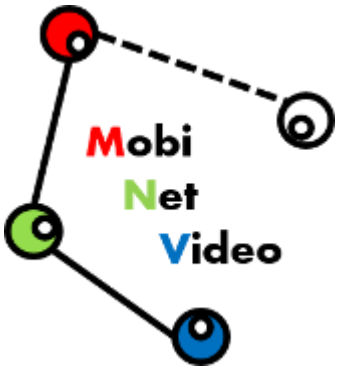


TEC2017-88169-R MobiNetVideo (2018-2020)

Visual Analysis for Practical Deployment of Cooperative Mobile Camera Networks

D3 v1

Technologies for Mobile Camera Networks

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 C. San Miguel Avedillo	Juancarlos.sanmiguel@uam.es
Elena Luna García	Elena.luna@uam.es
Alejandro Lopez Cifuentes	alejandro.lopezc@uam.es

HISTORY

Version	Date	Editor	Description
0.1	31/07/2019	jsa	Initial draft version
0.2	23/08/2019	agm	People/Car re-identification approach
0.3	04/05/2019	Elg	Cooperative detection and tracking
0.4	05/05/2019	jsa	Review of Cooperative detection and tracking
0.5	13/09/2019	alc	Scene Recognition and Semantic Segmentation
0.6	27/09/2019	mev	Scene Recognition and Semantic Segmentation
0.7	30/09/2019	ims	Editorial checking
1.0	30/09/2019	ims	First 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).

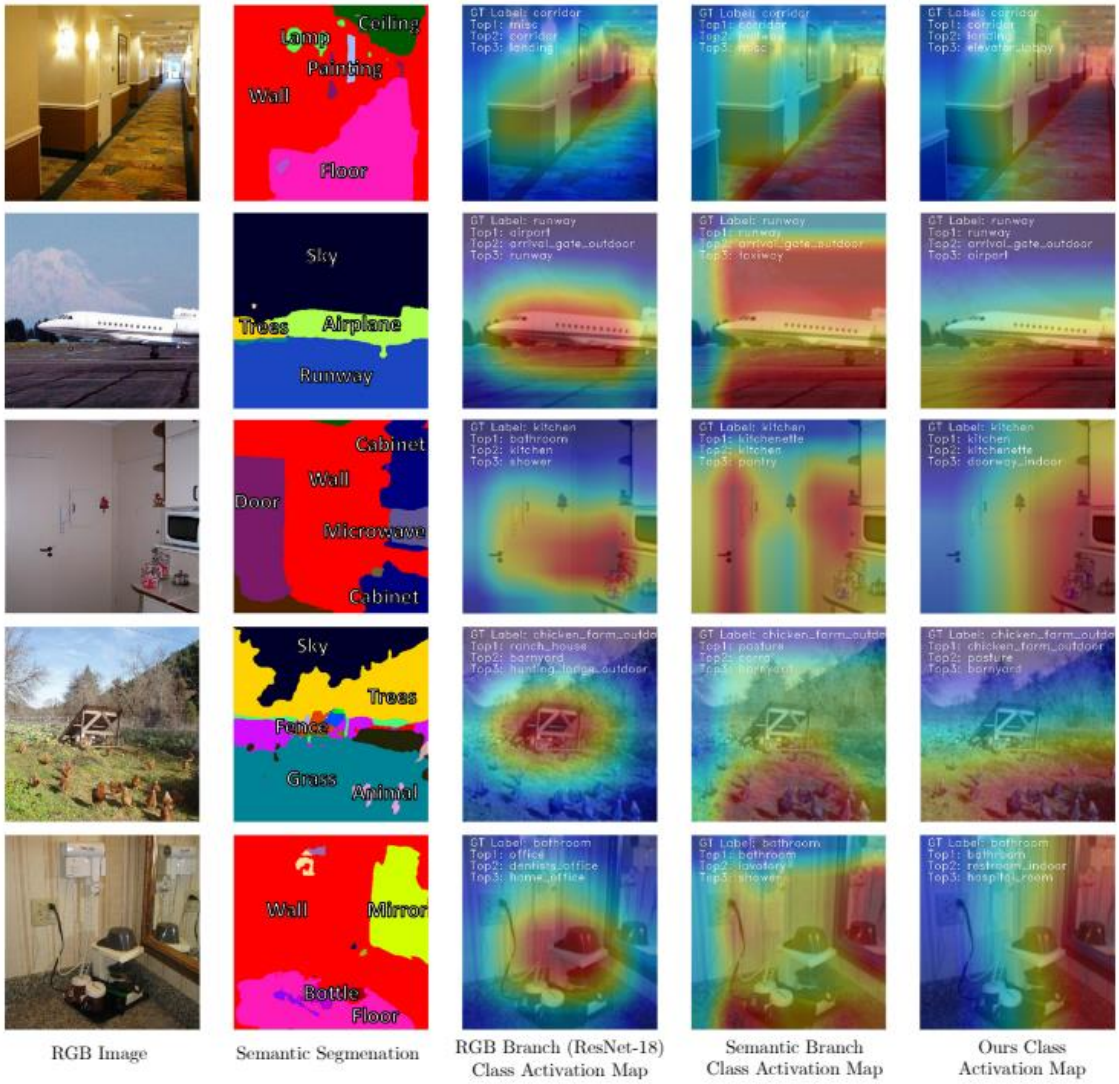


Figure 1. Qualitative results.

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.

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.

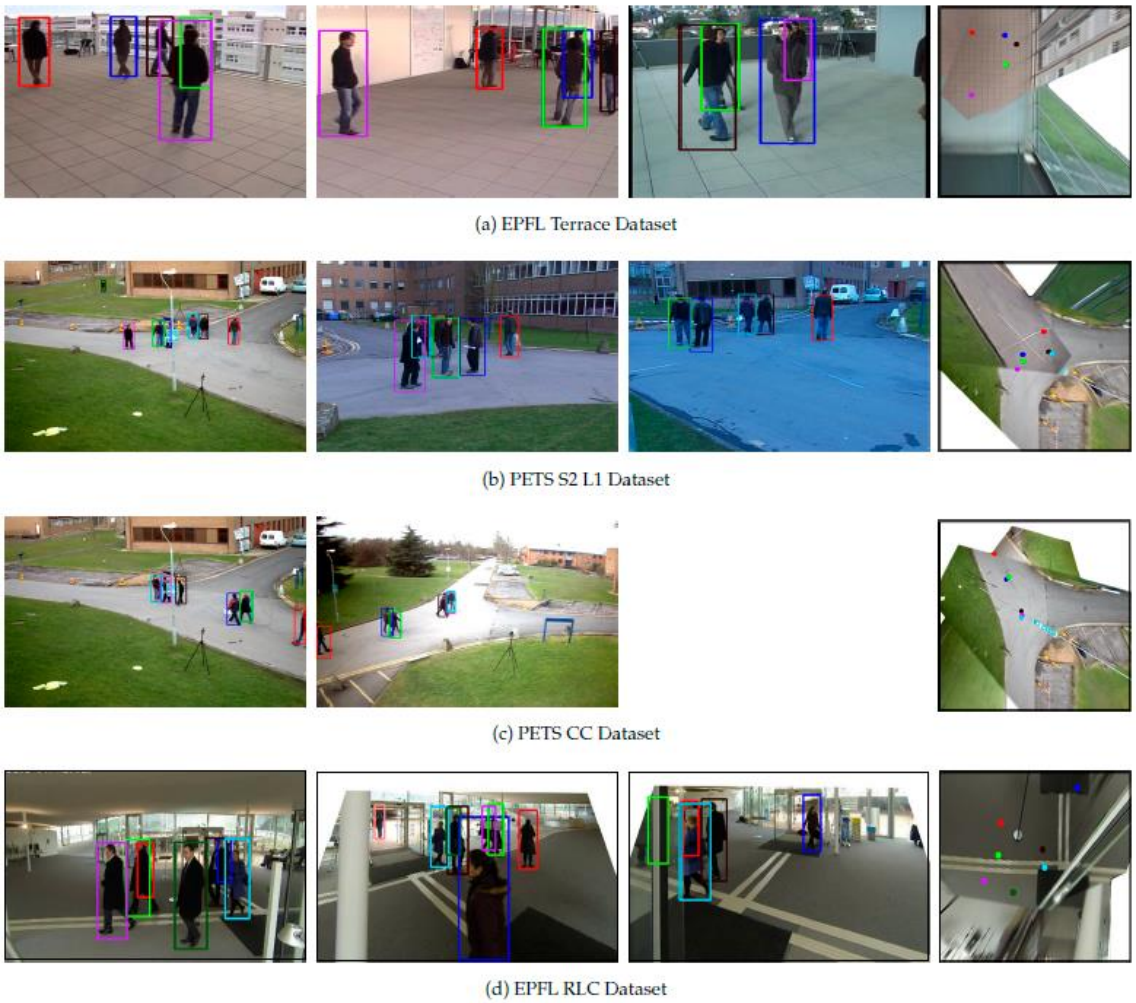
3.1.2. Experimental results

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 5 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 5. 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 2. Proposed method qualitative results on selected frames of the EPFL Terrace, PETS S2 L1, PETS CC and EPFL RLC datasets.

Qualitative results from Figure 2 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 6. 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 6 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 7. Wildtrack Dataset Comparison Results. All the stated methods (except both baselines) are multi-camera deep-learning based algorithms.

Table 7 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. A unified semantic segmentation (ongoing)

We have designed a python framework for the training of a semantic segmentation algorithm that jointly considers 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 align the semantic classes of five semantic segmentation dataset into a Unified Semantic Segmentation Dataset (see [2]). Currently, we are exploring the effect of the learning schedule and evaluating the hypothetical advantages and disadvantages of a so-trained semantic segmentation with respect to those trained with a single dataset.

This is an ongoing work that will be fully documented in the following version of this deliverable.

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 3** 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.

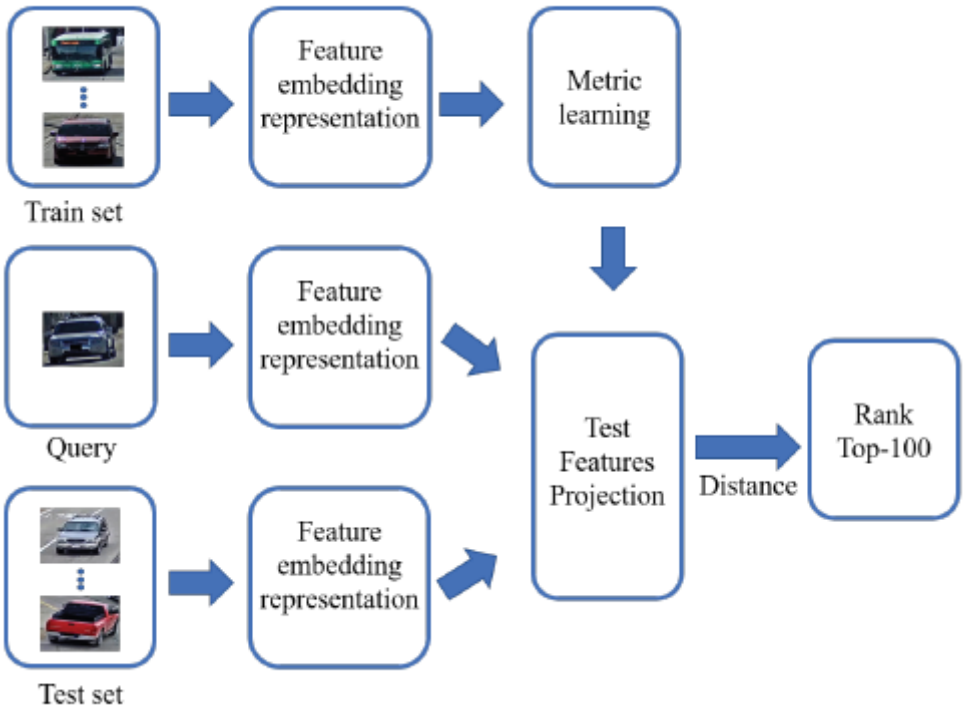


Figure 3. Flow diagram of the vehicle ReID system approach.

4.1.1.1. 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].

4.1.1.2. 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.

4.1.2. 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].

4.1.2.1. 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 4**, 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.

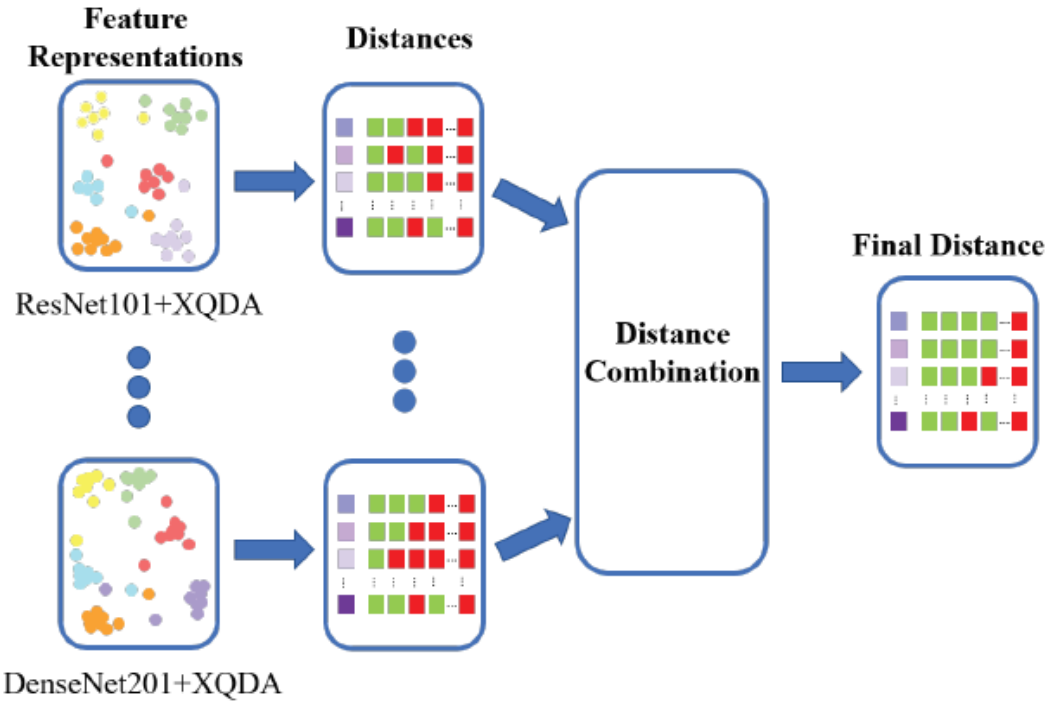


Figure 4. Feature combination at distance level.

4.1.2.1. 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.

4.1.3. 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 8** shows the people re-identification results obtained in dataset DuleMTMC4ReID [45] using Market1501 [44] as training dataset. **Table 9** shows the people re-

identification results obtained in dataset Market1501 [44] using DuleMTMC4ReID [45] as training dataset. **Table 10** 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 8 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 9 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 10 People re-identification results obtained in dataset ViPER [43].

4.1.4. 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 11** and **Table 12**. 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 11 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 12 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 13**. 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 13 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.

4.1.5. 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 14** 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 13** with the ones obtained in the AI City server in **Table 14**. 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 15** 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 14 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 15 Results of the leader board in [47].

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 5), 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.

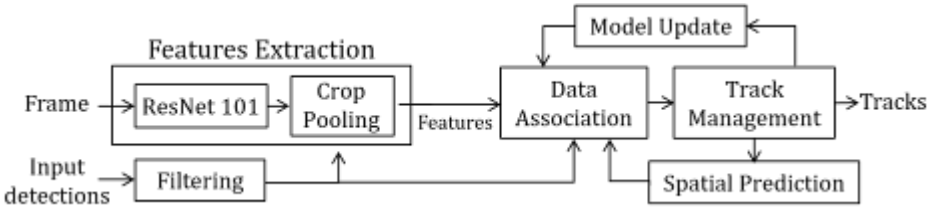


Figure 5. Block diagram of the proposed algorithm

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 16 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 6 depicts a sample frame with the identifiers and bounding boxes of the tracked vehicles.

Table 16. 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^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

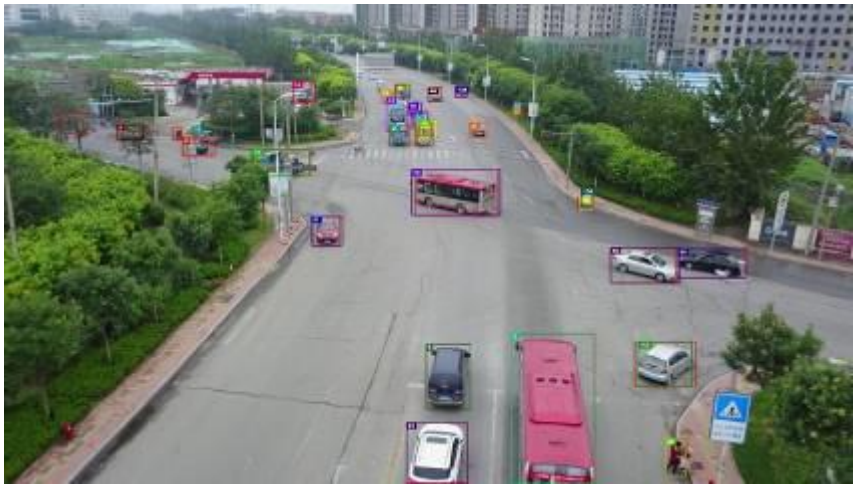


Figure 6. Sample frame with tracking results of one the sequences of the VisDrone 2018 dataset. Numbers stand for the identifiers of the tracked vehicles.

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 5, 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.

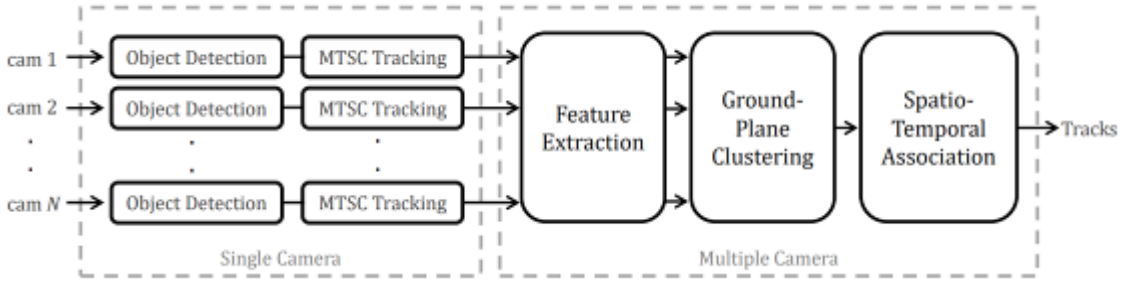


Figure 7. Block diagram of the proposed tracking method.

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. Results

Leaderboard of CityFlow Challenge is shown in Table 17. 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 17. 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 8 shows tracking results for scenario S01, formed by confronted cameras, in a similar way to S02.

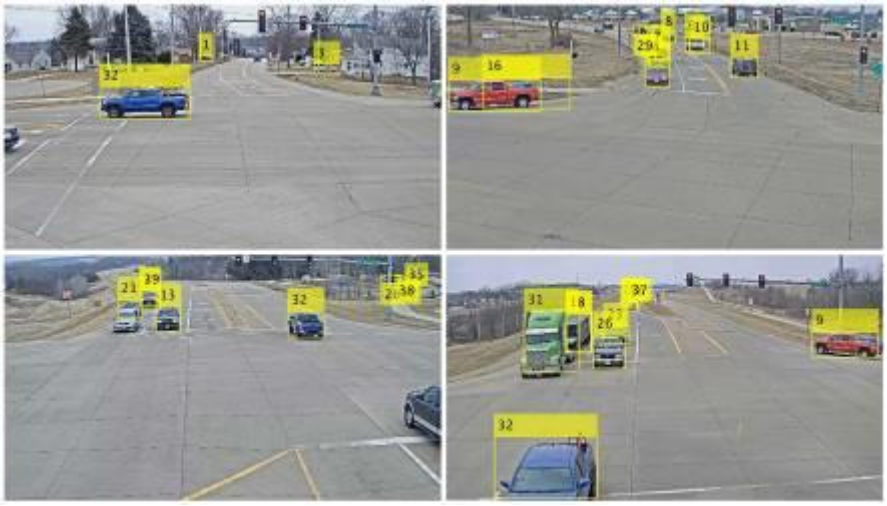


Figure 8. Sample visual results in train scenario S01, cameras 1-4. Tracked vehicles in yellow with their correspondent IDs. Same blue car is identified with the same ID, as well as the red car. However, an error in the single camera tracking leads to a tracking error in the red car in camera 2.

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