Domain Adaptation for Pedestrian DCNN Detector toward a Specific Scene and an Embedded Platform

Nada Hammami, Ala Mhalla and Alexis Landrault
Institut Pascal, Clermont Auvergne University, France

Keywords: Domain Adaptation, Deep Learning, Pedestrian Detection, Tracking, Optimization, Embedded System.

Abstract: Nowadays, the analysis and the understanding of traffic scenes become a topic of great interest in several computer vision applications. Despite the presence of robust detection methods for multi-categories of objects, the performance of detectors will decrease when applied on a specific scene due to a number of constraints such as the different categories of objects, the recording time of the scene (rush hour, ordinary time), the type of traffic (simple, dense) and the type of transport infrastructure. In order to deal with this problematic, the main idea of the proposed work is to develop a domain adaptation technique to automatically adapt detectors based on deep convolutional neural network toward a specific scene and to calibrate the network parameters in order to deploy it on an embedded platform. Results are presented for the proposed adapted detector in term of global performance in mAP and execution time onto a NVIDIA Jetson TX2 board.

1 INTRODUCTION

Pedestrian detection presents an essential task in many important applications of computer vision and artificial intelligence including security, road traffic control, behaviour analysis, driving assistance and video surveillance. This task is still complicated because of several factors such as crowded scenes, occlusion of pedestrians and illumination variation.

Over the past decade, there has been a significant effort dedicated to the development of effective pedestrian detection systems which are intended to support research in the development of safe intelligent vehicle systems.

In the recent years, Deep Convolutional Neural Networks (DCNNs) have been broadly adopted for pedestrian detection and achieved a significant progress on state-of-the-art performance; we cite mainly: Faster R-CNN (Ren et al., 2015), Single Shot Detector (SSD) (Liu et al., 2016) and YOLO (Redmon and Farhadi, 2017). Indeed, most pedestrian detectors are learned with a wide variety of labeled data that are selected from multiple situations to fully cover the appearance of pedestrians in the scenes to be studied. But usually when someone conducts a test on a specific scene, it can be noticed a drop in detector performance and the main reason of this performance degradation is the differences in appearance of example between the samples used for learning and those of the observed scene (variability in classes, difficult to have enough examples that can represent the scene). This problem can be solved by using domain adaptation techniques. These latter help to produce a scene-adapted detector that provides a superior performance than a generic one.

This paper provides an unsupervised domain adaptation approach to automatically adapt a DCNN detector toward a target scene to improve the detection performances. The proposed approach also introduces a solution to adapt the deployment of the proposed adapted DCNN detector on an embedded platform in order to install it on an autonomous vehicle.

A general synoptic of the proposed approach is shown in Figure 1. Given a generic DCNN detector trained by a generic labelled dataset and a target video sequence as inputs. The proposed approach allows to automatically adapt the parameters of the deep detector toward a target scene. After that, the generated DCNN detector is adapted to be deployed on an embedded platform.

- **Motivations and Contributions**

We propose a new approach which starts from a generic pedestrian detector to automatically generated a scene-specific CNN detector which can be deployed on an embedded platform. Our main motivations and contributions are summarized below.

Hammami, N., Mhalla, A. and Landrault, A.
Domain Adaptation for Pedestrian DCNN Detector toward a Specific Scene and an Embedded Platform.
DOI: 10.5220/0007360003210327
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 321-327
ISBN: 978-989-758-354-4

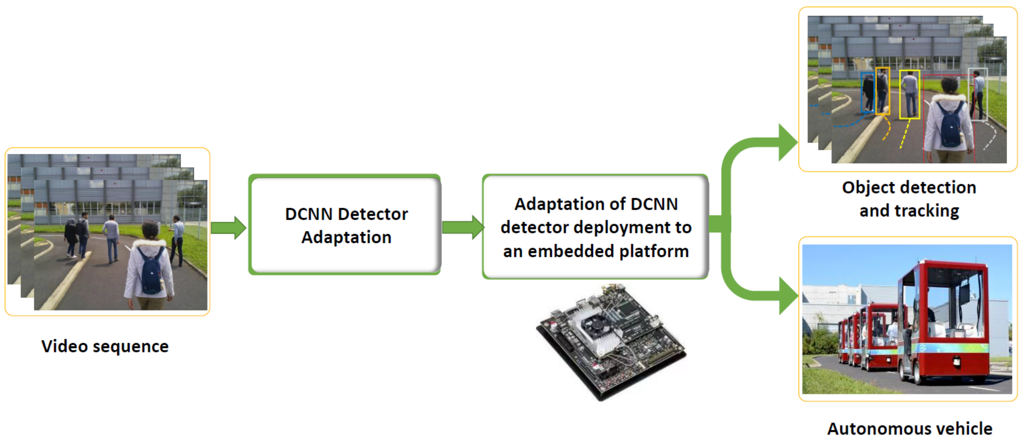
Video sequence
DCNN Detector Adaptation
Adaptation of DCNN detector deployment to an embedded platform
Object detection and tracking
Autonomous vehicle

Figure 1: General synoptic of proposed approach. Given a generic DCNN detector and a target video sequence as inputs, the proposed approach allows to adapt automatically the generic CNN detector to the target scene and to calibrate the parameters of the generated model toward an embedded platform.

- A domain adaptation method to adapt generic CNN pedestrian detector into a target video scene

- A correction function based on association algorithm that used to correctly select the positive samples and remove the negative ones from a specific scene

- An calibration method to deploy the pedestrian DCNN detector on an embedded platform for intelligent vehicles

The rest of the paper is organized as follows: the next section introduces some related works while the section 3 explains the different stages of the proposed approach. Experiments and results are given in the section 4 before to conclude and give some perspectives.

2 RELATED WORK

In this section, firstly, we describe the related works in pedestrian detection, after that recent works in domain adaptation and deep learning are outlined.

The problem of pedestrian detection and tracking tends to be one of the keys to create the future of intelligent vehicle. As a matter of facts, a lot of research groups are working on theories and new applications in this field.

Recently, several significant contributions have been proposed to improve the performance of pedestrian detection and tracking. Some of them have been based on DCNN theories (Zhang et al., 2016) (Li et al., 2018) (Mhalla et al., 2017b).

However, the performance of these above methods are often limited and drops significantly when tested to a specific scene due to the large variations between the source training dataset and the samples from the target scene. This problem can be resolved by domain adaptation approaches. These latter help to adapt a generic detector into a specific scene and provide a superior performance than a generic one.

The domain adaptation approaches aim to address the problem when the distribution of the learning data is different from that of the target distribution.

Over the past decades, several approaches have been suggested for domain adaptation in pedestrian detection (Htike and Hogg, 2014)(Mhalla et al., 2017a), which applied a generic detector on some frames in a target scene, predict samples and then collect most confident positive and negative samples to the original dataset for retraining. Rosenberg *et al.* (Rosenberg et al., 2005) used a semi-supervised approach based on background subtraction to label detection from a specific scene. The detection with high scores were selected in a learning dataset from one iteration to another. Contrarily, there was a risk of introducing wrong samples in the training dataset, which may degrade the performance of the detector. In the same direction, Wang *et al.* (Wang et al., 2014) used different visual information such as pedestrian motion, location and size to select positive and negative samples from the target scene and to collect the last ones in the training dataset for retraining. This method proved to be applied only to a specific detector. Accordingly, Mao *et al.* (Yunxiang Mao, 2015) presented an approach that used target samples verified by a tracklet method to adapt a generic detector toward a target scene.

Other solutions were proposed in (Maâmatou et al., 2016)(Li et al., 2015)(Wang et al., 2012), which collected new samples from the target scene and the source dataset. Mhalla *et al.* (Mhalla et al., 2017a) suggested an unsupervised domain adaptation approach based on the Monte Carlo filter steps to iteratively build a new pedestrian detector. Our proposed approach is inspired from this latter and is proposed to reduce several problems observed in the state-of-the-art adaptation frameworks.

Recently, addressing this problem with deep neuronal networks has gained an increased attention. In addition, deep learning has led to great performance on a variety problems of computer vision like multi-object detection (Redmon and Farhadi, 2017), action recognition (Will Y. Zou, 2011) and image classification (Duan et al., 2009).

Differently from the existing works mentioned above, our approach presents the first adapted approach which apply to adapt deep detector for mobile cameras contrarily to the state-of-the-art adaptation ones were limited only for stationary cameras (Mhalla et al., 2017a)(Maâmatou et al., 2016).

In this paper, we propose a new approach based on domain adaptation and deep learning techniques to adapt the recent deep detectors to a specific scenes and to optimize the DCNN parameters in order to deploy it on intelligent vehicle.

In what follows, we first describe the adaptation of the DCNN model, and then deal with the optimization step.

3 PROPOSED APPROACH

This section aims to present our approach for pedestrian detection and tracking to implement it on a autonomous vehicle. The block diagram of the proposed approach is shown in Figure 2. Our approach is divided into two parts, a domain adaptation technique to automatically adapt DCNN detectors to the scene to be analyzed and a second technique to adapt the deployment of the scene-specific detector on embedded platforms.

Our domain adaptation technique automatically estimates the target distribution as a set of learning samples. These are selected from the samples from the target scene based on a selection strategy that indicates that they belong to the estimated target distribution. In the following subsections, we will describe the different steps of our adaptation technique.

The first part of our framework ”Adaptation of the DCNN detector” follows three steps: sample proposal, correction and adaptation.

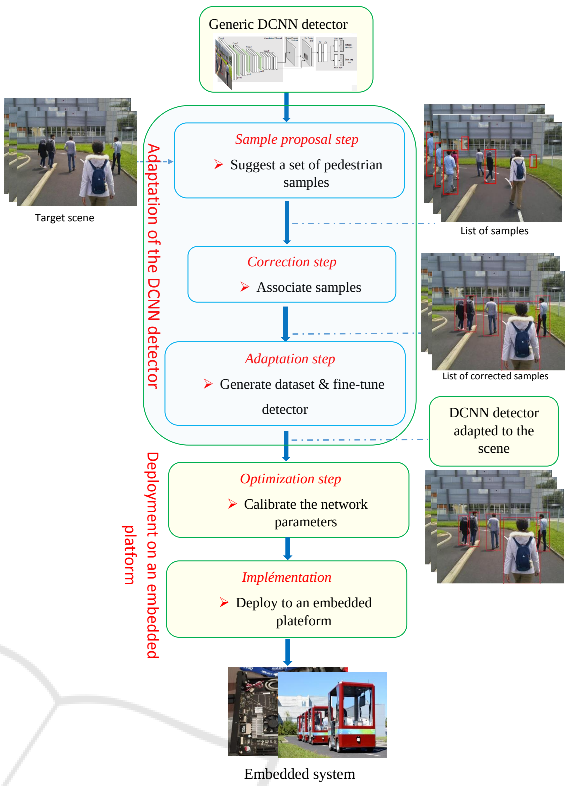
Generic DCNN detector
Target scene
Adaptation of the DCNN detector
Sample proposal step
➤ Suggest a set of pedestrian samples
List of samples
Correction step
➤ Associate samples
List of corrected samples
Adaptation step
➤ Generate dataset & fine-tune detector
DCNN detector adapted to the scene
Deployment on an embedded platform
Optimization step
➤ Calibrate the network parameters
Implémentation
➤ Deploy to an embedded plateform
Embedded system

Figure 2: Block diagram of the proposed approach.

3.1 Sample Proposal Step

The sample proposal step consists in providing a list of samples predicted by a detector based on DCNN applied to a target scene (Equation 1):

$$\{\mathbf{x}^{(n)}\}_{n=1}^{N} = \Theta(\{\mathbf{I}^{(i)}\}_{i=1}^{I}; \mathcal{DCNN}) \tag{1}$$

with *DCNN* represent the DCNN detection model, $\{\mathbf{I}^{(i)}\}_{i=1}^{I}$ list of images and $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ represents the samples provided by the output layer of the DCNN detector. Then, a selection of the samples is ensured in the correction step which makes it possible to collect the true positive samples and to eliminate the false ones.

3.2 Correction Step

The correction step consists in removing the false positive samples and keeps the true ones. This step is based on a data association algorithm which makes it possible to determine the trajectories of objects in motion and through these trajectories it is possible to predict the false positive samples (Equation 2).

$$\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}^{(n)}\}_{n=1}^{N} = f_{Corr}(\{\mathbf{x}^{(n)}\}_{n=1}^{N}) \tag{2}$$

with $\tilde{D}$ the adapted dataset including true positive samples $\{\tilde{\mathbf{x}}^{(n)}\}_{n=1}^{N}$ selected by the correction function f_{Corr}.

During the correction step, the correction function is applied to the samples of the list provided by the sample proposal step. This proposed function is based on a data association algorithm, which makes it possible to calculate the trajectories of moving objects by associating samples of the same objects during a video sequence. And in order to determine the object trajectories in the target scene, we use a DeepSORT (Wojke and Bewley, 2018) association algorithm which is a real-time multi-object tracker based on an inter-image association of samples that takes into account the coherence of the apparent movement and the appearance of the tracked targets. A CNN network extracts the characteristics of each sample generated by the sample proposal step ; these are then associated with the existing trajectories via a Hungarian algorithm (Kuhn, 2005) taking into account the position of the target and the feature vector generated by the CNN network. Unassociated trajectories are extended by sample proposals of a Kalman filter and unassociated detection generate new trajectories. DeepSORT is simple, very inexpensive in computing time and has shown good performance in the MOT Challenge 2017; for these reasons, it seems adapted to our constraints.

3.3 Adaptation Step

The adaptation step serves to produce an adapted CNN detector driven by the sample list generated by the previous step (Equation 3).

$$\{\tilde{\mathcal{DCNN}}\} = f_{Fine}(\{\mathbf{I}^{(i)}\}_{i=1}^{I}, \tilde{\mathcal{D}}; \mathcal{DCNN}) \quad (3)$$

with $\tilde{\mathcal{DCNN}}$ is the detector generated after fine-tuning with the list of samples $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}^{(n)}\}_{n=1}^{N}$ generated by the correction step and $\{\mathbf{I}^{(i)}\}_{i=1}^{I}$ represents the list of images.

The adaptation step consists in fine-tuning the DCNN detector with samples provided by the correction step of our domain adaptation approach in order to generate a new detector adapted to the target scene with a significant performance compared to the generic detector (input).

The second part of our approach is a solution to adapt the deployment of our adapted DCNN detector to an embedded platform in order to install it on a autonomous vehicle. To do this, we proposed an optimization step to accelerate and to compress the DCNN model generated by our domain adaptation technique. This part follows two steps: Optimization and Implementation. In the following subsections, we will describe these different steps.

3.4 Optimization Step

Given the trained DCNN model provided by our first adaptation technique, the optimization step optimizes the DCNN model parameters. To do this, we propose to convert the weights and activations parameters of the adapted model, which is stored on 32-bit floating point, on 8-bit integer. This conversion function (Equation 4) allows to reduce memory usage, allowing deployment of larger networks and transferring data takes less time.

$$\{\tilde{\mathbf{W}}^{(i)}\}_{i=1}^{K} = f_{Conv}(\{\mathbf{W}^{(i)}\}_{i=1}^{K}) \quad (4)$$

where $\{\mathbf{W}^{(i)}\}_{i=1}^{K}$ represents the DCNN model parameters and $\{\tilde{\mathbf{W}}^{(i)}\}_{i=1}^{K}$ is the set of the output parameters converted on 8-bit integer. During this step, we used the ”TensorRT” library.

3.5 Implementation step

The implementation step consists in deploying our adapted and optimized detector on the embedded platform based on GPU or strong CPU devices in order to guarantee a real-time running of the detector.

4 EXPERIMENTS

The improvements in performance and speed brought by the method of adapting the detector to a specific scene and his adaptation on an embedded platform are evaluated in this section. In fact, this approach is generic and can be applied to any DCNN detector. For the evaluation we mainly used the YOLO V3 (Redmon and Farhadi, 2018) deep detector.

4.1 Datasets

The MSCOCO dataset (Lin et al., 2014) was used to learn the generic DCNN detector. In our experiments, we used all annotated pedestrians to train the generic DCNN detector.

The evaluation was achieved on a private dataset:

- **PAVIN Dataset.** It is a private dataset containing a video sequence of 30 minutes, recording a scene of road traffic by a mobile camera fixed on an autonomous vehicle. We have uniformly extracted as proposed in (Mhalla et al., 2017b) 452 images of this video were used for adaptation and the last 100 images were used for the test. Figure 3 illustrates image examples of the PAVIN dataset.

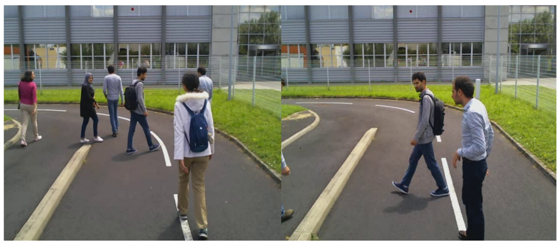

Figure 3: Image examples of PAVIN dataset.

Table 1: Comparison between adapted and generic detectors on the PAVIN dataset.

Specification	Performance (mAP)
Generic DCNN detector	64.6%
Adapted DCNN detector	**76.37%**

4.2 Implementation Details

The implementation details of the given approach are described in this section. In our experiments, YOLO detectors are used as input. These detectors are used under Darknet (Redmon, 2016). The shared part of these detectors is an architecture invented by the creators and includes 19 layers of convolution and pooling. YOLO detectors use two completely connected layers. The particularity of this detector intervenes in the learning phase. The network is fully resized during learning while retaining the settings between each size change. This strategy makes it possible to learn the influence of a resizing on the quality of the sample proposals. Indeed, there are several versions of YOLO. In our experiments, we used the Tiny YOLO V2 (Redmon and Farhadi, 2016) which is composed of 9 convolutional layers and 6 max pooling ones. Also, the YOLO V3 version has been used. This detector has undergone some changes and improvements in the number of classification layers to improve classification accuracy compared to previous versions of YOLO V2.

4.3 Results and Analysis

Given the PAVIN dataset and its ground-truth, a comparative summary table for pedestrian detection (see Table 1) is provided.Where the global performance in mAP of the adapted detector compared to the generic one.

Table 1 shows that the adapted detector generated by our domain adaptation approach significantly exceeds the generic detector. The performance on the test set of the PAVIN dataset has been improved from 64.6% to 76.37%. In particular, it has an improvement rate of **12%**. The illustration in Figure 4

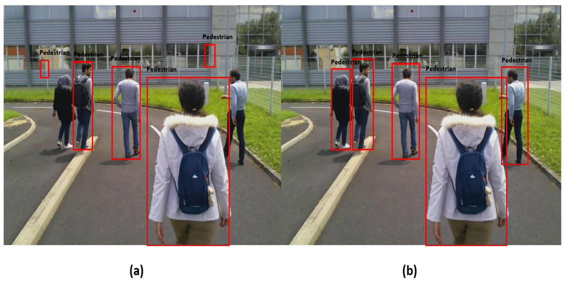

Figure 4: (a), (b): improvement of our proposed adaptation technique (the left images shows the detection results for the generic DCNN detector and the right for the adapted one) on PAVIN dataset.

Table 2: Description of the correction step effect.

Specification	Performance
Correction step based score	69.42%
Correction step based association	**76.37%**

clearly shows the improvement of the adapted DCNN detector in detecting pedestrians and in removing the false positive samples compared to the generic one.

To show the effectiveness of our correction step, Table 2 shows the comparison between the use of the correction step based on the confidence scores predicted by the DCNN detector (selected samples that have a confidence score higher than a score threshold of 0.5) and between our correction step based on association algorithm.

Line 2 of Table 2 presents our proposed approach based on the association algorithm, while the values in the first line indicate the use of the confidence score provided by the output layer of the DCNN detector. The results show that the proposed correction step based on the use of the association strategy improves the performance of our domain adaptation approach (see Figure 5).

However, we can improve the correction step with other complex visual cues, such as for example a combination of the association algorithm with other spatio-temporal information, to improve the selection of true positive samples.

For the second part of our approach ”Adaptation of the detector on an embedded platform”, we provide a comparative table containing the DCNN detector running speed in Frames Per Second (FPS) before and after the optimization step (see Table 3).

The experiments showed promising results from

Table 3: DCNN detector optimization results on the Jetson TX2.

Specification	Running
Adapted detector before optimization	2.5 FPS
Adapted detector after optimization	**9 FPS**

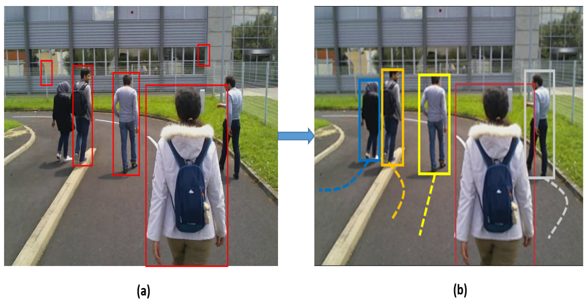

Figure 5: Efficiency of proposed correction step. The red blobs in image (a) present the inputs of the correction step and the color ones in image (b) are the outputs.

Table 4: Description of running optimized detectors on the Jetson TX2 board.

Detector	Running time
YOLO V3 (Baseline)	2.5 FPS
Optimized YOLO V3	**9 FPS**
Tiny YOLO V2 (Baseline)	17 FPS
Optimized Tiny YOLO V2	**23 FPS**

the application of our optimization step. The first column of Table 3 represents the DCNN detector specifications before and after the optimization step. The second column shows the DCNN detector running speed in FPS. The results show that the optimization step based on calibrating the parameters of DCNN models significantly improves the running speed of the DCNN detector.

For the hardware components, we used the Jetson Tegra TX2 embedded platform. The latter is a recent technology developed by NVIDIA. This device delivers the performance of the NVIDIA Maxwell architecture with 256 CUDA cores delivering more than 1 Tera FLOP performance, 64-bit processors and a 5 mega pixel camera. To run our adapted DCNN detector on the NVIDIA Jetson TX2 device, we use the Darknet deep learning framework with the C ++ programming language.

In Table 4, we summarize the running speed of our adapted and optimized detectors on the NVIDIA Jetson TX2. According to Table 4, we test various adapted and optimized DCNN detector compared to the baseline ones on our embedded system with the NVIDIA Jetson TX2, we chose the optimized adapted Tiny YOLO V2 detector to obtain an embedded system which can be run in real-time.

Our approach of detector adaptation towards a specific scene and an embedded platform, allow to provide an embedded system with a good detection performance and which can work in real time for an autonomous vehicle.

5 CONCLUSION

This article introduces an approach for pedestrian detection that consists first of all in proposing a new domain adaptation technique from a DCNN detector to a specific scene by adapting a generic detector to an urban traffic scene without labeling. The method gave good results on real world-scenarios compared to the generic DCNN detector in term of mAP and running time. Furthermore, the proposed approach presents the first domain adaptation approach which apply to adapt deep detector for mobile cameras. The experiment results obtained on an Jetson TX2 embedded platform have shown that adapted detector presents very interesting performance in term of real-time running time and the future works include the extension to other types of detectors as the proposed approach is generic and flexible.

ACKNOWLEDGEMENTS

This work is part of a master 2 internship in Artificial Perception and Robotics at Clermont Auvergne University (France). It is sponsored by the French government research program ”Investissements d'avenir” through the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2007-2013 (ERDF - Auvergne region), and by the Auvergne region.

REFERENCES

Duan, L., Tsang, I. W., Xu, D., and Maybank, S. J. (2009). Domain transfer svm for video concept detection. In *CVPR*, pages 1375–1381. IEEE.

Htike, K. K. and Hogg, D. C. (2014). Efficient non-iterative domain adaptation of pedestrian detectors to video scenes. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 654–659. IEEE.

Kuhn, H. W. (2005). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1):7–21.

Li, J., Liang, X., Shen, S., Xu, T., Feng, J., and Yan, S. (2018). Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996.

Li, X., Ye, M., Fu, M., Xu, P., and Li, T. (2015). Domain adaption of vehicle detector based on convolutional neural networks. *International Journal of Control, Automation and Systems*, pages 1020–1031.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context.

In *European conference on computer vision*, pages 740–755. Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Maâmatou, H., Chateau, T., Gazzah, S., Goyat, Y., and Essoukri Ben Amara, N. (2016). Transductive transfer learning to specialize a generic classifier towards a specific scene. In *VISAPP*, pages 411–422.

Mhalla, A., Chateau, T., Gazzah, S., and Amara, N. E. B. (2017a). Specialization of a generic pedestrian detector to a specific traffic scene by the sequential monte-carlo filter and the faster r-cnn. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, (VISIGRAPP 2017)*, pages 17–23. INSTICC, SciTePress.

Mhalla, A., Maâmatou, H., Chateau, T., Gazzah, S., and Essoukri Ben Amara, N. (2017b). Smc faster r-cnn: Toward a scene-specialized multi-object detector. *Computer Vision and Image Understanding*, 164:3–15.

Redmon, J. (2013–2016). Darknet: Open source neural networks in c. http://pjreddie.com/darknet/.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. *arXiv preprint*.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*.

Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *WACV/MOTION*, pages 29–36.

Wang, M., Li, W., and Wang, X. (2012). Transferring a generic pedestrian detector towards specific scenes. In *CVPR*, pages 3274–3281. IEEE.

Wang, X., Wang, M., and Li, W. (2014). Scene-specific pedestrian detection for static video surveillance. *PAMI*, pages 361–362.

Will Y. Zou, Serena Y. Yeung, A. Y. N. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CSD*, pages 3361–3368.

Wojke, N. and Bewley, A. (2018). Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE.

Yunxiang Mao, Z. Y. (2015). Training a scene-specific pedestrian detector using tracklets. pages 170–176.

Zhang, L., Lin, L., Liang, X., and He, K. (2016). Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer.