TEC2017-88169-R MobiNetVideo (2018-2020)

Visual Analysis for Practical Deployment of Cooperative Mobile Camera Networks

D2 v1

Feasibility studies algorithms and findings

Video Processing and Understanding Lab

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Supported by GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA, INDUSTRIA Y COMPETITIVIDAD

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 Carlos San Miguel Avedillo	Juancarlos.sanmiguel@uam.es

HISTORY

Version	**Date**	**Editor**	**Description**
0.1	24/04/2019	José M. Martínez	Initial draft version
0.2	30/04/2019	Álvaro García	Contributions: object detection and re-identification: VIPer, Market and City
0.3	20/07/2019	Marcos Escudero	Contributions: Scene categorization for lifelogging
0.4	23/07/2019	Pablo Carballeira	Contributions: Google Street view dataset for lifelogging applications
0.5	25/07/2019	Juan Carlos San Miguel	Contributions: multi-target tracking for UAV monitoring
0.6	26/07/2019	José M. Martínez	Editorial checking
1.0	26/07/2019		First version

CONTENTS:

1. Introduction

1.1. Motivation

Work package 2 (WP2) aims at performing a study of current technologies for applications related to heterogeneous camera networks where camera mobility plays a key role. Such studies will be performed on public datasets. If required, small scenarios will be recorded. The main objective is the identification of suitable state-of-the-art video analysis tools (e.g., segmentation, tracking and detection), by the implementation and evaluation of their performance in single mobile cameras for use as a baseline for comparison with the achievements to be developed within WP3 and WP4.

This deliverable describes the work related with tasks T.2.1 People tracking for active vision, T.2.2 Object detection for collision detection, T.2.3 Scene categorization for lifelogging and T.2.4 Multi-target tracking for UAV monitoring.

1.2. Document structure

This document contains the following chapters:

- Chapter 1: Introduction to this document
- Chapter 2: People tracking for active vision
- Chapter 3: Object detection for collision detection
- Chapter 4: Scene categorization for lifelogging
- Chapter 5: Multi-target tracking for UAV monitoring
- Chapter 6: Conclusions

2. People tracking for active vision

2.1. People detection with omnidirectional cameras

This work [1] adapts a people detection system for omnidirectional cameras [2] to deep learning descriptors. The original system is based on a grid of spatial-aware classifiers implemented by support vector machines (SVMs) distributed throughout the image, with an area of action called fovea. As a result, the classifier detection system is able to distinguish different types of appearances that occur in an omnidirectional image according to the spatial location of an object, in this a case, a person. This work used the Histogram of Oriented Gradients detector (HOG) [8] as the base of the image descriptor, providing satisfactory detection results at the expense of a costly training process for the SVMs

This work substitutes HOG descriptors with descriptors extracted from pre-trained Convolutional Neural Networks (CNNs), leveraging the training process of the SVMs and achieving detection results that are comparable (and superior in several cases) to those of the original work. Different CNN architectures have been evaluated AlexNet [20], VGG19 [25], Densenet201 [22] and MobileNetV2 [3], as well as descriptors from several layers at different depth levels. The evaluated CNN architectures have been trained with the ImageNet database [4], that comprises millions of images and thousands of classes. An example of detection results can be seen in Figure 1.

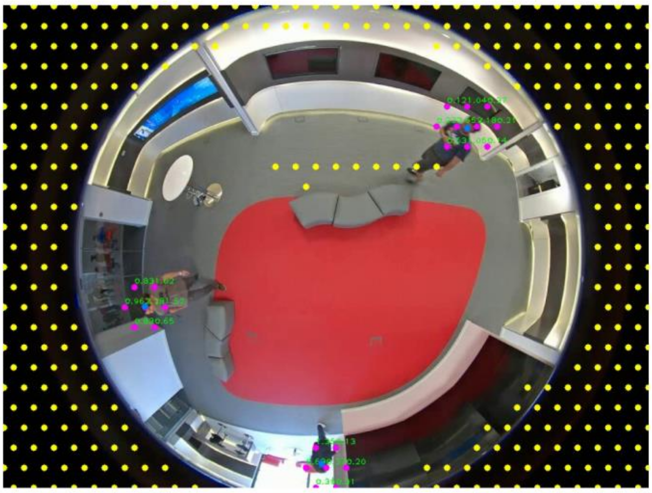
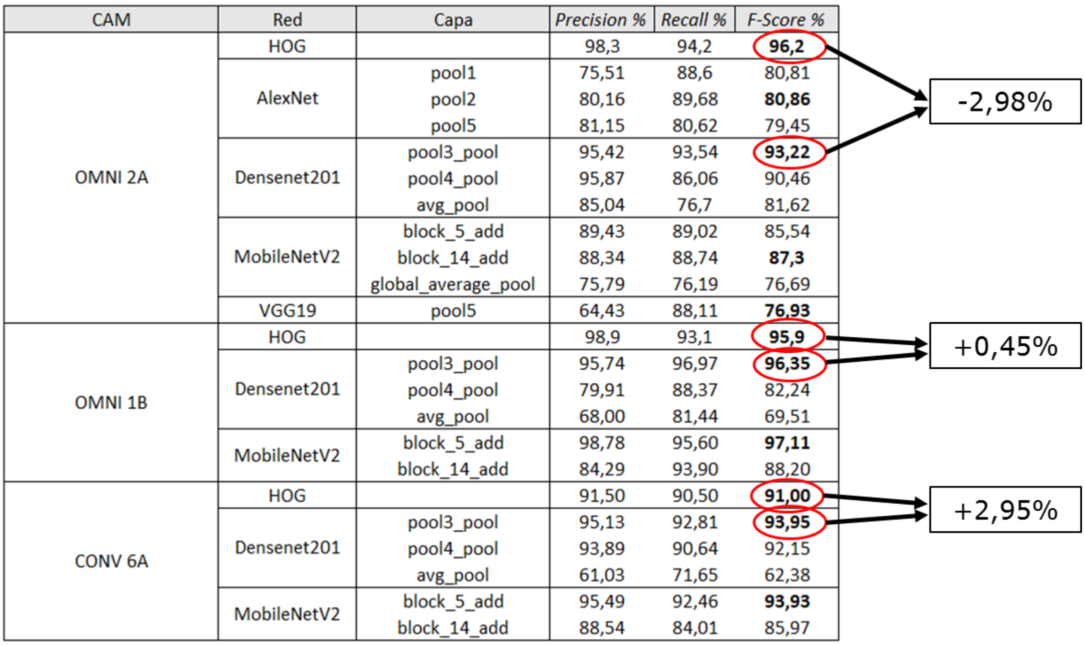


Figure 1. Example of detection results in an omnidirectional image. Purple dots represent the activated classifiers, blue dots represent the final person location. Yellow dots correspond to untrained classifiers.

The performance of the modified system has been evaluated in terms of Precision. Recall and F-Score metrics. Table 1 summarizes the results of several deep learning descriptors with respect to HOG. Densenet201 shows the best performance, improving the results of HOG for some of the cameras of the dataset.



CAM	Red	Capa	*Precision %*	*Recall %*	*F-Score %*
OMNI 2A	HOG		98,3	94,2	**96,2**
	AlexNet	pool1	75,51	88,6	80,81
		pool2	80,16	89,68	**80,86**
		pool5	81,15	80,62	79,45
	Densenet201	pool3_pool	95,42	93,54	**93,22**
		pool4_pool	95,87	86,06	90,46
		avg_pool	85,04	76,7	81,62
	MobileNetV2	block_5_add	89,43	89,02	85,54
		block_14_add	88,34	88,74	**87,3**
		global_average_pool	75,79	76,19	76,69
	VGG19	pool5	64,43	88,11	**76,93**
OMNI 1B	HOG		98,9	93,1	**95,9**
	Densenet201	pool3_pool	95,74	96,97	**96,35**
		pool4_pool	79,91	88,37	82,24
		avg_pool	68,00	81,44	69,51
	MobileNetV2	block_5_add	98,78	95,60	**97,11**
		block_14_add	84,29	93,90	88,20
CONV 6A	HOG		91,50	90,50	**91,00**
	Densenet201	pool3_pool	95,13	92,81	**93,95**
		pool4_pool	93,89	90,64	92,15
		avg_pool	61,03	71,65	62,38
	MobileNetV2	block_5_add	95,49	92,46	**93,93**
		block_14_add	88,54	84,01	85,97

Table 1 Precision, Recall and F-Scores of the system using different descriptors; retrieved from different CNN architectures and layers. The performance of these descriptors is compared to HOG, the descriptor used in the original system

3. Object detection for collision detection

3.1. People detection evaluation with a wearable camera

We present the results of 4 different people detector state of the art approaches over the detection part of the MOT2016/2017 dataset, i.e., MOT2017Det dataset [5]. Some examples from this dataset can be found in **Figure 2**. The selected approaches are the Aggregate Channel Features (ACF) [6], Deformable Parts Model (DPM) [7], Hierarchical detection of persons in groups (HDGP) [12] and Faster Regions with Convolutional Neural Network Features (FRCNN) [9]. **Figure 3** shows the detection performance and **Figure 4** shows the Precision-Recall Curves of each approach.

The ACF detector is a fast and effective sliding window detector (30 fps on a single core). It is an evolution of the Viola & Jones (VJ) detector but with an ~1000 fold decrease in false positives (at the same detection rate). ACF is best suited for quasi-rigid object detection (e.g. faces, pedestrians, cars).

The DPM detector is based on exhaustive search and a part-based model. It is a part-based adaptation of the original Histogram of Oriented Gradients detector (HOG) [8]. It proposes an object detection system based on mixtures of multiscale deformable part models where each deformable object part is 5nalyse5 as the original HOG detector [8].

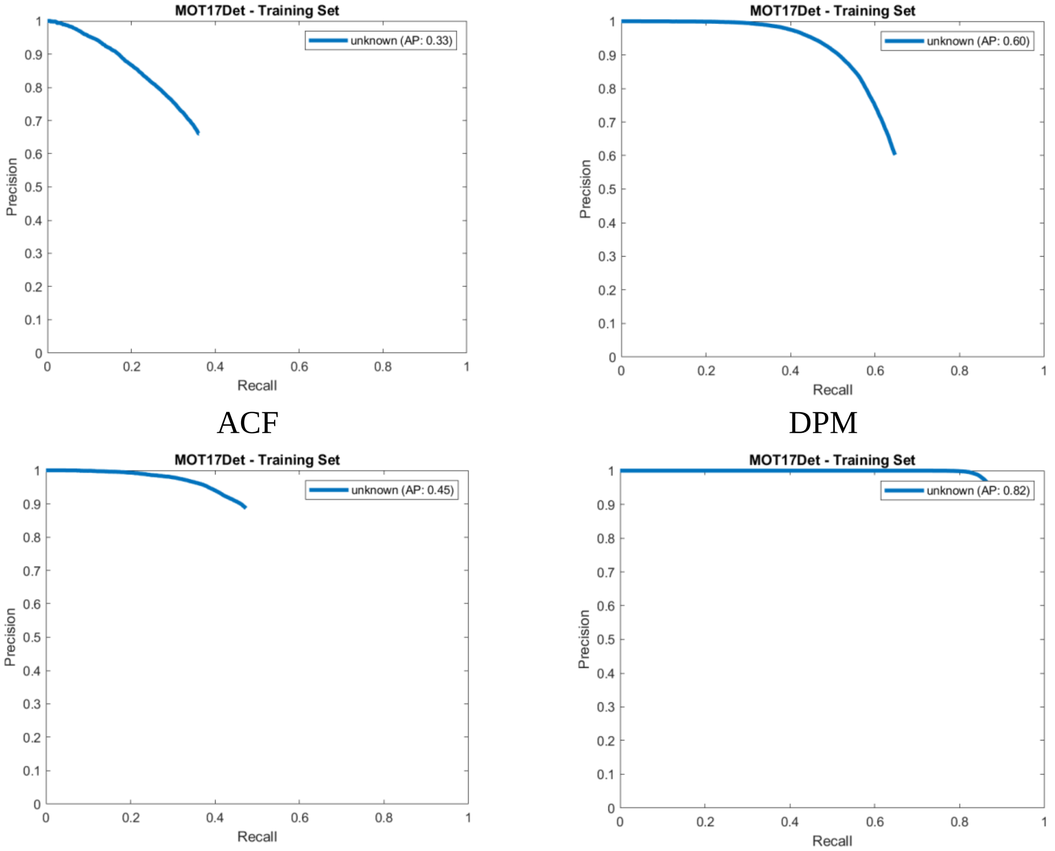
The HDGP is a DPM variation in order to detect pedestrians in scenarios with the presence of groups of persons. The HDGP detector propose a hierarchy of persons in groups, where the detection of the most visible person could help to detect the occluded ones, and a hierarchy of body parts, which main principle is to use the body parts with most useful information.

The FRCNN detector, which is a more efficient variation, mainly in terms of computational cost but also in performance, of the previous R-CNN [10] and FRCNN [11] detectors. The three variations have in common the combination of bottom-up region proposals with rich features computed by a convolutional neural network. The main difference of the FRCNN is the use of a Region Proposal Network (RPN) that enables nearly cost-free region proposals.

Figure 2. MOT16/17 visual examples.

Detector	Recall	Precision	FAR	TP	FP	FN	Av Precision
ACF	36.1	66.1	2.31	23937	12302	42456	0.3250
DPM	64.7	60.2	5.34	42979	28405	23414	0.6027
HDGP	47.3	88.6	0.76	31422	4038	34971	0.4464
FRCNN	87.9	93.8	0.72	58342	3844	8051	0.8180

Figure 3. MOT2017Det train sequences pedestrian detection evaluation.



ACF

DPM

HDGP

FRCNN

Figure 4. MOT2017Det train sequences pedestrian detection evaluation. Precision-Recall Curves.

3.2. People re-identification evaluation

We present the results of 11 different feature extraction schemes [42] that are commonly used in the re-id literature [13] over two traditional people re-identification datasets VIPeR [14] and Market150 [15]. The 11 different features are WHOS [16], GOG [17], AlexNet [20], ResNet18 [21], ResNet50 [21] , ResNet101 [21] , DenseNet201 [22] and InceptionResNetv2 [23].

Figure 5 shows the state of the art re-identification performance over VIPeR dataset and **Figure 6** over Market1501dataset.

In WHOS [50], is a feature descriptor based on weighted histograms of overlapping stripes. It is a discriminative and efficient descriptor of person appearance for re-identification based on coarse, striped pooling of local features. It exploits a simple yet effective center support kernel to approximately segment foreground from background

In GOG [48], an image is divided into horizontal strips and local patches in each strip are 7nalyse7 using a Gaussian distribution. Each strip is then regarded as a set of such Gaussian distributions, which is then summarized using a single Gaussian distribution.

In IDECaffeNet, IDE-ResNet, and IDE-VGGNet, we use the idea first presented in the DeepFace paper [18] and applied to reid by Zheng et al. [19], in which every person is treated as a separate class and a convolutional neural network is trained for a classification objective: AlexNet [20], ResNet18 [21] , ResNet50 [21] , ResNet101 [21] , DenseNet201 [22] and InceptionResNetv2 [23] are used respectively.

Feature	Rank1	Rank5	Rank10	Rank20	mAP-100
WHOS	27.52	56.34	70.57	83.77	34.35
GOG	32.66	65.25	77.97	88.48	40.03
AlexNet	11.82	30.02	42.77	57.69	16.80
ResNet18	6.09	18.86	29.19	44.45	10.10
ResNet50	12.96	34.32	48.05	64.27	18.61
ResNet101	15.06	36.60	49.64	65.27	20.70
DenseNet201	12.61	32.25	44.98	61.01	17.98
InceptionResNetv2	6.80	20.30	30.51	43.75	10.82

Figure 5. Viper pedestrian re-identification evaluation.

WHOS	Rank1	Rank5	Rank10	Rank20	mAP-100
GOG	39.96	62.00	71.14	79.99	18.83
AlexNet	42.84	64.10	73.19	80.40	21.65
ResNet18	21.44	41.66	51.60	60.57	8.34
ResNet50	11.07	26.51	35.30	46.35	4.04
ResNet101	18.29	37.98	48.28	57.84	7.14
DenseNet201	17.90	37.89	47.86	59.26	7.38
InceptionResNetv2	17.55	37.53	47.80	58.40	7.13
WHOS	8.14	20.07	27.49	36.16	2.72

Figure 6. Market1501 pedestrian re-identification evaluation.

3.3. Car re-identification evaluation

Following the same evaluation than with person re-identification, we present the results of 11 different feature extraction schemes that are commonly used in the re-id literature [13] over one car re-identification dataset CityFlow-ReID [24] The 11 different features are WHOS [16], GOG [17], AlexNet [20], ResNet18 [21], ResNet50 [21] , ResNet101 [21] , DenseNet201 [22] and InceptionResNetv2 [23]. **Figure 7** shows the state of the art re-identification performance over CityFlow-ReID dataset

WHOS	Rank1	Rank5	Rank10	Rank20	mAP-100
GOG	21.82	35.72	43.76	55.16	6.10
AlexNet	17.70	32.57	41.37	49.95	5.78
ResNet18	22.26	39.20	46.91	57.11	6.91
ResNet50	17.48	35.07	42.89	53.75	5.54
ResNet101	25.73	44.52	53.42	62.98	8.90
DenseNet201	25.73	42.56	51.14	60.26	8.72
InceptionResNetv2	28.99	46.58	54.07	63.41	10.03
WHOS	21.93	35.94	44.63	55.16	6.10

Figure 7. CityFlow-ReID car re-identification evaluation.

3.4. People detection in presence of groups

In this work [43] we address one of the most typical problems of people detection in presence of groups of people: in this kind of scenarios, traditional people detectors have difficulties dealing with several occlusions. In order to deal with this problem, we propose the use of two different hierarchies. The first one consists of a hierarchy of people, i.e., the use of the detections of different people belonging to a group in order to refine the individual's detections. The second one consists of a hierarchy of parts [45], i.e., the use of different combinations of body parts in order to refine the final detection (see **Figure 8**).

In the last years, the use of the Artificial Intelligence, in concrete, Convolutional Neuronal Networks (CNNs) for the treatment of images, has managed to improve these algorithms, which usually used traditional techniques such as gradient descent. The results are very encouraging in all areas of detection such an objects, people, and another elements in images.

This is why we also propose in this work to improve the original approach [45] with an algorithm based on Deformable Parts Models (DPMs) and CNNs [44], with the aim of adding those cases in which we find occlusion of people by others, using a model that not only focuses at the main person but also analyzes its closest environment in search of more persons that may be occluded and in which we only see, for example, the right half of the body.

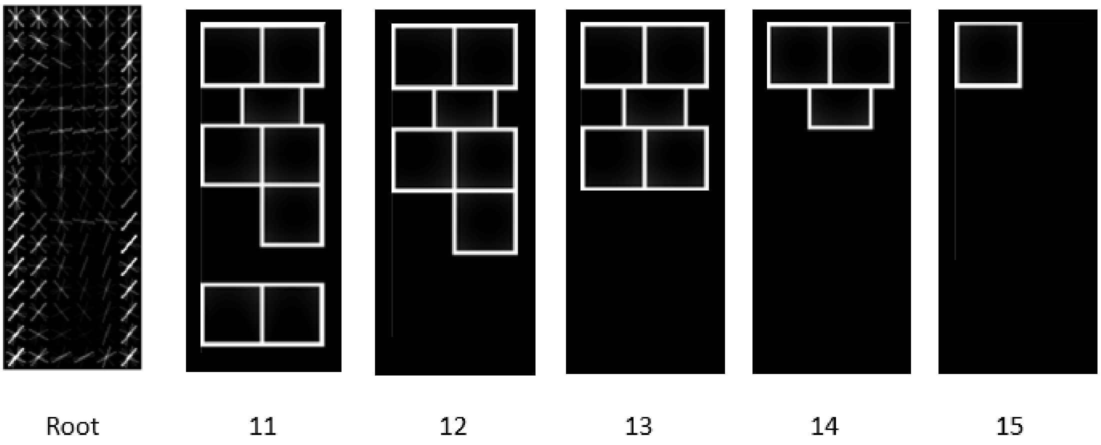


Figure 8. Different combinations of body parts in order to refine the final detection.

4. Scene categorization for lifelogging

4.1. Comparative performance of scene recognition methods.

The aim of this study is to assess the performance of scene recognition methods and to explore their capabilities and robustness to challenges. To this aim, we here report quantitative performances of five of the most used deep learning methods based on convolutional neural networks for the task of automatic scene recognition. The evaluated architectures are: AlexNet

[20], VGG16 [25], Densenet161 [22], ResNet18 and ResNet50 [21]. The dataset used for evaluation is the validation set of the Places365 [26] (see **Figure 9**).

To further assess the performance of these methods, we explore their diversity by accounting for the scene classes to which any two networks return the same results, disregarding their correctness (see confusion matrix of **Figure 10**). We also measure their operation limits by inspecting the scene categories to which the architectures return less (see **Figure 11**) and more (see **Figure 12**) coincidences.

Likewise, the robustness of the analysed solutions to image noise is also evaluated [27]. Finally, the responses of the networks to images that represent untrained scene categories are studied. (see example results in **Figure 13** and **Figure 14**).

Experimental results suggest that:

1. Performance for scene recognition is below coeval tasks such as object detection.
2. DenseNet161, the deeper network, is the top performing architecture. However, performance does not increase at the same rhythm as architectures' complexity.
3. There are scene classes that entail a higher complexity than others, mainly due to the presence of semantically similar classes in the dataset.
4. Human-interpretable concepts seem to be learning, as the networks' responses to previously unobserved classes somehow agrees with human interpretation.
5. There is room for improvement of both the performance and the interpretability of deep learning solutions for scene recognition.

Architecture	#Layers	#correct images	Rank1	Rank5	Rank10
AlexNet	8	15.622	45,33%	67,26%	85,31%
VGG16	16	16.794	46,01%	68,47%	86,81%
ResNet18	18	19.874	54,45%	76,97%	91,91%
ResNet50	50	20.272	**55,54%**	**78,22%**	**92,89%**
DenseNet161	161	**20.477**	53,80%	76,55%	92,04%

Figure 9. Evaluation of deep learning architectures on Places365. Best result in **bold**.

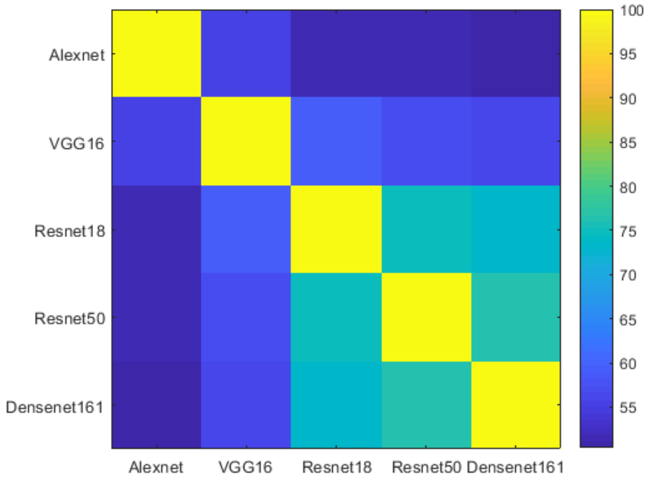


Figure 10. Confusion matrix of deep learning architectures on Places365.

	Alexnet	VGG16	Resnet18	Resnet50	Densenet161
Alexnet	-	*science_museum* *(29)*	*music_studio* *(19)*	*music_studio* *(16)*	*science_museum* *(15)*
VGG16	*science_museum* *(29)*	-	*pub/indoor* *(17)*	*pub/indoor* *(16)*	*pub/indoor* *(21)*
Resnet18	*music_studio* *(19)*	*pub/indoor* *(17)*	-	*11nalyse* *(53)*	*11nalyse_loft* *(51)*
Resnet50	*music_studio* *(16)*	*pub/indoor* *(16)*	*11nalyse* *(53)*	-	*valley* *(56)*
Densenet161	*science_museum* *(15)*	*pub/indoor* *(21)*	*11nalyse_loft* *(51)*	*Valley* *(56)*	-

Figure 11. Scene categories with a lower number of inter-network coincidences. Number of coincidences (out of 100) is indicated in brackets.

	Alexnet	VGG16	Resnet18	Resnet50	Densenet161
Alexnet	-	*volleyball_court/ outdoor (89)*	*car_interior, volleyball_court/ outdoor (86)*	*car_interior (87)*	*volleyball_court/ outdoor (87)*
VGG16	*volleyball_court/ outdoor (89)*	-	*volleyball_court/ outdoor (92)*	*wind_farm (89)*	*volleyball_court/ outdoor (91)*
Resnet18	*car_interior, volleyball_court/ outdoor (86)*	*volleyball_court/ outdoor (92)*	-	*cockpit (98)*	*bowling_alley (95)*
Resnet50	*car_interior (87)*	*wind_farm (89)*	*cockpit (98)*	-	*arena/hockey (97)*
Densenet161	*volleyball_court/ outdoor (87)*	*volleyball_court/ outdoor (91)*	*bowling_alley (95)*	*arena/hockey (97)*	-

Figure 12. Scene categories with a higher number of inter-network coincidences. Number of coincidences (out of 100) is indicated in brackets.

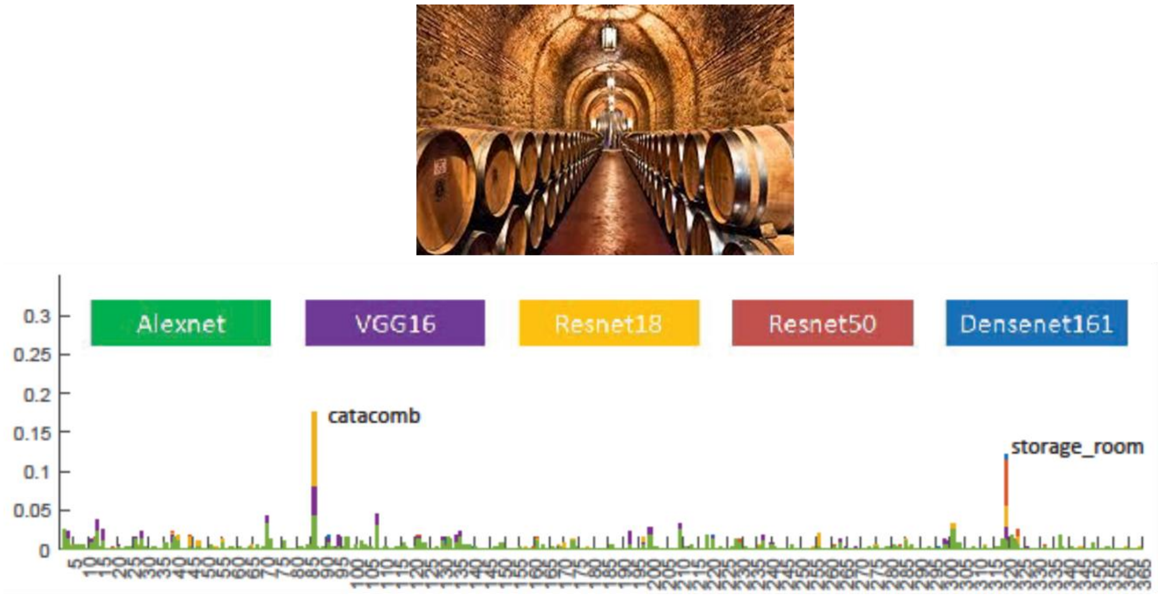


Figure 13. Networks predictions for untrained class *cellar*.

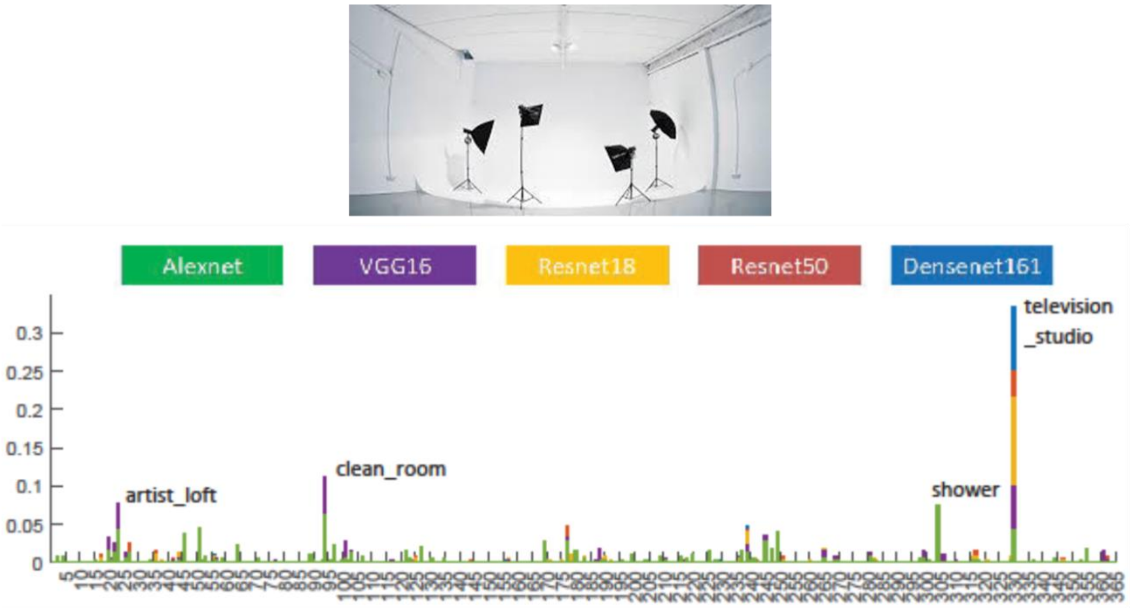


Figure 14. Networks predictions for untrained class *photograph studio*.

4.2. Scene recognition for lifelogging.

4.2.1. Performance of scene recognition methods for lifelogging videos

The aim of this study is to assess the effectiveness of previously studied scene recognition methods for the analysis of lifelogging videos. To this aim, we have created a dataset that is made up of 450 videos (up to 263607 frames) and is arranged into 365 classes following the class definitions in Places365 dataset [26]. Details of the dataset can be found at [35].

The protocol for evaluation is as follows:

1. First, we extract scene class likelihoods for each frame of each video by using scene-trained convolutional neural networks—AlexNet [20], ResNet18, ResNet50 [21] and Densenet161 [22]—.

2. Then, we extract global performance metrics to classify the whole video to a scene class. We explore the application of two simple global measures: the mean (**μ**) and the median (**M**) along the whole video of the scores obtained for each image.

Performances are reported in **Figure 15** that also includes—for comparison—individual performances (**I**), i.e. those resulting for accounting the number of correctly recognised frames out of the total number of frames per scene class.

Architecture	RANK 1 (%)			RANK 5 (%)			RANK 10 (%)		
	I	µ	M	I	µ	M	I	µ	M
AlexNet	20.365	28.914	24.578	43.102	51.561	49.408	53.477	62.926	61.019
ResNet18	27.445	35.372	31.262	52.526	60.407	56.775	61.417	69.425	67.936
ResNet50	29.127	36.127	34.485	55.120	**64.159**	60.765	63.731	70.039	69.393
DenseNet161	29.783	**36.605**	34.560	55.430	63.645	60.543	63.882	**70.095**	67.974

Figure 15. Performance of scene recognition for lifelogging videos. Best result in **bold**.

Experimental comparison is completed with a Graphical User Interface (GUI) that enables the observation of per frame results. **Figure 16** includes a snapshot of the GUI:

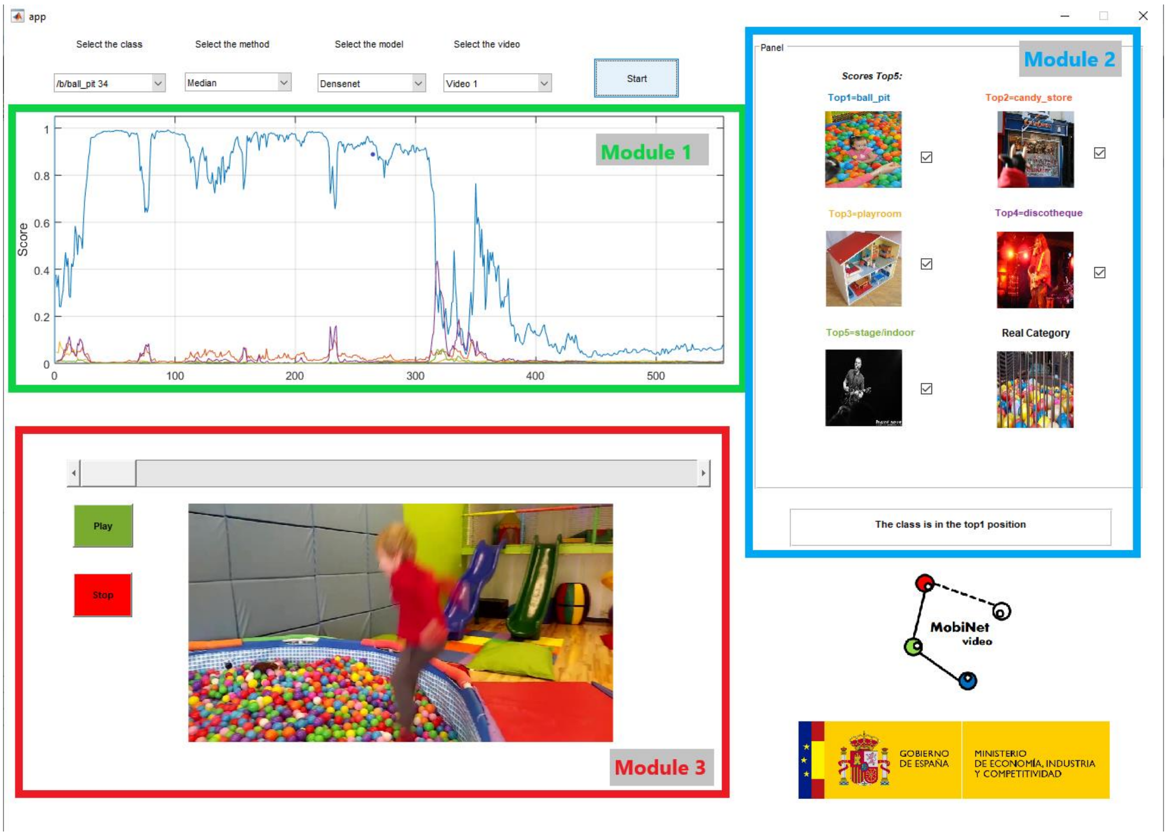
Module 1 -includes a graphic visualization of the temporal distribution of the scores of the five best categories throughout the video. The module allows to select the global metric, the desired scene class to be explored and the convolutional neural network whose results are to be observed.

Module 2 -includes example images of the five top categories obtained by the selected network, on the selected sequence according to the selected global metric. For each one of these images, the GUI includes a checkbox, that can be selected/deselected to include graphical results for the associated scene class in Module 1. For comparison, we also include in this module (bottom right corner) an example image of the ground-truth class.

Module 3 -includes the video under analysis, it allows to play or stop the video at will and to move freely at frame level.

In the example, the ball pit class (blue line) is dominant along the video, but the score for this class drops at some frames. This situation is recurrent along the dataset. Overall, experimental results suggest that scene recognition in lifelogging videos is challenging due to the temporal instability of scene predictions and to the heterogeneity of lifelogging videos (i.e., several scene classes may coexist inside a single video).

The GUI is available at: http://www-vpu.eps.uam.es/webvpu/en/recursos-publicos/software/.



Select the class: /b/ball_pit 34

Select the method: Median

Select the model: Densenet

Select the video: Video 1

The class is in the top1 position

Figure 16. Graphical User Interface for assessing scene recognition for lifelogging scenarios. See text for modules' description.

4.2.2. Interpretability of deep learning networks

The aim of this study is to gather information on the internal processes carried out for recognition inside a scene recognition method based on convolutional neural network. To this aim, and due to the complexity of creating a visualization tool from scratch, we focus on the adaptation of an existing visualization tool. Specifically, we target to enhance the scalability and modularity of an existing tool, maximizing the diversity of the networks that can be observed.

In the design of a training strategy, one of the main challenges is the difficulty in understanding what is happening inside the convolutional neural network. This difficulty can lead to the interpretation of the network as a *black box*; hence, hindering the ability to optimize it for a given task. The use of a visualization tool promises to enable the design of dynamic training strategies suited for scene recognition on large heterogeneous lifelogging datasets, by allowing an orderly access to information for all the layers of a given network.

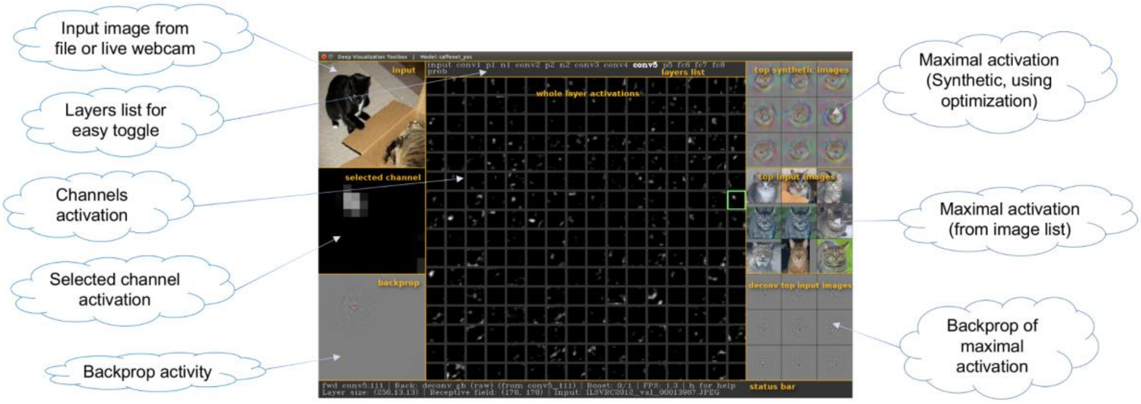

Input image from file or live webcam
Layers list for easy toggle
Channels activation
Selected channel activation
Backprop activity
Maximal activation (Synthetic, using optimization)
Maximal activation (from image list)
Backprop of maximal activation
input
layers list
whole layer activations
selected channel
backprop
top synthetic images
top input images
deconv top input images
status bar

Figure 17. Functionalities of the DeepVis Toolbox interface.

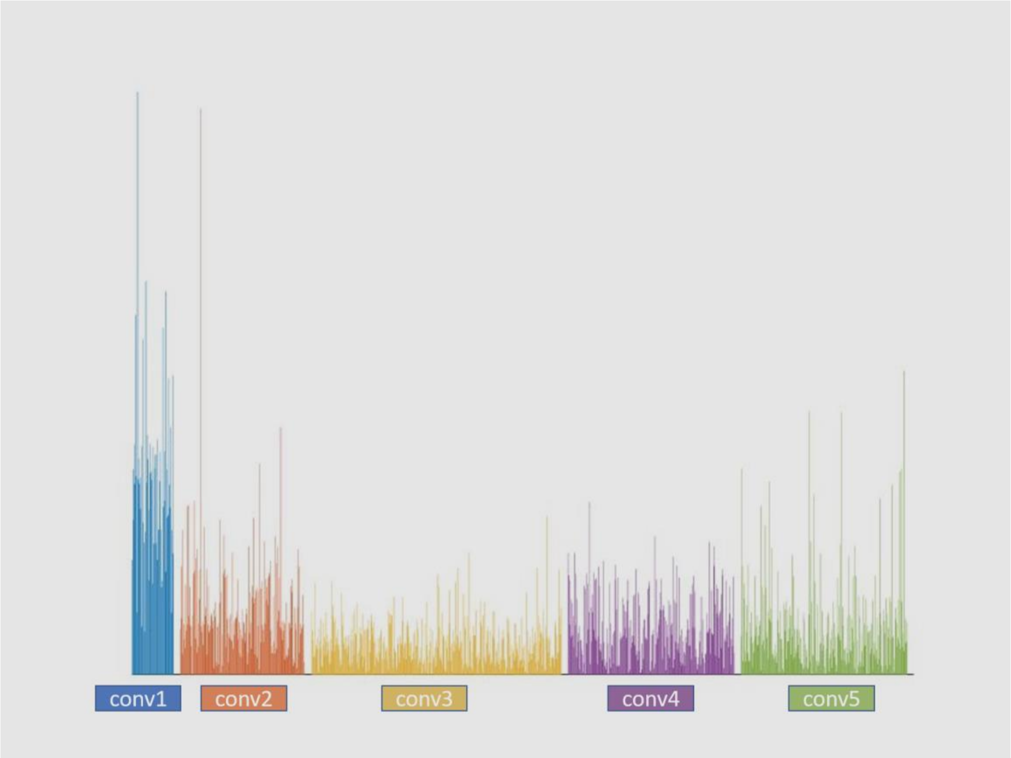


Figure 18. Entropy of neurons’ activations for AlexNet trained for scene recognition.

We adapted The DeepVis Toolbox [28][29]—with functionalities depicted in **Figure 17**. DeepVis provides information on the activation patterns, the maximal activation image, the receptive field and the canonical filter nature for every neuron in the convolutional neural network. The main problem of DeepVis is its scalability. DeepVis has been developed in Caffe [30] and provides functionalities just for Caffe-like architectures. Whereas Caffe was a widely used platform for deep learning, currently, the focus is on alternative development environments such as Pytorch [31]. In this vein, we propose to establish an adaptation protocol such that the visualization of non-Caffe convolutional neural networks can be also carried out via DeepVis.

To this aim, for a given convolutional neural network in Pytorch, we start by using a method to transform for Python to Caffe [32].

We are currently working in quantifying these observations and in designing strategies to maximize the learning capabilities of a given architecture. An example of the potential advantages enabled by the visualization tool is included in **Figure 18**. The Figure represents the entropy of AlexNet neurons' activations for all the images of a scene recognition dataset. Low entropy is an indicator of homogeneous stable activations—or continuous inhibition—, high entropy is an indicator of heterogeneous diverse activation. We hypothesize that the representation capacity of low entropy neurons is underused and propose to design a scheme to convey dynamic learning rates for underused neurons. In the example depicted in in **Figure 18**, the intermediate layer conv3 appears to be less used than generic (shallower) and specialized (deeper) layers.

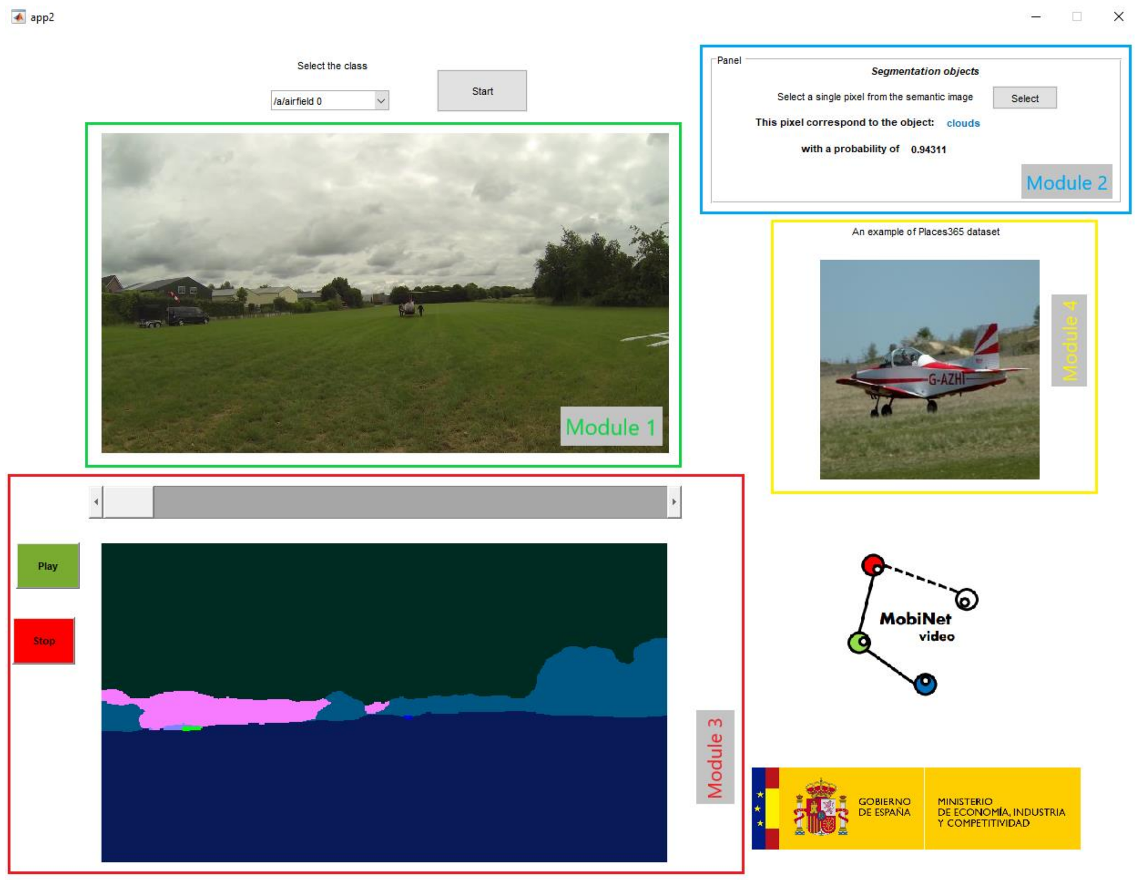
The visualization tool together with the adaptation protocol is available at: http://www-vpu.eps.uam.es/webvpu/en/recursos-publicos/software/.

4.3. Semantic segmentation for lifelogging.

Recent studies on the semantic interpretability of convolutional neural networks, suggest that the learning of scenes is inherent to the learning of the objects they include [33][34]. Object *detectors* somehow act as latent-variables in hidden-units within networks trained to recognize scenes. These *detectors* are learnt without constraining the networks to decompose the scene recognition problem [33]. Under this premise, this line of research is focused on evaluating the benefits of explicitly including object information in the scene recognition process. To this aim, we start by studying the performance of semantic segmentation methods on lifelogging scenarios. Alternatively, the performance of object detectors in these scenarios is evaluated in section 4.4.

The performance of semantic segmentation methods has boosted since the advent of deep learning solutions. The highly complex task of labelling a pixel as an instance of a given class has been substantially simplified using encoder-decoder convolutional neural networks. However, performance is still constrained to specific scenarios and sets of classes, which is one of the pending challenges of this technique.

We aim to evaluate the performance of existing semantic segmentation methods on lifelogging videos. However, to our knowledge there are not public-available lifelogging datasets annotated for semantic segmentation. To cope with this issue, we designed a GUI to (at less qualitatively) can be used assess the performance at pixel level for the videos in the lifelogging dataset described in [35].



Select the class: /a/airfield 0

Panel

Segmentation objects

Select a single pixel from the semantic image

This pixel correspond to the object: clouds

with a probability of 0.94311

An example of Places365 dataset

Figure 19. Graphical User Interface for assessing semantic segmentation recognition for lifelogging scenarios. See text for modules' description.

The GUI is depicted in **Figure 19**:

Module 1 -includes the video under analysis and enables the selection of any pixel in the current frame whose information is depicted at Module 2.

Module 2 -includes the information of the selected pixel, including top semantic class and associated likelihood.

Module 3 -includes the semantic information of the video under analysis, it allows to play or stop the video at will and to move freely at frame level.

Module 4 -includes an example image from the (Places365 dataset) of the scene class associated to the video under analysis.

The GUI is available at: http://www-vpu.eps.uam.es/webvpu/en/recursos-publicos/software/ and allows to qualitative asses that the semantic segmentation is temporally unstable at pixel level. However, top semantic classes remain somehow stable for scene classes.

4.4. Object detection for lifelogging.

The work in [36] performs an initial evaluation of state-of-the-art object detectors in Google Street View images obtained with the tool described in Section 3.2 deliverable D1.3 [35] of the MobiNet video project. YOLOv3 [37], trained with the COCO database [38], has been used to perform detection of objects that are common in urban scenes. Some examples of detection results for different objects in different locations are shown in Figure 20.

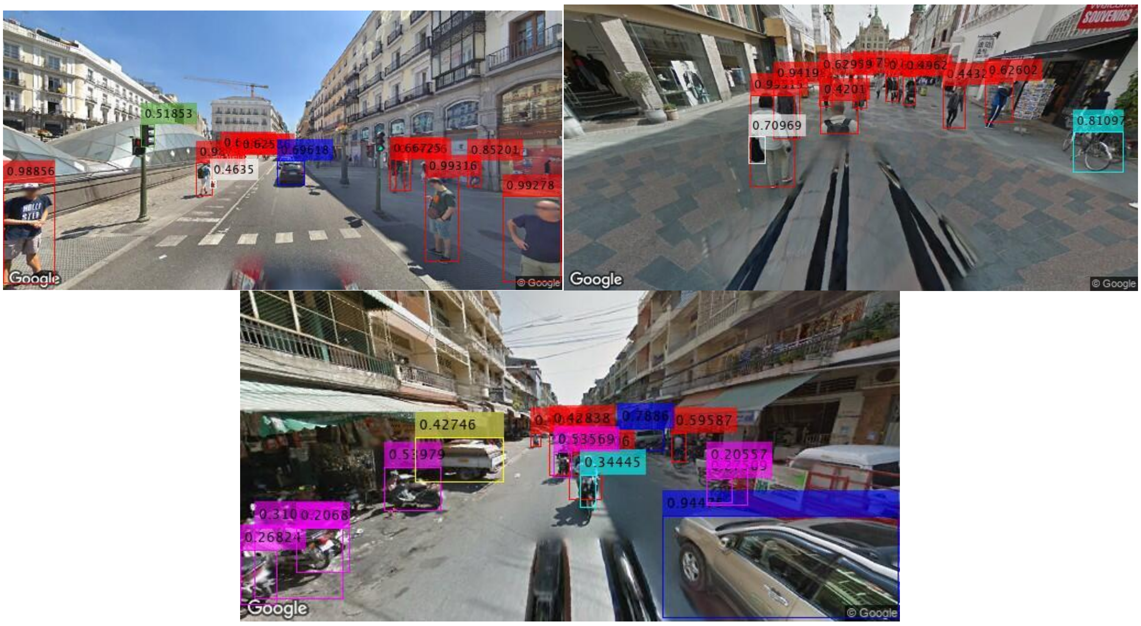
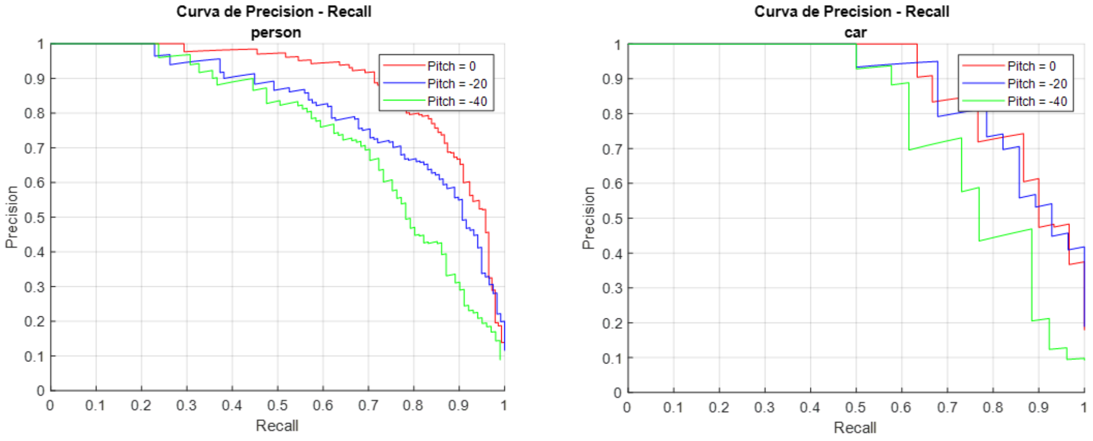


Figure 20 Examples of object detection in Google Street View images. Colors of bounding boxes represent different object classes; red: person, dark blue: car, light blue: bicycle.

The evaluation of the performance of object detection in Google Street View images has been primarily oriented to investigate the effect of geometric distortion in the appearance of objects caused by images with a large field of view (this effect can be observed in the car at the bottom of the third image in Figure 20). An example of the results can be found in Figure 21. The results obtained in this work are preliminary and obtained using a limited dataset, but suggest that the geometric distortion that is present in these images (caused by certain parameters in the image retrieval) affects the performance of state-of-the-art object detectors on these types of images. This effect should be investigated further during the lifetime of the project.



(b)

(a)

Figure 21 Precision-Recall curves for object detection in Google Street View images: classes person and car. Curves for configurations with different geometric distortion levels. Pitch= 0 and pitch=-40 corresponds to the configurations with lower and higher geometric distortions respectively.

5. Multi-target tracking for UAV monitoring

5.1. Visdrone challenge 2018

In 2018 a new challenge, the VisDrone-VDT2018 ("Vision Meets Drone Video Object Detection and Tracking") challenge appeared in which all the sequences consist in drone or UAVs videos with multiple objects. VisDrone-VDT2018 [39] is a large-scale video object detection and tracking dataset, including 79 video clips with approximate 1.5 million annotated bounding boxes in 33,366 frames. Some other useful annotations, such as object category, occlusion, and truncation ratios, are also provided for better data usage. The dataset is collected with several drones, in various scenarios, which are taken at different locations, but share similar environments and attributes. **Figure 22** and **Table 2** summarize the dataset.

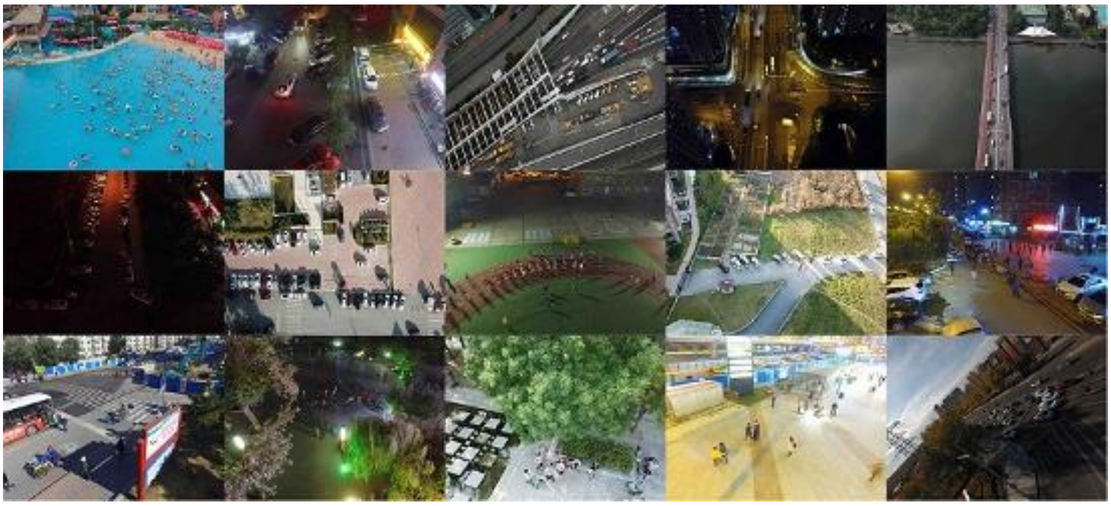
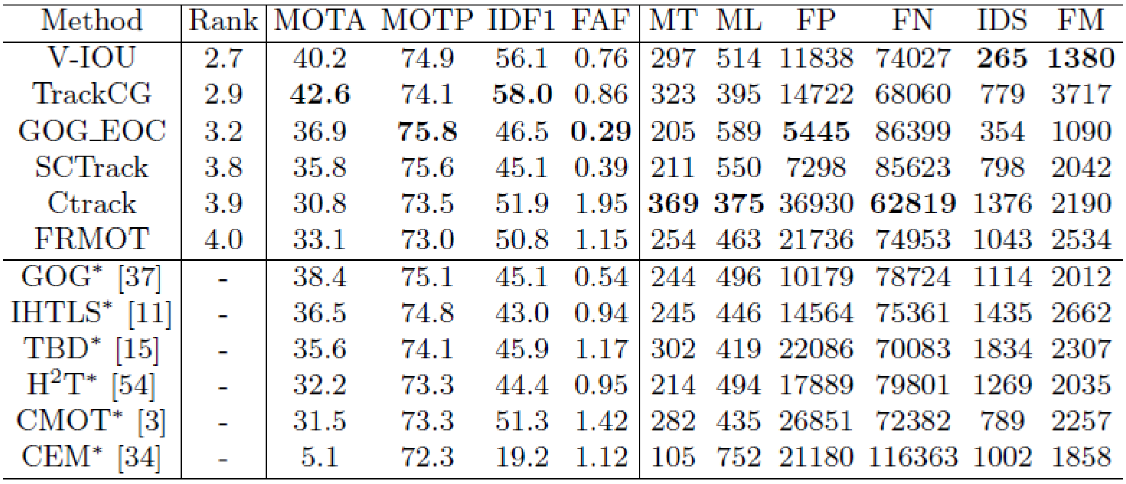


Figure 22. An overview of the VisDrone dataset.

Number of snippets			
Dataset	Training	Validation	Test-challenge
Multiple object tracking	56 clips 24,201 frames	7 clips 2,819 frames	16 clips 6,333 frames

Table 2. VisDrone MOT challenge dataset 2019

The results obtained in the competition are as indicated in the following **Table 3**.



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^2T^* [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

Table 3 Multi-object tracking results with prior object detection in each frame on the VisDrone-VDT2018 testing set. The submitted algorithms are ranked based on the average rank of the ten metrics. (* indicates that the tracking algorithm is submitted by the committee)

We can observe that the winner was V-IOU. According to [39], the algorithm V-IOU is based on the IOU tracker 33[40][41] which associates detections to tracks solely by their spatial overlap (Intersection-over-Union) in consecutive frames. We 22nalyse the tracker in the next subsection.

5.2. Analysis of IOU and V-IOU trackers

5.2.1. Tracker IOU: description

The main objective of the IOU 33[40] tracker is the fast multiple object tracking based on the improved object detectors. This is possible due to recent advances in the detection domain including CNN-based and traditional approaches.

The main idea is that improvements in the object detection tasks allow to simplify the tracking task. By utilizing the tracking-by-detection approach, the tracker itself can be thought of as a simple filtering procedure on the detection level. The proposed in the paper tracker uses the state-of-the-art object detector, the output of which is filtered according to the following rules. The algorithm of the IOU (Intersection Over Union) tracker is based on two assumptions:

- The detector produces a detection per frame for every object to be tracked, i.e. there are none or only a few ”gaps” in the detections (otherwise fragmentation and ID-switches rates will grow)

- The detections of an object in consecutive frames have an unmistakably high overlap (IOU) (therefore the video sequence must have high fps or contain only slowly moving objects)

The algorithm itself can be summarised as follows: a track continues by associating the detection with the highest IOU (see eq. 1) to the last detection in the previous frame if a certain threshold σIOU is met. All detections not assigned to an existing track will start a new one. All tracks without an assigned detection will end.

$$IOU(a,b) = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \qquad (1)$$

The performance is further improved by filtering out all tracks with a length shorter than *tmin* and the ones without at least one detection with a score above σ*h*.

5.2.2. Tracker IOU: Conclusions and experiments

- The presented IOU tracker outperforms the state-of-the-art at only a fraction of the complexity and computational cost without any analysis of the visual information on the *DETRAC* dataset, however it shows worse performance on the *MOT16/MOT17 dataset*

- The proposed tracker does not handle missing detections in any way, therefore false negatives of the detector lead to high fragmentation and ID-switches

- The input video must have high fps or capture objects that move very slowly

- The tracker demonstrates performance shown in the **Table 4**, on the *MOT17 Dataset* on the machine with following specifications

    - CPU: Intel Core i7-7700 HQ @ 2.80 GHz, 4 physical cores, 8 logical cores
    - GPU: NVIDIA GeForce GTX 1050, 4GB, GDDR5
    - RAM: SODIMM Samsung [M471A1K43CB1-CRC], 16 GB, DDR4, 2400 MHz
    - OS: Windows 10 Home edition, 64-bit
    - Python version: 3.7.1

	Detector	
	FRCNN	SDP
MOT17 **Train** Sequences	Average: **5222** fps Min: 891 fps Max: 9887 fps	Average: **4939** fps Min: 497 fps Max: 10739 fps
MOT17 **Test** Sequences	Average: **4571** fps Min: 346 fps Max: 12273 fps	Average: **3188** fps Min: 223 fps Max: 8036 fps

Table 4. Performance of the IOU-tracker

- On the same machine and *CARPK dataset* the tracker reaches 108-110 fps. For this experiment there were no FRCNN/SDP or other detections available on the Internet, so instead of them, the ground truth was used. As can be seen from the **Figure 23**, the tracker does not yield good results on this dataset even with the perfect detections, because FPS of the video is very low

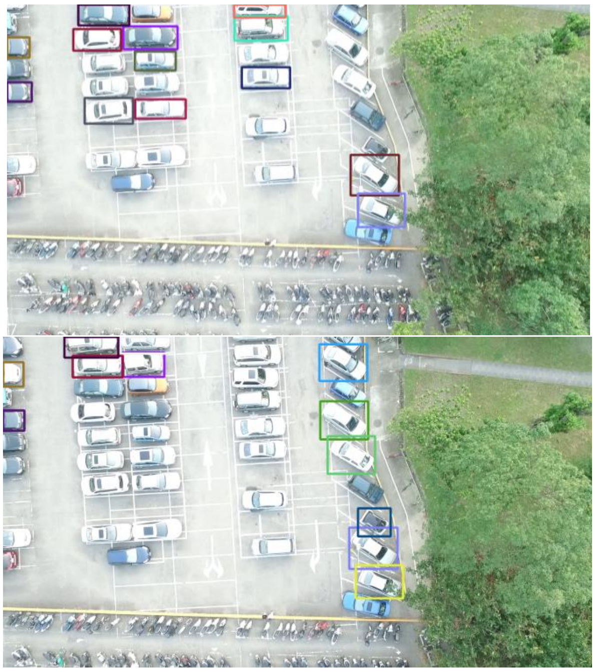


Figure 23 IOU-tracker on the CARPK dataset with ground-truth as the detections. Because of low FPS of video footage, fragmentation and the number ID-switches are very high, therefore this tracker is not applicable to such datasets.

5.2.3. Tracker V-IOU: description

The method is further improved by visual tracking to continue a track if no detection is available by the method V-IOU [41]. If a valid detection can be associated again, visual tracking is stopped and the tracker returns to the original IOU tracker functionality. Otherwise, the visual tracking is aborted after ttl frames. For each new track, visual tracking is performed backwards for a maximum of ttl previous frames or until the track can be merged with a finished one if the IOU criteria of [40] is satisfied. This extension is made to efficiently reduce the high amount of fragmentation of the tracks produced by the original IOU tracker. V-IOU can be used in association with a wide range of visual single-object trackers. Please refer to [41] for further details.

Using visual information

The modification to the IOU-tracker is designed to reduce the high amount of ID switches and the high rate of fragmentation of the resulting tracks of the baseline method. This is achieved by incorporating a visual single-object tracker into the tracking framework to compensate for missing object detections. When no detection is available for track association, the IOU-tracker falls back to visual single-object tracking performed in two directions to help to merge discontinued tracks. Two directional tracking is justified by the assumption that it lowers the risk of visual tracker losing the target.

The conclusions for this extension are as follows:

- The presented V-IOU-tracker reduces the number of ID switches and fragmentations compared to the original IOU-tracker
- Although, the speed of such modification is significantly lower than that of the original tracker, it is still very high even for high-definition video footage
- The proposed tracker outperforms the state-of-the-art on the DETRAC and VisDrone datasets
- **There is no code available for this tracker**

Using Kalman Filter

The idea is to use Kalman Filter to skip frames in order to increase speed and deal with false negatives of the detector. In the real-world applications, few missing detections cause a high number of ID switches and fragmentations which degrades the quality of the tracks significantly.

The Kalman filter’s capability of making predictions allows to skip frames while still keeping track of the object. Skipping frames in a tracking-by-detection task means the detector will process significantly less frames. The Kalman-IOU Tracker, when used with the EB detector and configured to skip 2/3 of the frames, can outperform the original IOU Tracker on the DETRAC-Train dataset.

The conclusions for this extension are as follows:

- The presented KIOU-tracker reduces the number of ID switches and fragmentations compared to the original IOU-tracker
- To evaluate the performance of the Kalman IOU tracker and compare it with others, the authors present the results tables which presumably reflect the superiority of the K-IOU tracker. Based on the tables, the tracker performs better on both DETRAC Train and Test datasets. However, in reality, references to the original IOU paper show that the authors of K-IOU tracker did not include the best versions of IOU-tracker that achieve the best performance in their comparison report. Neither did they include any comparison with the V-IOU tracker, nor the speed of the K-IOU tracker.
- A complete comparison is presented in the **Table 5.** Comparison of IOU-based trackers performance on the DETRAC Test dataset.

Tracker	Detector	PR-MOTA	PR-MOTP	PR-IDs	PR-FM	Speed
IOU	R-CNN	16.0%	38.3%	5029.4	5795.7	100,840 fps
IOU	EB	19.4%	28.9%	2311.3	2445.9	6,902 fps
KIOU	**EB**	**21.1%**	**28.6%**	**462.2**	**712.1**	**-**
IOU	**Mask R-CNN**	**30.7%**	**37.0%**	**668.0**	**733.6**	**14,956 fps**
V-IOU	**Mask R-CNN**	**30.7%**	**37.0%**	**162.6**	**286.2**	**359.18 fps**

Table 5. Comparison of IOU-based trackers performance on the DETRAC Test dataset.

- The tracker demonstrates performance shown in the Table 3, on the MOT17 Dataset on the machine with following specifications
    - CPU: Intel Core i7-7700 HQ @ 2.80 GHz, 4 physical cores, 8 logical cores
    - GPU: NVIDIA GeForce GTX 1050, 4GB, GDDR5
    - RAM: SODIMM Samsung [M471A1K43CB1-CRC], 16 GB, DDR4, 2400 MHz
    - OS: Windows 10 Home edition, 64-bit
    - Python version: 3.7.1

As can be seen from the **Table 6.** Performance of the K-IOU-tracker, on average the K-IOU-tracker shows real-time performance, but sometimes it drops below 10 fps.

	Detector	
	FRCNN	SDP
MOT17 **Train** Sequences	Average: **50** fps Min: 20 fps Max: 294 fps	Average: **76** fps Min: 13 fps Max: 152 fps
MOT17 **Test** Sequences	Average: **126** fps Min: 9 fps Max: 291 fps	Average: **57** fps Min: 6 fps Max: 126 fps

Table 6. Performance of the K-IOU-tracker

- On the same machine and CARPK dataset the tracker reaches **8 fps** skipping ⅔ of the frames, as opposed to the original tracker that reaches 110 fps. For this experiment there were no FRCNN/SDP or other detections available on the Internet, so instead of them, the ground truth was used.

- There is no paper for this tracker

The code is available on GitHub

6. Conclusions

This deliverable has described the state of the art in the areas of

- People tracking for active vision
- Object detection for collision detection
- Scene categorization for lifelogging
- Multi-target tracking for UAV monitoring.

As well implementing and evaluating, on public datasets, different algorithms, that will be used for comparisons with the algorithms to be developed within PW3 and WP4.

References

[1] 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.

[2] Development of an algorithm for people detection using omnidirectional cameras, Lorena García de Lucas (advisor: Pablo Carballeira López), Proyecto Fin de Carrera (Master Thesis), Ingeniería de Telecomunicación, Escuela Técnica Superior de Ingenieros de Telecomunicación, Univ. Politécnica de Madrid, Jan. 2016.

[3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". *arXiv preprint arXiv:1704.04861*, Apr. 2017.

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

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, Places: A 10 million image database for scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, no. 99, pp. 1–1, 2017.

[27] Raúl García Jiménez, Mejora del rendimiento de redes convolucionales entrenadas para el reconocimiento de escena mediante el uso de información sobre los objetos comunes a éstas. Tutor: Marcos Escudero Viñolo. Escuela Politécnica Superior. Junio 2018.

[28] Jason Yosinski and Je_ Clune and Anh Nguyen and Thomas Fuchs and Hod Lipson, "Understanding Neural Networks Through Deep Visualization"(2015), https://github.com/yosinski/deep-visualization-toolbox.

[29] Arikpoz, "deep-visualization-toolbox modified". https://github.com/arikpoz/deep-visualization-toolbox.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding”. In

Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678), ACM 2014.
[31] Ketkar, N. (2017). Introduction to pytorch. In *Deep learning with python* (pp. 195-208). Apress, Berkeley, CA.
[32] https://github.com/xxradon/PytorchToCaffe.
[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene cnns,” arXiv preprint arXiv:1412.6856, 2014.
[34] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6541–6549.
[35] D1.3 v1 Evaluation datasets. TEC2017-88169-R MobiNetVideo (2018-2020): Visual Analysis for Practical Deployment of Cooperative Mobile Camera Networks. July 2019.
[36] Detección de objetos en imágenes urbanas de Google Street View (Object detection in Google Street View urban images), Paula Guerra Toni, (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.
[37] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in IEEE Conference on Computer Vision and Pattern Recognition , July 2017, pp.6517–6525.
[38] Common Objects in COntext (COCO) http://cocodataset.org (accessed Jul. 2019).
[39] Zhu, P., Wen, L., Bian, X., Ling, H., & Hu, Q. (2018). Vision meets drones: a challenge. arXiv preprint arXiv:1804.07437.
[40] Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: IEEE International Conference on Advanced Video and Signal-Based Surveillance. pp. 1–6 (2017)
[41] Bochinski, E., Senst, T., Sikora, T.: Extending iou based multi-object tracking by visual information. In: AVSS. IEEE (2018)
[42] 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.
[43] Detección jerárquica de grupos de personas con CNNs, Antonio Campoy Cordero, (Tutor: Álvaro García Martín), Trabajo Fin de Grado, Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, Univ. Autónoma de Madrid, Jul. 2019.
[44] R. Girshick, F. Iandola, T. Darrell, J. Malik. Deformable Part Models are Convolutional Neural Networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 437-446.
[45] Á. García-Martín, R. Sánchez-Matilla, J. M. Martínez, "Hierarchical detection of persons in groups", Signal Image and Video Processing, Volume 11, Issue 7, October 2017, pp 1181–1188.