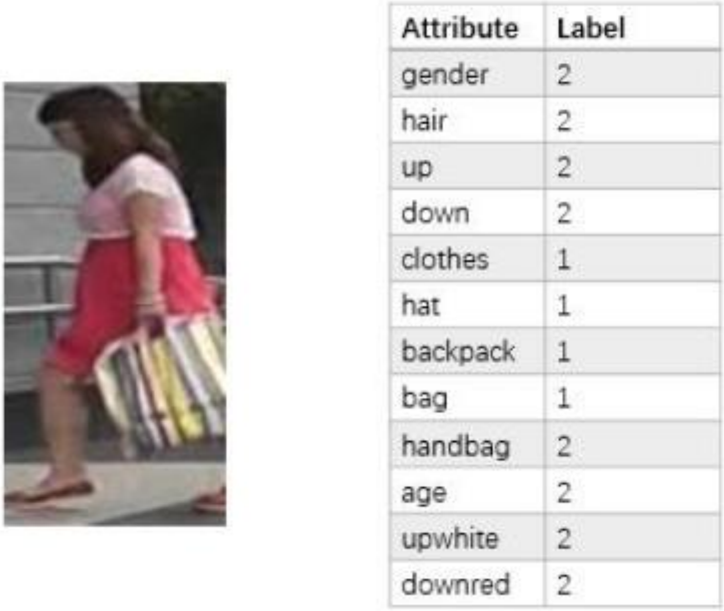


Figure 11. Example 2 of the visual results for the proposed ReID system. It shows two queries (in yellow), the upper rows of each query is the result of ensemble the four feature representation, and lower rows is the result of Ensemble the four features representation and trajectory method 2. Green blobs represent true matches and red blobs false matches.

4.4. Use of attributes for People/Car re-identification

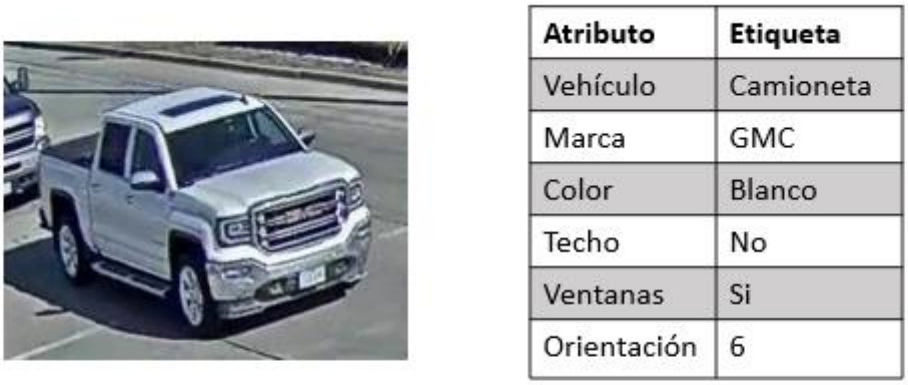
The focus of this project is the study of people and vehicle re-identification systems based on the combination of deep learning characteristics and traditional characteristics that describe the data used.

The main idea is the combination of deep learning architectures for re-identification with the attributes extracted automatically with a pre-trained attributes classification. The proposal will be evaluate for both people and car re-identification. The corresponding dataset will be Market and Aicity, with twelve and six annotated attributes. See **Figure 12** and **Figure 13** for more details.



Attribute	Label
gender	2
hair	2
up	2
down	2
clothes	1
hat	1
backpack	1
bag	1
handbag	2
age	2
upwhite	2
downred	2

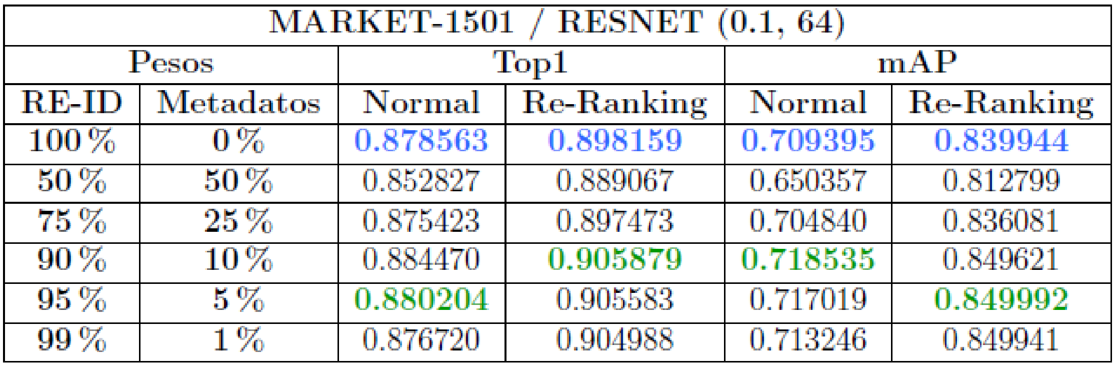
Figure 12. Market example with the twelve attributes.



Atributo	Etiqueta
Vehículo	Camioneta
Marca	GMC
Color	Blanco
Techo	No
Ventanas	Si
Orientación	6

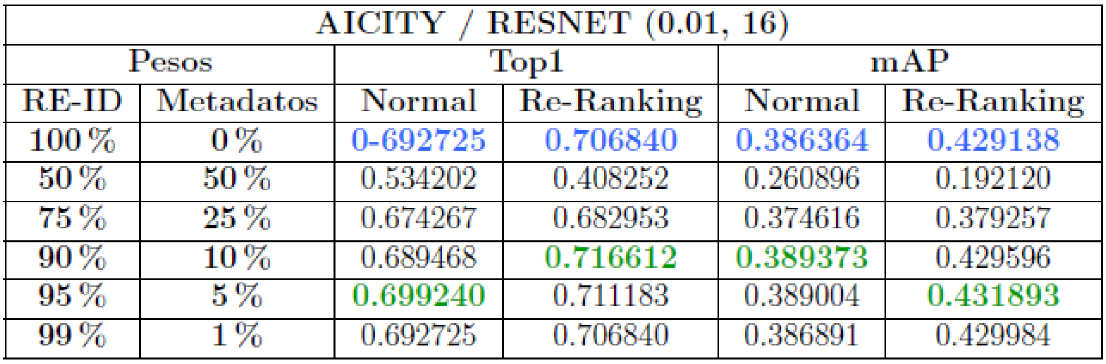
Figure 13. Aicity example with six attributes.

The results combine the feature extracted with the already described re-identification baseline (see section 4.1) and the feature extracted from the trained attribute classifier. This combination has been tested for two deep network architectures ResNet [35] and Densenet [36] and different weighting between the deep learning feature and the attribute classifier. **Table 20** and **Table 21** show example of obtained results for people and car re-identification respectively.



MARKET-1501 / RESNET (0.1, 64)					
Pesos		Top1		mAP	
RE-ID	Metadatos	Normal	Re-Ranking	Normal	Re-Ranking
100 %	0 %	0.878563	0.898159	0.709395	0.839944
50 %	50 %	0.852827	0.889067	0.650357	0.812799
75 %	25 %	0.875423	0.897473	0.704840	0.836081
90 %	10 %	0.884470	0.905879	0.718535	0.849621
95 %	5 %	0.880204	0.905583	0.717019	0.849992
99 %	1 %	0.876720	0.904988	0.713246	0.849941

Table 20. Re-identification results after combining teh original deep learning features with the features extracted from the attribute classifier (“Metadatos”). In blue the original or baseline results, in green the best results.



AICITY / RESNET (0.01, 16)					
Pesos		Top1		mAP	
RE-ID	Metadatos	Normal	Re-Ranking	Normal	Re-Ranking
100 %	0 %	0-692725	0.706840	0.386364	0.429138
50 %	50 %	0.534202	0.408252	0.260896	0.192120
75 %	25 %	0.674267	0.682953	0.374616	0.379257
90 %	10 %	0.689468	0.716612	0.389373	0.429596
95 %	5 %	0.699240	0.711183	0.389004	0.431893
99 %	1 %	0.692725	0.706840	0.386891	0.429984

Table 21. Re-identification results after combining teh original deep learning features with the features extracted from the attribute classifier (“Metadatos”). In blue the original or baseline results, in green the best results.

In general, although the general improvement is relatively small, the results show how the use of attributes always gets an improvement with a small weight of the attribute classifier. In the future, we will explore other strategies for combining both sources of information.

5. Cooperative detection and tracking

5.1. Single-target tracking

5.1.1. Description of the algorithm

We present a detection-based multiple object tracker from Unmanned Aerial Vehicles (UAVs). This work is included in the European Conference on Computer Vision (ECCV) 2018 proceedings[48].

The proposed detection-based tracker models the targets by their visual appearance (via deep features) and their spatial location (via bounding boxes). It is composed of five main modules (see Figure 14), which are described hereunder, and receives as inputs the frame under consideration and the detections for each frame (i.e. bounding box, confidence and object class), provided by an external object detection algorithm. The output for each target is a track describing the sequential information over time.

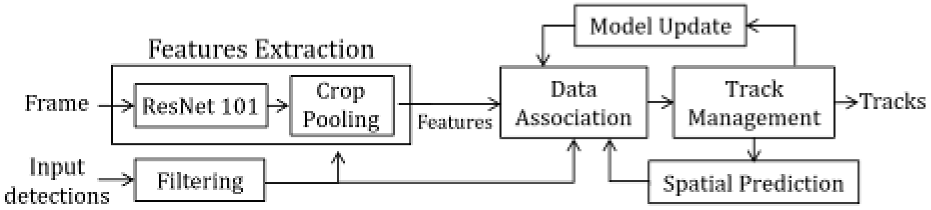


Figure 14. Block diagram of the proposed algorithm

5.1.1.1. Features Extraction

The feature extraction module describes the appearance of bounding boxes. Based on Faster-RCNN [49], we compute features from the input frame with the ResNet-101[50]. deep residual network (pre-trained on the COCO dataset[1]) at layer $conv3\_12$. We use as region proposals the provided detections after confidence-based filtering. For each proposal we get a $512 x 7 x 7$ feature map by crop pooling [51], which becomes a 512 features vector by average pooling.

[1] https://github.com/ruotianluo/pytorch-faster-rcnn

5.1.1.2. Spatial Prediction

The spatial prediction module infers each target location in following frames. We use an eight dimensional state-space for each target, containing its bounding box center position (x, y), aspect ratio (r), height (h), and respective velocities (vx, vy, vr, vh). We employ Kalman filtering [70] for predicting the state space. For updating the predictions, we use the associated filtered detections as observations in the model update module. State prediction is performed at the end of the current frame, being employed for data association in the next frame.

5.1.1.3. Data Association

The data association module matches the filtered detections with the trajectories of tracked targets by using the Hungarian algorithm[52]. We propose to perform association in two stages. First, we use appearance features to match detections and predicted targets. Similarity is computed as the cosine distance between the detection appearance descriptor and the target appearance model (i.e. the last N appearances of the target). Second, we consider the unmatched detections and predictions in the previous stage and we apply again the Hungarian algorithm using their spatial predicted descriptors (i.e. bounding boxes). The similarity between bounding boxes is determined on the basis of the Intersection over-Union criterion[53].

5.1.1.4. Track Management

The track management module is in charge of operations such as track initialization and suppression. We employ two counters per track for handling initialization and suppression. One counter focuses on the number of consecutive frames where the track is kept. Another counter focuses on the number of consecutive frames where the track is lost. Track initialization is defined when unmatched detections exist and the first counter is above a threshold (*min_life*) whereas track suppression is performed when the second counter is above another threshold (*max_unmatched*).

5.1.1.5. Model Update

The model update module keeps a buffer of the last appearances for each track (i.e. features vector of detections associated to the track).

5.1.2. Results

We evaluated our approach (FRMOT) on the VisDrone 2018 Benchmark [54] held in ECCV 2018. Table 22 shows the ranking of the challenge. Although our algorithm (FRMOT) ranks 4.0, due to the averaging of the ten metrics that are considered, we obtain better MOTA, IDF1, FAF, MT, ML, FP, FN, IDS and FM than at least one or more algorithms. Figure 15 depicts a sample frame with the identifiers and bounding boxes of the tracked vehicles.

Table 22. From [48], multi-object tracking results on the VisDrone-VDT2018 testing set. Rank is computed averaging ten metrics. Algorithms with $*$ were submitted by the commitee

Method	Rank	MOTA	MOTP	IDF1	FAF	MT	ML	FP	FN	IDS	FM
V-IOU	2.7	40.2	74.9	56.1	0.76	297	514	11838	74027	**265**	**1380**
TrackCG	2.9	**42.6**	74.1	**58.0**	0.86	323	395	14722	68060	779	3717
GOG_EOC	3.2	36.9	**75.8**	46.5	**0.29**	205	589	**5445**	86399	354	1090
SCTrack	3.8	35.8	75.6	45.1	0.39	211	550	7298	85623	798	2042
Ctrack	3.9	30.8	73.5	51.9	1.95	**369**	**375**	36930	**62819**	1376	2190
FRMOT	4.0	33.1	73.0	50.8	1.15	254	463	21736	74953	1043	2534
GOG* [37]	-	38.4	75.1	45.1	0.54	244	496	10179	78724	1114	2012
IHTLS* [11]	-	36.5	74.8	43.0	0.94	245	446	14564	75361	1435	2662
TBD* [15]	-	35.6	74.1	45.9	1.17	302	419	22086	70083	1834	2307
H^{2}T* [54]	-	32.2	73.3	44.4	0.95	214	494	17889	79801	1269	2035
CMOT* [3]	-	31.5	73.3	51.3	1.42	282	435	26851	72382	789	2257
CEM* [34]	-	5.1	72.3	19.2	1.12	105	752	21180	116363	1002	1858

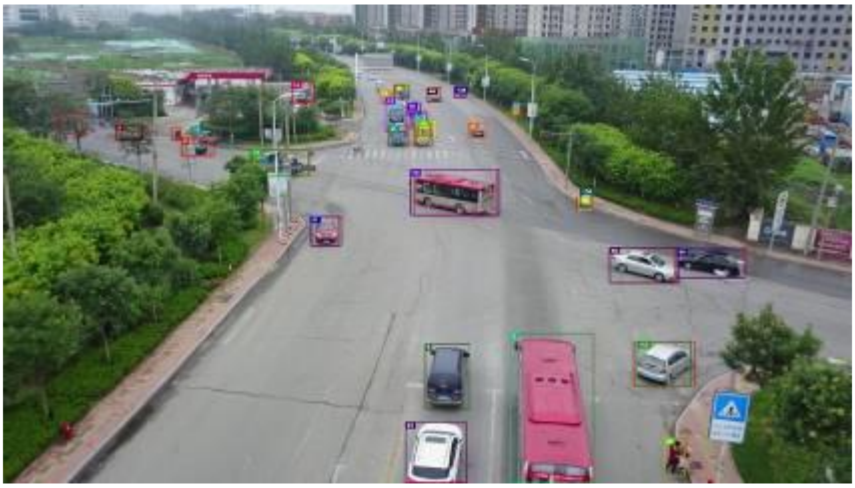


Figure 15. Sample frame with tracking results of one the sequences of the VisDrone 2018 dataset. Numbers stand for the identifiers of the tracked vehicles.

5.2. Multi-target tracking

5.2.1. Description of the algorithm

The proposed Multi Target Multi Camera (MTMC) tracking method was published in the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2019 [56] within the scope of CityFlow: A City-Scale Benchmark for Multi-Target MultiCamera Vehicle Tracking and Re-Identification [57].

The proposed tracking approach is mainly composed of two main blocks, as shown in Figure 14, for analysing data in single and multiple cameras set-ups, respectively. The first block aims to detect and track vehicles from each independent camera. The second block performs tracking across multiple cameras by modelling appearance of bounding boxes detected for each camera; projects them into a common plane to group detections of the same object coming from different cameras; and, finally, associates trajectories over time to compute the final tracks.

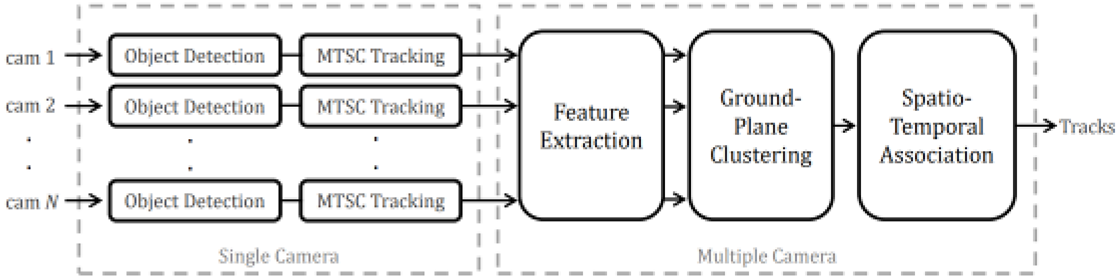


Figure 16. Block diagram of the proposed tracking method.

5.2.1.1. Single-camera Tracking and Object Detection

Multi Target Single Camera (MTSC) tracking is performed solving the tracking-by-detection problem. The CityFlow benchmark provides detections as bounding boxes using three popular detectors: YOLOv3[58], SSD512 [59] and Faster R-CNN [49]. These three detectors make use of pre-trained models on the COCO benchmark [60] and the threshold value of 0.2 is applied to finally obtain the detections. For tracking based on these detections, two online approaches such as DeepSORT [61] and MOANA [62] are employed, and also TC [63] as an off-line method. The CityFlow benchmark provides results for nine MTSC tracking solutions by combining the above-mentioned detectors (three) and trackers (three).

5.2.1.2. Feature Extraction

Feature extraction module models the appearance of each detected box via deep learning features by considering the AlexNet [64] and ResNet-101 [65] architectures, both pretrained on the ImageNet database [66]. Since ImageNet covers 1000 classes and we need to adapt the model to our target, i.e. vehicles, we train some layers of the network while leaving others frozen. In detail, ResNet-101 is frozen before $block3$, and AlexNet is frozen before $pool1$ layer, following [67]. To fine-tune the network, we have employed 36,935 sample images of 333 vehicle identities, extracted from the training set of ReID track 2 in the 2019 AI City Challenge. We also set the learning rate to $3e-4$ and batch size to 10. We train for 6 epochs and use Stochastic Gradient Descent with Momentum optimizer [68]. AlexNet architecture give us a

4096-dimensional feature vector at the output of $fc7$ layer, while we obtain a 2048-dimensional vector at $pool5$ layer in ResNet-101 network.

5.2.1.3. Ground-Plane Clustering

This module is in charge of associating detections of the same vehicle from different cameras obtained at the same time. At every frame, we project all detections of every camera to a common plane and apply hierarchical clustering to cluster such projected detections. In addition, we employ cluster validity indexes to determine which cluster structure is more suitable for our problem (i.e. find the optimal number of clusters).

For ground-plane projection, we use homography matrices from 2D image pixel location to GPS coordinates. Therefore, we consider GPS coordinates plane.

For clustering, we employ Hierarchical clustering based on two features: visual appearance and spatial distance in the ground-plane. Since two detections widely separated are highly unlikely to come from the same vehicle, we set a threshold such that the distance between vehicles' detections further than 6 meters in GPS plane is set to a much higher value, i.e. impossible association. Similarly, as two detections coming from the same camera cannot be merged into the same cluster, the distance between them is also set to the same high value (100 meters). By this way, two detections are more likely to fit the same vehicle if they are spatially close on the ground-plane and have similar visual appearance.

Ideally, each cluster represents a vehicle and it can be composed of several detections from different cameras or composed of merely one detection. As the number of the number of clusters is unknown a priori, we have to determine empirically such optimal number. We therefore validate different clustering results using validation indexes. We use internal validation, more specifically, Dunn's index [69], which aims to identify dense and well-separated clusters. By this way, all possible associations with different number of clusters are computed and we obtain an index value for each one. We obtain the optimal number of clusters, i.e. the number of vehicles, by taking the index with maximum derivative, i.e. the point of higher gradient. We empirically found that maximum derivative provides better information than maximum value.

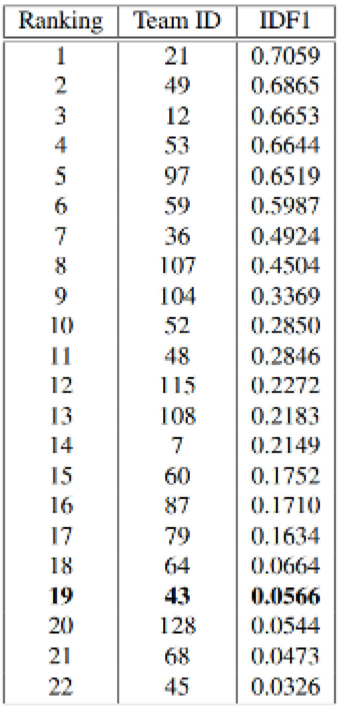
5.2.1.4. Spatio-Temporal Association

The following task, consisting on linking clusters over time, is performed by the spatio-temporal association module. Positions of each cluster along time form a track. Tracks motion is estimated via a constant-velocity Kalman Filter [70] and association between clusters and predicted tracks is performed by the Hungarian Algorithm [52] using Euclidean distance between the spatial distances. As for track management, we initialize tracks for clusters (i.e. associated detections across cameras) that remain unassigned for 10 frames. Moreover, we also remove tracks which are not associated to any cluster for 20 consecutive frames.

5.2.2. Initial results

Leaderboard of CityFlow Challenge is shown in Table 23. This classification ranks identification precision (IDF1) on the test scenarios (S02 and S05). Both scenarios comprise a total of 23 cameras. S02 is formed by 4 confronted cameras in a road intersection. However, S05 consists of 19 cameras, spread out over a wide extension, where maximum distance between two cameras is 2.5 kilometres. It is important to remark that cameras in S02 are completely overlapped between each other, while in S05 there is no overlap between most of them. Since our approach is completely dependent on projections, and therefore on overlap, predictably, it results in a low performance, as can be seen in Table 2.

Table 23. Leaderboard of City-Scale Multi-Camera Vehicle Tracking, evaluated on test scenarios



Ranking	Team ID	IDF1
1	21	0.7059
2	49	0.6865
3	12	0.6653
4	53	0.6644
5	97	0.6519
6	59	0.5987
7	36	0.4924
8	107	0.4504
9	104	0.3369
10	52	0.2850
11	48	0.2846
12	115	0.2272
13	108	0.2183
14	7	0.2149
15	60	0.1752
16	87	0.1710
17	79	0.1634
18	64	0.0664
19	**43**	**0.0566**
20	128	0.0544
21	68	0.0473
22	45	0.0326

Figure 17 shows tracking results for scenario S01, formed by confronted cameras, in a similar way to S02.

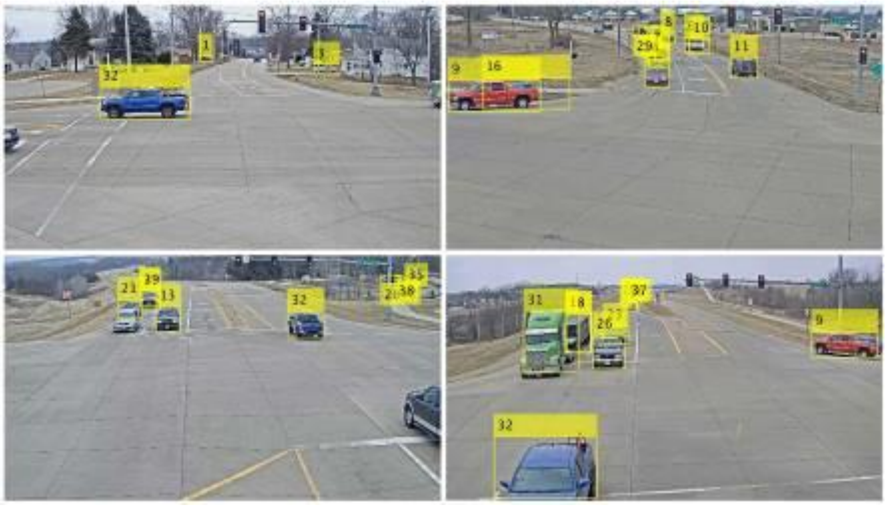


Figure 17. Sample visual results in train scenario S01, cameras 1-4. Tracked vehicles in yellow with their correspondent IDs. Same blue car is identified with the same ID, as well as the red car. However, an error in the single camera tracking leads to a tracking error in the red car in camera 2.

5.2.3. Further extensions and experiments

5.2.3.1. Window-based processing

This extension affects the ground-plane clustering module. Due to noise in the video transmission while capturing the data of the CityFlow benchmark [88], some frames are skipped within some videos, so some cameras suffer from a few misalignments of synchronization along time. We can observe this misalignment in the figure below, where the red car appears at different position on the road depending on the camera view.

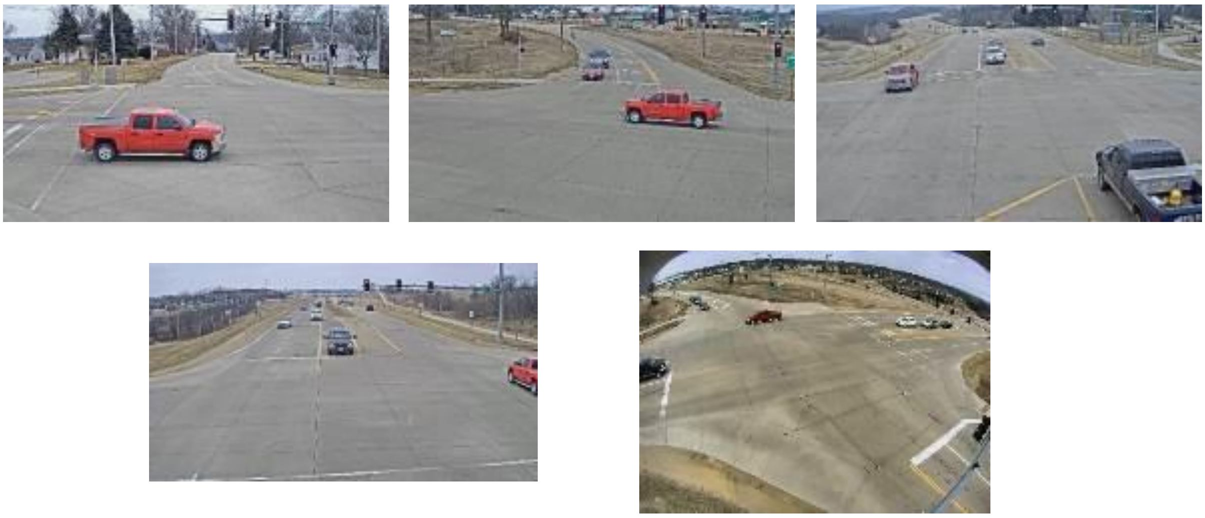


Figure 18. Cameras 1-5 at frame 291.

In order to deal with this bas synchronization, we designed and implement a new version of the Multi Target Multi Camera (MTMC) tracking algorithm, employing temporal window-based processing. The figure below depicts a diagram of the proposed window. Windows may have a variable size, as well as variable stride.

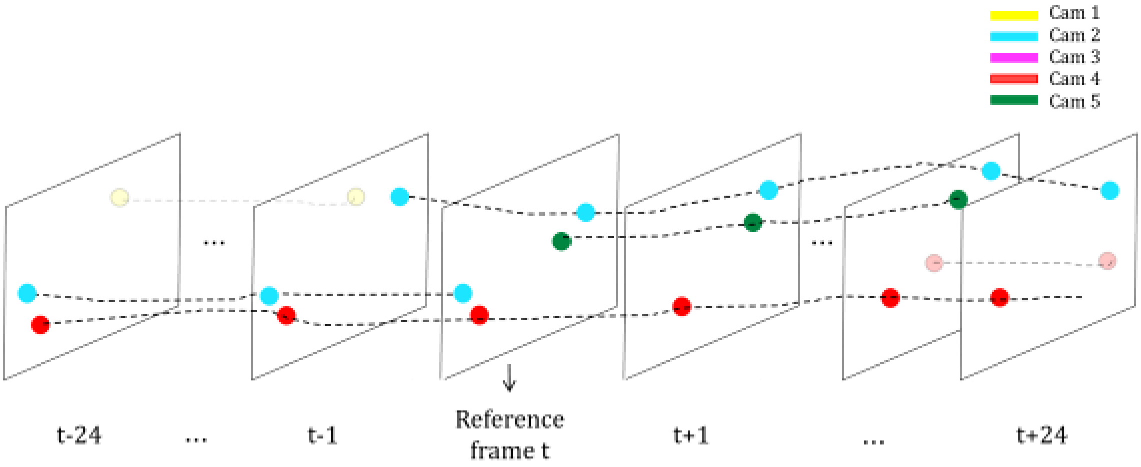


Figure 19. This example considers a window of 49 frames. Only the trajectories in the reference frame are considered.

The proposal considers only the trajectories that are present in the frame under analysis. As appearance descriptor we considered the average of all the descriptors in the trajectory. The similarity between appearance descriptors is computed as the Euclidean distance, as in the original approach. The spatial distance between trajectories is computed using Dynamic Time Warping [89]. In order to evaluate the approach individually, we assessed only the ground-plane clustering results, without performing the spatio-temporal association. The table shows Precision, Recall, F-Score, the number of vehicles in the ground-truth, the number of computed vehicles, True Positives, False Positives, False Negatives, the window size W and the appearance model used, i.e. ResNet101 as backbone with pre-trained weights on Imagenet.

The following table indicates that the proposed algorithm with W = 1, 5 and 11 frames works in a very similar way, however increasing the window size decreases considerably the performance. From the data we have observed than the misalignment is such that at least a window of 40 frames is required to join misaligned trajectories. Also, it is important to remark that the bigger the window is, the higher is the computational cost and time. Having this and the results under consideration, we decided not to follow this line of work. In addition, window-based processing would lose the causality of the approach.

Table 24. Clustering performance of the proposed approach varying the window size W

P	R	F	GT	PRED	TP	FP	FN	W	MODEL
21,78	57,43	31,11	20772	53433	11733	41700	41700	1	ResNet101 Imagenet
21,59	57,05	30,85	20772	53433	11618	41815	41815	5	ResNet101 Imagenet
21,79	57,46	31,12	20772	53433	11728	41705	41705	11	ResNet101 Imagenet
19,85	53,01	28,45	20772	53433	10627	42806	42806	49	ResNet101 Imagenet

5.2.3.2. Improvement of feature appearance: vehicle discriminator Siamese network

We have design, trained and evaluated a siamese network architecture for discriminating pairs of given vehicles at different camera views. The figure below illustrates an overview of the block diagram. The network requires at the input a pair of images, in the form of bounding boxes, depicting two view of vehicles. Both bounding boxes feed a Convolutional Neural Network (CNN) in order to extract their N-dimensional feature descriptors. Both descriptors are concatenated to compute a 2N-dimensional embedding. Lastly, a classifier provides the likelihood of the pair of images depicting the same vehicle.

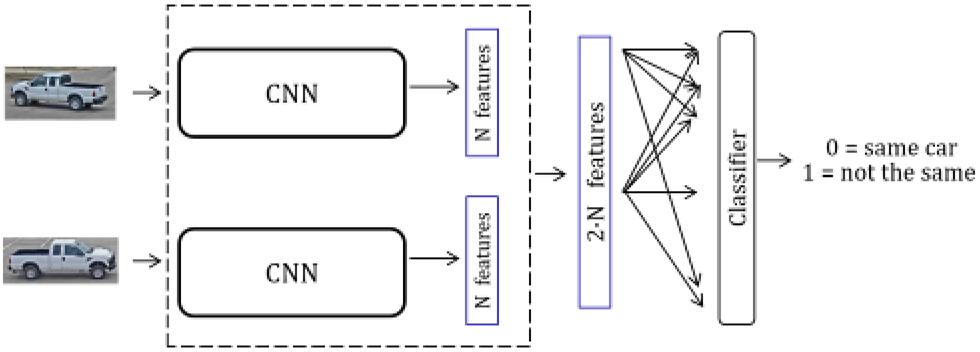


Figure 20. Block diagram of the vehicle discriminator network

The following figure shows samples of pairs and the class label associated to them. This figure also displays the viewpoint variation problem, the major challenge in MTMC vehicle tracking. Due the intrinsic geometry, distinct vehicles may appear quite similar from the same viewpoint, however the same vehicle from different viewpoints may be difficult to recognise. It can be extremely hard, even for humans, to determine if two vehicles from different points of view depict the same car.

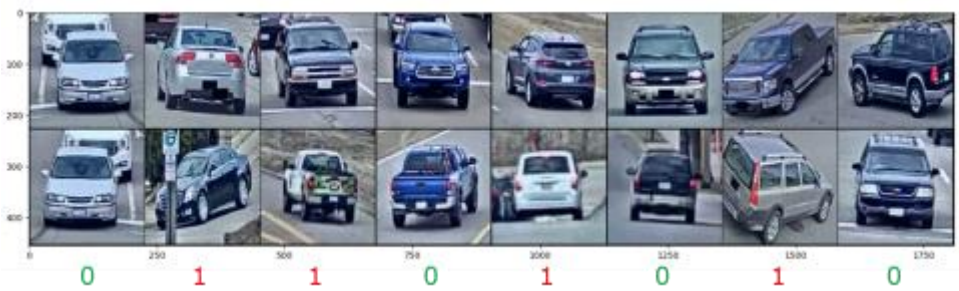


Figure 21. Samples of pairs of vehicles and the corresponding class label: 0 (same vehicle) and 1 (different vehicle)

As the CNN backbone we employed ResNet-50 pretrained on the Imagenet dataset, and the feature embeddings are taken just after the last average pool layer and before the fully connected layer (fc_1 layer). Thus, N = 2048. The classifier is composed by a batch normalization, ReLU and a 4096-d fully connected layer.

5.2.3.3. Regularization techniques

New mixup training proposal: siamese mixup

As a regularization technique, to deal with the overfitting problem during training, the original mixup strategy was proposed by [90]. In essence, mixup trains a neural network on combinations of pairs of examples and their labels. By doing so, mixup regularizes the neural network. To apply the mixup strategy in the training of our siamese network we propose the siamese mixup. $\alpha \in [0,1]$ is the mixing weight. $label_x$ determines the vehicle ID. Figure 5 shows how we obtained two mixed images that will be the input of the discriminator network (see the following figure).

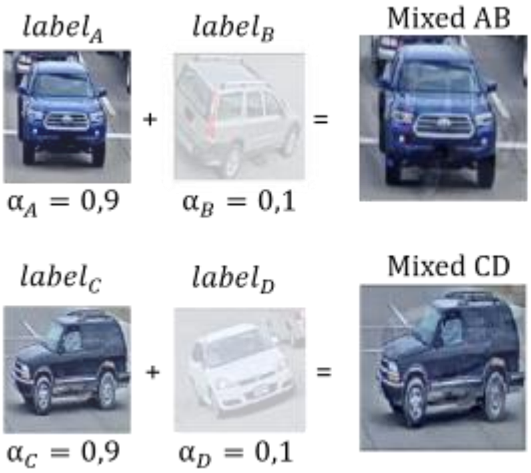


Figure 22. Example of siamese mixup strategy

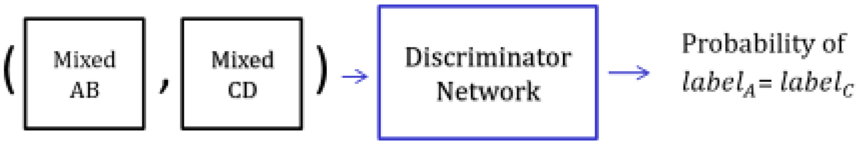


Figure 23. Block scheme of the siame mixup strategy

When training the discriminator network, the loss criterion employed has to be modified in the following way

$$\mathcal{L}_{Total} = \alpha_{AC} \cdot \mathcal{L}(p, label_{AC}) + \alpha_{BC} \cdot \mathcal{L}(p, label_{BC}) + \alpha_{AD} \cdot \mathcal{L}(p, label_{AD}) + \alpha_{BD} \cdot \mathcal{L}(p, label_{BD})$$

being $\alpha_{XY} = \alpha_X \cdot \alpha_Y$ and

$$label_{XY} = \begin{cases} 0, & label_X = label_Y \\ 1, & label_X \neq label_Y \end{cases}$$

By doing so, all the possible pair combinations are proportionally considered in the loss function. We used the Cross Entropy loss as loss criterion.

Dropout regularization

We also included the dropout strategy [91]. During training, some number of layer outputs are randomly ignored or “dropped out.” Dropout has the effect of making the training process noisy, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs. It results in a network that is capable of better generalization and is less likely to overfit the training data. This was simply implemented in the classifier just by adding a Dropout layer after the ReLU layer and before the fully connected layer.

Gradual warmup

The intuition behind warmup training strategies [92] is to help the network to slowly adapt to the data and also, to allow adaptive optimizers (e.g. Adam, RMSProp, …) to correctly compute the gradients. Gradual warmup consists in starting with a small learning rate and gradually increase it by a constant until it reaches the initial desired learning rate, see Figure 7.

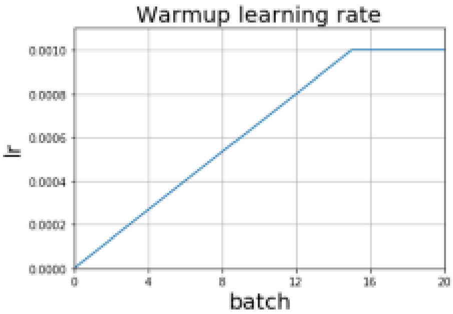


Figure 24. Gradual warmup training

Experiments

For validating our proposal, we have considered the training set of the CityFlow benchmark (S01, S03 and S04 scenarios). It comprises 129 vehicle IDs and 29669 bounding boxes (230 in average per ID). We randomly split the data into the training subset (90%) and the validation subset (10%). Each input image containing a bounding box of a vehicle is adapted to the network by resizing it to 224 x 224 x 3 and the pixels are normalized by the mean and standard deviation of ImageNet dataset.

We performed a validation methodology by entering pairs of vehicles to the network and computing the accuracy between the ground-truth and the network prediction. The ground-truth is computed by comparing the vehicle IDs (0 = same car, 1 = different car). Tables 2 and 3 shows the impact of the different strategies. They include the training batch size, the starting learning rate and the number of epochs for decaying the learning rate. Finetuning denotes that also the last encoder of ResNet-50 is trained.

Table 25. Ablation study with batch size = 100

BATCH SIZE	STARTING LR	DECAY LR	MIXUP	FINETUNE	DROPOUT (20 %)	WARMUP (0,001)	PREC. VAL	PREC. TRAIN
100	0,01	15	--	--	--	--	78,34	87,06
100	0,01	15	--	yes	--	--	89,47	94,69
100	0,01	15	--	yes	--	yes	88,71	97,19
100	0,01	15	yes	yes	--	--	85,49	94,82
100	0,01	15	yes	yes	yes	--	84,06	93,50
100	0,01	15	yes	yes	--	yes	88,72	94,59
100	0,01	15	yes	yes	yes	yes	83,88	95,44
100	0,01	15	--	yes	yes	yes	86,81	95,44

Table 26. Ablation study with batch size = 200

BATCH SIZE	LR	DECAY LR	MIXUP A	FINETUNE	DROPOUT (20 %)	WARMUP (0,001)	PREC. VAL	PREC. TRAIN
200	0,01	15	yes	--	--	--	73,95	81,44
200	0,01	15	yes	yes	--	--	85,47	93,89
200	0,01	15	yes	yes	yes	--	83,69	89,54
200	0,01	15	yes	yes	--	yes	86,04	93,46
200	0,01	15	yes	yes	yes	yes	82,99	94,98

The validation and trainig precision shown in the table are taken from the best epoch (i.e. when the validation loss reach a minimum peak), however these number may not be representative of the real behaviour of the model. For a better visualization Figure 8 shows the graphs of the training and validation loss during the training process. For instance, the fist graphic shows a peak performance of the validation loss in a very early epoch, however the loss afterwards, tends to increase.

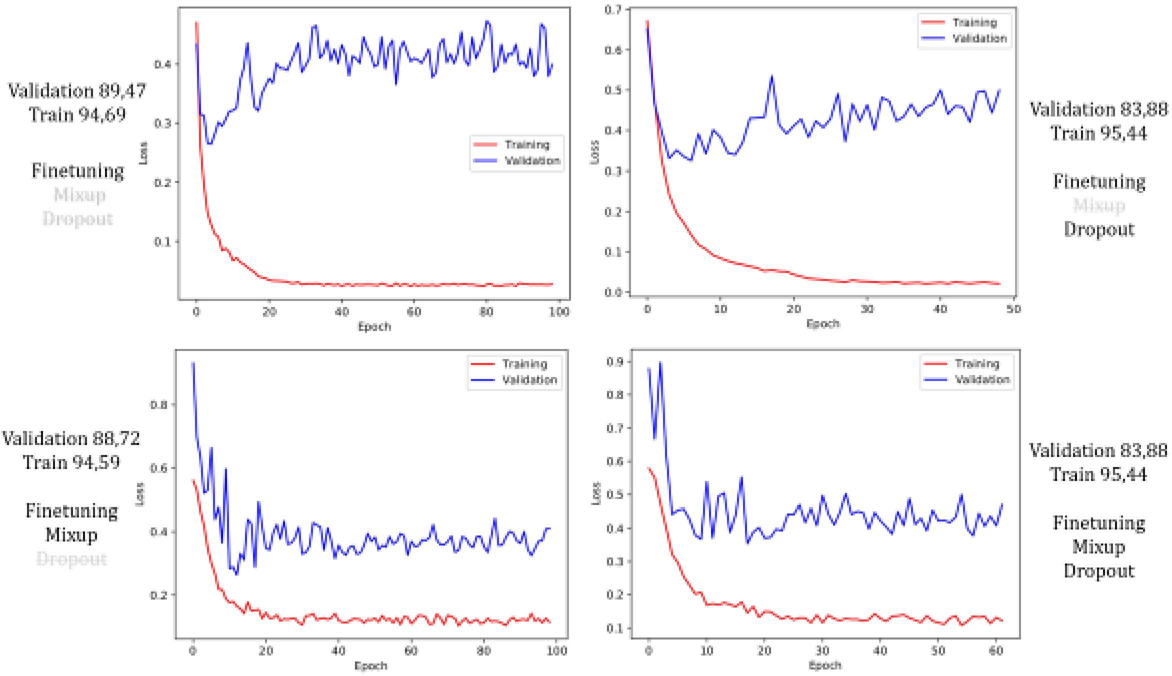


Figure 25. Training and validation losses. Batchsize = 100, starting LR = 0.01 and LR decay = 15

From the graphs we can observe that just fine-tuning the network with no additional regularization technique leads to a wrong training where the validation loss tends to increase, instead of decrease. Adding the dropout strategy helps a few to reduce the overfitting, however it does not have a great impact, since the validation loss still tends to increase. Including the mixup strategy makes really the difference in solving the overfitting problem.

5.2.3.4. New data: Veri-776 Dataset

We have also used VeRi-776 [93] dataset for improving the feature extraction model by increasing the training data. VeRi-776 is one of the largest and most commondataset for vehicle re-identification in multi-camera scenarios. It comprises about 50,000 bounding boxes of 776 vehi-cles captured by 20 cameras.

The following figure shows a comparison between considering only the Cityflow dataset and both of them. In these trainings BS = 100. Mixup, dropout and warmup strategies are used. ResNet-50 is also finetuned. The starting LR = 0.01. The graphics of the losses evolution show a correct training where both training and validation losses decreases and converge. The precision graphics show that the Cityflow training converge at around 80% of precision, while the combination of both datasets provides more than 90% of precision. Note also, that training with more data makes the curves to be smoother.

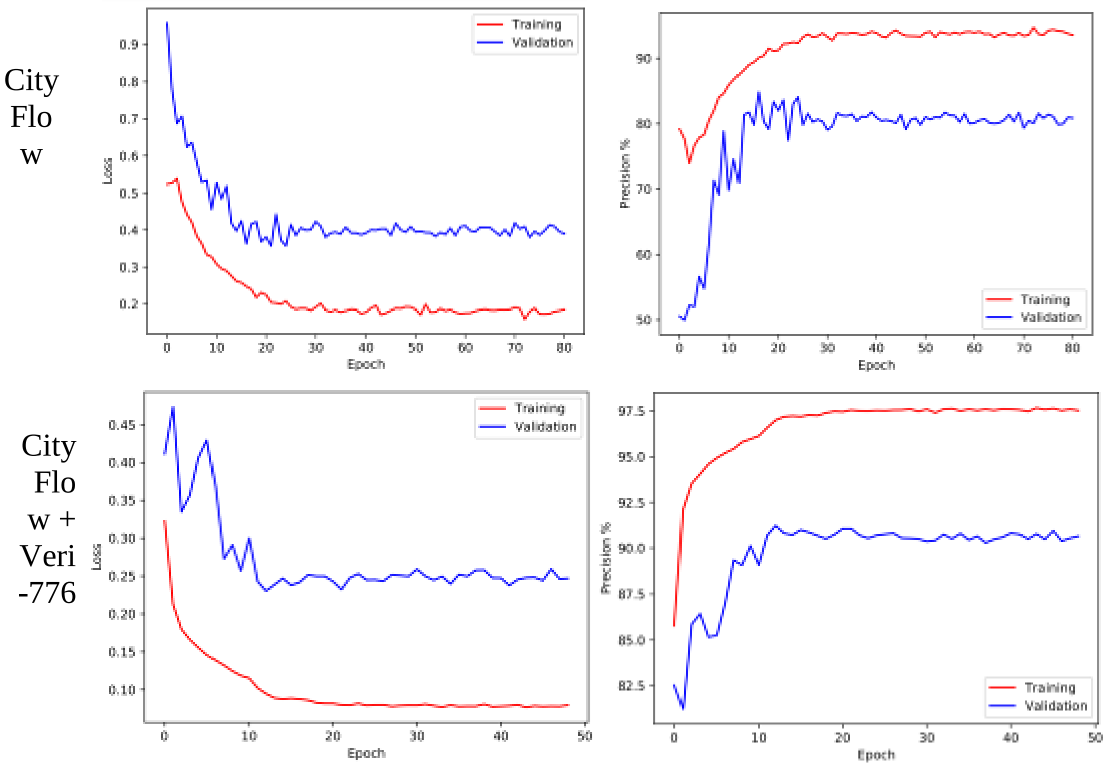


Figure 26. Training and validation loss and precision graphs

In order to show the complexity of the problem under consideration the net figure shows examples of pairs of vehicle the trained network fails to discriminate.

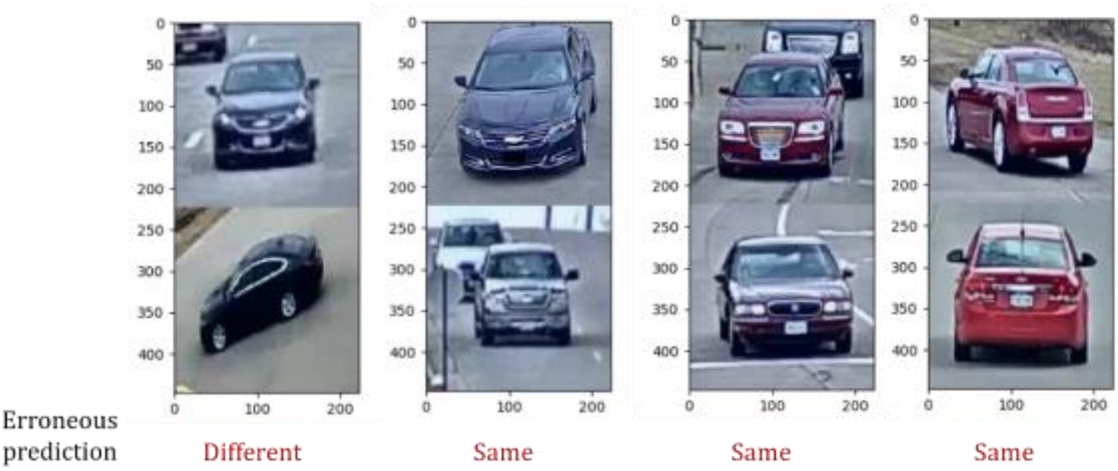


Figure 27. Samples of erroneous predictions

Note that while the first two error seems to be due to the bad quality of the images (due to far vehicles), the last two seems to be conducted by the colour of the vehicle.

5.2.3.5. Calibration

Confident calibration (the problem of predicting probability estimates representative of the true correctness likelihood) is important for classification models [94]. Classification networks must not only be accurate, but also should indicate when they are likely to be incorrect. In order to check whether our network is calibrated we computed the following reliability diagram. To compute this diagram, we analysed all the given predictions by intervals and compute the real accuracy of them.

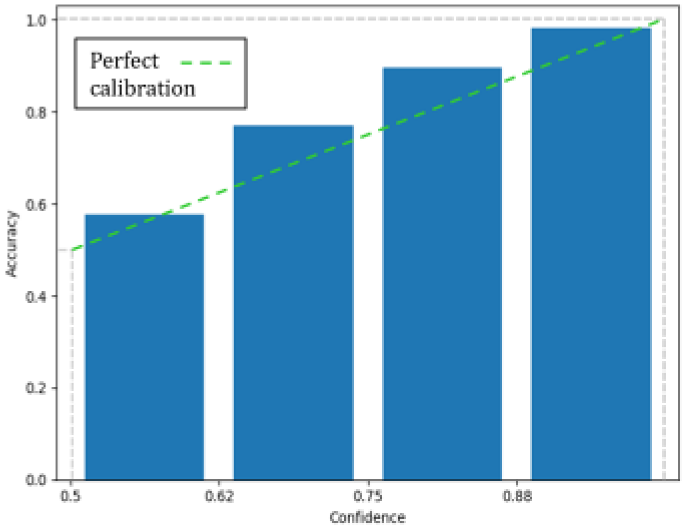


Figure 28. Reliability diagram. Confidence refers to the output prediction of the network, while the accuracy is computed by intervals.

5.2.3.6. Additional studies: study of the LR decay

We performed this study to check the influence of the BS and the starting LR jointly. The results indicates that 0.001 is the optimal starting LR in our approach. Also, we achieve a better performance when the LR decay is higher and BS = 64 seems to work best to our problem

Table 27. Ablation study for LR decay

BATCH SIZE	STARTING LR	DECAY LR	PRECISION VALIDATION MEAN / BEST	PRECISION TRAINING MEAN / BEST
64	0,001	20	75,82 / 78,48	94,12 / 94,09
64	0,001	30	75,70 / 79,35	95,08 / 94,80
64	0,0001	20	50,37 / 78,12	76,54 / 51,42

BATCH SIZE	STARTING LR	DECAY LR	MIXUP	DROPOUT (20 %)	WARMUP (0,001)	PREC. VAL	PREC. TRAIN	ADDITIONAL DATA AUG.
100	0.01	20	yes	yes	yes	83.76	95,66	---
100	0,01	20	yes	Yes	yes	88,56	93,38	Hue 0,05
100	0,01	20	yes	yes	yes	85,64	95,04	Rot 45º
100	0.001	20	yes	yes	yes	84.85	90,16	---
100	0,001	20	yes	yes	yes	84,31	92,78	Hue 0,05
100	0,001	20	yes	yes	yes	81,44	88,72	Rot 45º

64	0,0001	30	57,92 / 61,76	78,36 / 78,73
128	0,001	20	74,68 / 76,40	91,55 / 91,16
128	0,001	30	75,75 / 78,86	93,62 / 92,25
128	0,0001	20	49,70 / 51,08	76,73 / 75,78
128	0,0001	30	49,85 / 51,52	77,56 / 76,52
256	0,001	20	73,56 / 75,95	87,12 / 85,98
256	0,001	30	74,39 / 76,69	88,08 / 86,99
256	0,0001	20	50,25 / 51,15	75,86 / 75,84
256	0,0001	30	49,85 / 51,27	75,58 / 78,94

5.2.3.7. Additional studies: Hue and rotation data augmentation

Behind the commonly used data augmentation (i.e. random horizontal) we also experimented with hue and rotation data augmentation.

Hue augmentation sdd a random hue jitter to images. Hue can be thought of as the ‘shade’ of the colors in an image. Hue changing parameter is set to 0.05 in order to not to affect so much to the colour of the vehicles. The rotation augmentation randomly rotates the image clockwise by a given number of degrees from 0 to a given parameter, we used 45º.

Results on the table shows that these two techniques do not really impact in our approach. This may be due to the fact that due to the nature of the dataset cars already appears in different illumination condition (depending on the camera) and in different rotation angles.

Table 28. Ablation study for data augmentation

5.3. People detection in omnidirectional cameras

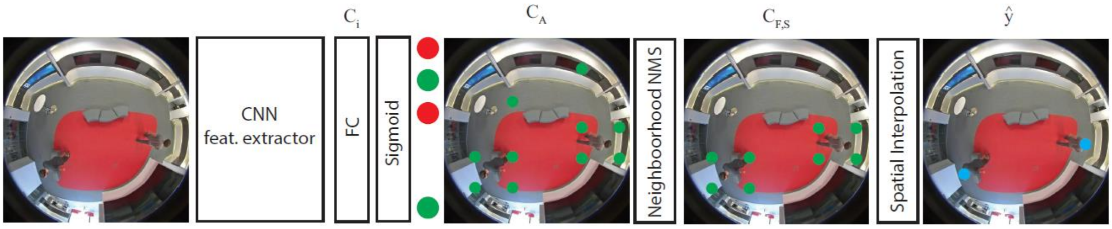


Figure 29 Block scheme of the proposed DL-GSAC object detection system. The circles at the output of the CNN represent the predictions of the grid of classifiers. Green circles correspond to active classifiers. Red circles correspond to inactive classifiers. For clarity, inactive classifiers are omitted from the image.

This work [95] continues the work in [96] in the development of a people detection system for omnidirectional cameras. The main advantages of the detector design are two-fold: the operation of the system with omnidirectional cameras allows to cover larger areas with a single camera, and the point-based annotation of persons for the training stage (instead of bounding boxes) alleviates the annotation requirements in the deployment of the system.

Specifically, the work in [95] in improves the original system, described in [97], adapting the Grid of Spatially Aware Classifiers (GSAC) to an end-to-end deep learning architecture (DL-GSAC). The inclusion of a CNN-based architecture for the descriptor and classifiers allows to increase the generalization capability of the system, allowing to train a single detection model for different scenarios. This overcomes the main limitation of the GSAC version described in [97], based in HOG descriptors and SMV classifiers, which must be re-trained for every specific camera setting. The block scheme of the implemented system is depicted in Figure 29.

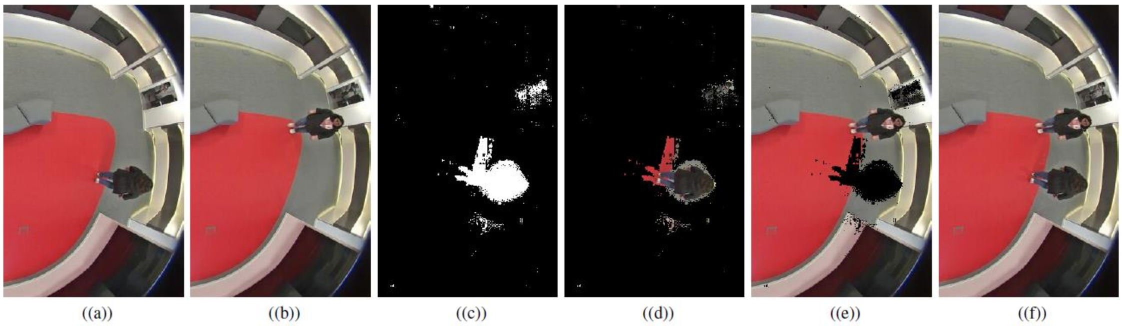


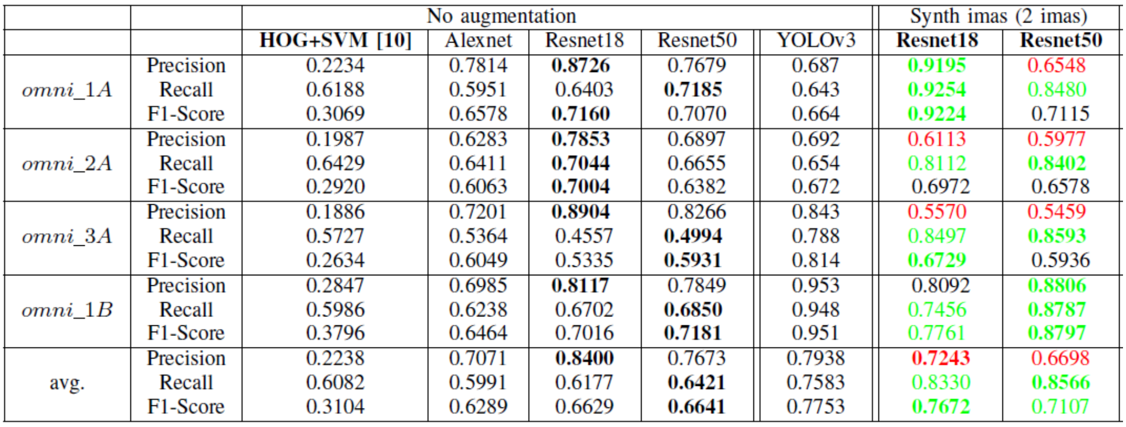
Figure 30 Example of the process of creation of a multiple-people synthetic training sample. a) and (b) show the original training images. Using a GMM background subtraction algorithm, a mask of the person is created, (c) and (d). The inverse mask is used on one original image to remove the corresponding region (e) and the extracted person is added to generate the final synthetic image (f).

Additionally, to improve the performance of DL-GSAC has been improved including the following features:

- **Positive sample weighting to alleviate training class imbalance:** One of the inherent problems in the training process of one-stage object detectors is the foreground-background class imbalance [98]. In the DL-GSAC architecture, the number of images that are considered a positive sample for a given classifier is consistently lower than negative samples, biasing the classifiers towards low scores. To mitigate the effect of this imbalance, in each classifier, we weight the loss of positive samples according to the ratio of negative samples with respect to each positive for that classifier.

- **Synthetic multiple-people data augmentation:** In the PIROPO dataset [97], the training data is composed by sequences of a single person walking through the room, covering all possible locations. The work in [97] shows that the requirement of such limited training data (plus the point-based annotations) supposes an advantage in the annotation requirements for the deployment of the HOG+SVM-GSAC system. However, when training DL-GSAC to cope with multiple scenarios, the detection performance drops in images with multiple people. Thus, here we explore whether this limitation can be overcome by creating additional synthetic training samples that fuse multiple people in a single image, and thus not requiring additional effort to collect or annotate new training data. An example of the creation of this multiple-people training samples is depicted in Figure 30.

Table 29 Comparative results between different GSAC architectures, HOG+SVM [97], Alexnet and Resnet-18/50, without additional data augmentation and YOLOv3. Also, the table includes performance results of DL-GSAC improved with multiple-people synthetic data augmentation. Text in bold indicate the best value for comparable GSAC configurations. In the data augmentation tests, green/red indicate a relevant improvement/decrease in the metric with respect to the corresponding baseline (no data augmentation).



		No augmentation					Synth imas (2 imas)	
		HOG+SVM [10]	Alexnet	Resnet18	Resnet50	YOLOv3	**Resnet18**	**Resnet50**
$omni\_1A$	Precision	0.2234	0.7814	**0.8726**	0.7679	0.687	**0.9195**	0.6548
	Recall	0.6188	0.5951	0.6403	**0.7185**	0.643	**0.9254**	0.8480
	F1-Score	0.3069	0.6578	**0.7160**	0.7070	0.664	**0.9224**	0.7115
$omni\_2A$	Precision	0.1987	0.6283	**0.7853**	0.6897	0.692	0.6113	0.5977
	Recall	0.6429	0.6411	**0.7044**	0.6655	0.654	0.8112	**0.8402**
	F1-Score	0.2920	0.6063	**0.7004**	0.6382	0.672	0.6972	0.6578
$omni\_3A$	Precision	0.1886	0.7201	**0.8904**	0.8266	0.843	0.5570	0.5459
	Recall	0.5727	0.5364	0.4557	**0.4994**	0.788	0.8497	**0.8593**
	F1-Score	0.2634	0.6049	0.5335	**0.5931**	0.814	**0.6729**	0.5936
$omni\_1B$	Precision	0.2847	0.6985	**0.8117**	0.7849	0.953	0.8092	**0.8806**
	Recall	0.5986	0.6238	0.6702	**0.6850**	0.948	0.7456	**0.8787**
	F1-Score	0.3796	0.6464	0.7016	**0.7181**	0.951	0.7761	**0.8797**
avg.	Precision	0.2238	0.7071	**0.8400**	0.7673	0.7938	**0.7243**	0.6698
	Recall	0.6082	0.5991	0.6177	**0.6421**	0.7583	0.8330	**0.8566**
	F1-Score	0.3104	0.6289	0.6629	**0.6641**	0.7753	**0.7672**	0.7107

The main results of the system are presented in Table 29. The detector is evaluated using the PIROPO dataset [97], using a single model trained with the training data of the four omnidirectional cameras. In these results, we compare Precision-Recall performance of DL-GSAC with the HOG+SVM-GSAC version of [97] and YOLOv3 [8]. For DL-GSAC, different CNN backbone architectures have been evaluated (Alexnet, Resnet-18, Resnet-50). Additionally, the efficiency of the synthetic multiple people data augmentation is evaluated for the Resnet-18 and -50 DL-GSAC versions. The main conclusions of the experiments indicate that the performance of DL-GSAC is clearly superior to HOG+SVM-GSAC and comparable to state-of-the-art detectors (YOLOv3). Also, the inclusion of synthetic multiple-people training samples improve the performance of DL-GSAC, specially regarding the Recall metric. However, it also incurs in some cases in decrease in the Precision, which needs to be further investigated.

6. Conclusions

This current version of D3, recapitulates the current research outcomes from Workpackage 3, focusing on new proposals for Scene Recognition, Semantic Segmentation, Multiview Matching and Cooperative Detection and Tracking, in scenarios were at least one of the following aspects is covered: heterogeneous modalities, multiple cameras and mobile cameras. Evaluation has been rigorous, over public datasets (including some created within the project), and some of the approaches have been presented in international challenges.

References

[1] López-Cifuentes, A., Escudero-Viñolo, M., Bescós, J., & García-Martín, Á. (2019). Semantic-Aware Scene Recognition. arXiv preprint arXiv:1909.02410.

[2] D1.3v1: Evaluation Datasets. TEC2017-88169-R MobiNetVideo (2018-2020). Video Processing and Understanding Lab. July 2019.

[3] López-Cifuentes, A., Escudero-Viñolo, M., Bescós, J., & Carballeira, P. (2018). Semantic Driven Multi-Camera Pedestrian Detection. arXiv preprint arXiv:1812.10779.

[4] Cheng, X., Lu, J., Feng, J., Yuan, B., & Zhou, J. (2018). Scene recognition with objectness. Pattern Recognition, 74, 474-487.

[5] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 413–420.

[6] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3485–3492.

[7] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (6) (2018) 1452–1464.

[8] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[9] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[10] Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., and Fleuret, F. (2018). WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5030-5039).

[11] Chavdarova, T. Deep multi-camera people detection. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 848-853), 2017, December.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in NIPS, 2012.

[13] K. Simonyan and A. Zisserman, Very deep convolutional networks for large scale image recognition, International Conference on Learning Representations, ICLR, 2015.

[14] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2017.

[15] Object detection and association in multiview scenarios based on Deep Learning, Paula Moral de Eusebio (advisor: Álvaro García-Martín), Trabajo Fin de Máster (Master Thesis), Master en Investigación e Innovación en TIC – Programa Internacional de Múltiple Titulación IPCV (Image Processing and Computer Vision Master Program), Univ. Autónoma de Madrid, Jul. 2019.

[16] Elena Luna, Paula Moral, Juan C. SanMiguel, Álvaro García-Martín, José M. Martínez, "VPULab participation at AI City Challenge 2019", Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR2019), Long Beach, CA, USA, Jun. 2019, in press.

[17] Re-identificación de personas, Daniel Sáez García, (Tutor: Álvaro García Martín), Trabajo Fin de Grado, Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Jul. 2019.

[18] ImageNet database: http://www.image-net.org/ (accessed Jul. 2019)

[19] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, Konrad Schindler, MOT16: A Benchmark for Multi-Object Tracking, proceedings of the IEEE Computer Vision and Pattern Recognition conference, 2016.

[20] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. Transactions on IEEE Pattern Analysis and Machine Intelligence, 36(8):1532–1545, 2014.

[21] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. Transactions on IEEE Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010b.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection, proceedings of the IEEE Computer Vision and Pattern Recognition conference, 2005.

[23] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks, proceedings of the IEEE Computer Vision and Pattern Recognition conference, 2015.

[24] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2013.

[25] R. B. Girshick, F. N. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2014.

[26] Alvaro Garcia-Martin, Ricardo Sanchez-Matilla, José M. Martinez. Hierarchical detection of persons in groups. In Signal, Image and Video Processing, 2017.

[27] Srikrishna Karanam, Mengran Gou, Ziyan Wu, Angels Rates-Borras, Octavia Camps, Richard J. Radke, A Systematic Evaluation and Benchmark for Person Re-Identification: Features, Metrics, and Datasets, IEEE Transactions on Pattern Analysis and Machine Intelligence, accepted February 2018.

[28] D. Gray and H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, proceedings of the IEEE European Conference on Computer Vision, 2008.

[29] L. Zheng et al, Scalable person re-identification: A benchmark, proceedings of the IEEE International Conference on Computer Vision, 2015.

[30] G. Lisanti et al., Person re-identification by iterative re-weighted sparse ranking, IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 8, pp. 1629–1642, 2015.

[31] T. Matsukawa et al., Hierarchical gaussian descriptor for person re-identification, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2016.

[32] Y. Taigman et al., Deepface: Closing the gap to human-level performance in face verification, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2014.

[33] L. Zheng et al., MARS: A video benchmark for large-scale person re-identification, proceedings of the IEEE European Conference on Computer Vision, 2016.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in NIPS, 2012.

[35] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep residual learning for image recognition, proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[36] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks, proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 2017.

[37] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceeding of AAAI, 2017.

[38] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, On pre-trained image features and synthetic images for deep learning, in Proceedings of the European Conference on Computer Vision, 2018.

[39] [54] N. Qian, On the momentum term in gradient descent learning algorithms, Neural networks, vol. 12, no. 1, pp. 145-151, 1999.

[40] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, Person re-identification by local maximal occurrence representation and metric learning, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2197-2206, 2015.

[41] F. Xiong, M. Gou, O. Camps, and M. Sznaier, Person re-identification using kernelbased metric learning methods, in Proceedings of the European Conference on Computer Vision, pp. 1-16, Springer, 2014.

[42] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, Large scale metric learning from equivalence constraints, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2288-2295, IEEE, 2012.

[43] D. Gray and H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, proceedings of the IEEE European Conference on Computer Vision, 2008.

[44] L. Zheng et al, Scalable person re-identification: A benchmark, proceedings of the IEEE International Conference on Computer Vision, 2015.

[45] M. Gou et al, DukeMTMC4ReID: A large-scale multi-camera person re-identification dataset, In CVPR Workshops, 2017.

[46] Zheng Tang et al, CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification, proceedings of the IEEE Computer Vision and Pattern Recognition conference, 2019.

[47] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. C. Anastasiu, and J.-N. Hwang, Cityfow: A city-scale benchmark for multi-target multicamera vehicle tracking and re-identification, in CVPR 2019: IEEE Conference on Computer Vision and Pattern Recognition, 2019. NVIDIA AI City Challenge. https://www.aicitychallenge.org.

[48] Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., ... & Liu, X., VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results. In Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[49] Ren, S., He, K., Girshick, R., Sun, J., Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the conference on Neural Information Processing Systems (NIPS), 2015.

[50] He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[51] Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., Spatial transformer networks. In Proceedings of the conference on Neural Information Processing Systems (NIPS), 2015.

[52] Kuhn, H.W., The hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97 (1955)

[53] Čehovin, L., Leonardis, A., Kristan, M., Visual object tracking performance measures revisited. In IEEE Transactions on Image Processing 25(3), 1261–1274 (2016)

[54] Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. (2018). Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437.

[55] Bernardin, K., Stiefelhagen, R., Evaluating multiple object tracking performance: the clear mot metrics. Journal on Image and Video Processing 2008, 1 (2008)

[56] Luna, E., Moral, P., SanMiguel, J. C., Garcıa-Martın, A., & Martınez, J. M.. VPULab participation at AI City Challenge 2019. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

[57] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. arXiv preprint arXiv:1903.09254, 2019.

[58] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[59] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[60] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014

[61] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In Proceedings of IEEE International Conference on Image Processing (ICIP), 2017

[62] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. IEEE Access, 7:31934–31945, 2019

[63] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the conference on Neural Information Processing Systems (NIPS), 2012.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[67] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In Proceedings of the European Conference on Computer Vision (ECCV), 2018

[68] Ning Qian. On the momentum term in gradient descent learning algorithms. Neural networks, 12(1):145–151, 1999.

[69] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics, 4(1):95–104, 1974.

[70]] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1):35–45, 1960

[71] Incorporating Depth in Egocentric Perception, Andrija Gajic (advisor: Marcos Escudero Viñolo), Master Thesis, Erasmus Mundus Joint Master Degree in Image Processing and Computer Vision (IPCV), Univ. Autónoma de Madrid, Jul. 2020.

[72] S. Song, S. P. Lichtenberg, and J. Xiao, “Sun rgb-d: A rgb-d scene understanding benchmark suite,” in2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 567–576, 2015.

[73] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,”The Journal of Machine Learning Research, vol. 13, no. 1, pp. 723–773, 2012.

[74] Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147.

[75] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

[76] González Jiménez, M. (2017). Sistema multi-cámara distribuido basado en UNITY (Bachelor's thesis).

[77] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3223).

[78] Zhu, Y., Sapra, K., Reda, F. A., Shih, K. J., Newsam, S., Tao, A., & Catanzaro, B. (2019). Improving semantic segmentation via video propagation and label relaxation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8856-8865).

[79] Neuhold, G., Ollmann, T., Rota Bulo, S., & Kontschieder, P. (2017). The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4990-4999).

[80] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majum-dar, J Krishna Murthy, and K Madhava Krishna. The earthain’t flat: Monocular reconstruction of vehicles on steep andgraded roads from a moving camera. InIEEE/RSJ Interna-tional Conference on Intelligent Robots and Systems (IROS),pages 8404–8410, 2018.

[81] Tsung-Wei Huang, Jiarui Cai, Hao Yang, Hung-Min Hsu,and Jenq-Neng Hwang. Multi-view vehicle re-identificationusing temporal attention model and metadata re-ranking. InProceedings of the IEEE Conference on Computer Visionand Pattern Recognition (CVPR) Workshops, pages 434–442, 2019

[82] Kai Lv, Heming Du, Yunzhong Hou, Weijian Deng, HaoSheng, Jianbin Jiao, and Liang Zheng.Vehicle re-identification with location and time stamps. InIEEE Conference on Computer Vision and Pattern Recognition (CVPR)Workshops, 2019

[83] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In de-fense of the triplet loss for person re-identification.arXivpreprint arXiv:1703.07737, 2017.

[84] Relja Arandjelovi´c and Andrew Zisserman. Three things ev-eryone should know to improve object retrieval. InIEEEConference on Computer Vision and Pattern Recognition,pages 2911–2918, 2012.

[85] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, andAndrew Zisserman. Total recall: Automatic query expan-sion with a generative feature model for object retrieval. InIEEE International Conference on Computer Vision, pages1–8, 2007.

[86] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding.InProceedings of the IEEE Conference on Computer Visionand Pattern Recognition, pages 1318–1327, 2017.

[87] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jing-dong Wang, and Qi Tian. Scalable person re-identification:A benchmark. InProceedings of the IEEE international con-ference on Computer Vision, pages 1116–1124, 2015.

[88] Zheng Tang, Milind Naphade, Ming-Yu Liu, XiaodongYang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, DavidAnastasiu, and Jenq-Neng Hwang. Cityflow: A city-scalebenchmark for multi-target multi-camera vehicle trackingand re-identification.arXiv preprint arXiv:1903.09254,2019

[89] Müller, M. (2007). Dynamic time warping. Information retrieval for music and motion, 69-84.

[90] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiao-gang Wang. Learning deep neural networks for vehicle re-idwith visual-spatio-temporal path proposals. InProceedingsof the IEEE International Conference on Computer Vision,pages 1900–1909, 2017

[91] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

[92] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[93] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[94] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. arXiv preprint arXiv:1706.04599.

[95] People detection in omnidirectional cameras: development of a deep learning architecture based on a spatial grid of classifiers, Enrique Sepúlveda Jorcano,

(advisor: Pablo Carballeira López), Master Thesis, Image Processing and Computer Vision Master Program, Univ. Autónoma de Madrid, Jul. 2020.

[96] Adaptación de un sistema de detección de personas en cámaras omnidireccionales a descriptores Deep Learning (Adaptation of a people detection system for omnidirectional cameras to Deep Learning descriptors), Nicolás García Crespo, (advisor: Pablo Carballeira López), Trabajo Fin de Grado (Graduate Thesis), Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Jul. 2019.

[97] C. R. del-Blanco, P. Carballeira, F. Jaureguizar, N. García, Robust people indoor localization with omnidirectional cameras using a Grid of Spatial-Aware Classifiers, Signal Processing: Image Communication, (in press) 2021.

[98] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[99] Clasificación de imágenes con redes neuronales profundas mediante conjuntos de entrenamiento reducidos y aprendizaje "few-shot" (Image classification through reduced training sets and “few-shot” learning), Guillermo Eliseo Torres Alonso (advisor: Miguel Ángel García), Trabajo Fin de Grado (Graduate Thesis), Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Jun. 2019.

[100] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr and T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 1199-1208, doi: 10.1109/CVPR.2018.00131.

[101] Reconocimiento de escenas exteriores mediante redes neuronales profundas entrenadas con la base de datos places (Scene recognition using deep neural networks trained with the PLACES database), Santiago Vicente Moñivar (advisor: Miguel Ángel García), Trabajo Fin de Grado (Graduate Thesis), Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Oct. 2019.

[102] Reconocimiento no-supervisado de escenas mediante características extraídas de redes neuronales pre-entrenadas (Unsupervised scene recognition using features extracted from pre-trained neural networks), Alejandro Gilabert Ramírez (advisor: Miguel Ángel García), Trabajo Fin de Grado (Graduate Thesis), Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Jul. 2020.

[103] Unsupervised scene and place recognition based on features extracted from pretrained convolutional neural networks, Andreas Sebastian Wolters (advisor: Moritz Milde), Master Research Project, University of Amsterdam, Nov. 2017.