Training My Car to See using Virtual Worlds

Antonio M. López[a,b,1], Gabriel Villalonga[a,b], Laura Sellart[a], Germán Ros[a,b], David Vázquez[a,b], Jiaolong Xu[a,b], Javier Marín[c,2], Azadeh Mozafari[a,3]

[a] *Computer Vision Center (CVC), Edifici O, Campus Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Barcelona, Spain*
[b] *Dpt. Ciències de la Computació (DCC), Escola d'Enginyeria, Campus UAB, 08193 Bellaterra, Barcelona, Spain*
[c] *Delphi Deutschland GmbH*

Abstract

Computer vision technologies are at the core of different advanced driver assistance systems (ADAS) and will play a key role in oncoming autonomous vehicles too. One of the main challenges for such technologies is to perceive the driving environment, *i.e.* to detect and track relevant driving information in a reliable manner (*e.g.* pedestrians in the vehicle route, free space to drive through). Nowadays it is clear that machine learning techniques are essential for developing such a visual perception for driving. In particular, the standard working pipeline consists of collecting data (*i.e.* on-board images), manually annotating the data (*e.g.* drawing bounding boxes around pedestrians), learning a discriminative data representation taking advantage of such annotations (*e.g.* a deformable part-based model, a deep convolutional neural network), and then assessing the reliability of such representation with the acquired data. In the last two decades most of the research efforts focused on representation learning (first, designing descriptors and learning classifiers; later doing it end-to-end). Hence, collecting data and, especially, annotating it, is essential for learning good representations. While this has been the case from the very beginning, only after the disruptive appearance of deep convolutional neural networks it became a serious issue due to their data hungry nature. In this context, the problem is that manual data annotation is a tiresome work prone to errors. Accordingly, in the late 00's we initiated a research line consisting in training visual models using photo-realistic computer graphics, especially focusing on assisted and autonomous driving. In this paper, we summarize such a work and show how it has become a new tendency with increasing acceptance.

Keywords: ADAS, Autonomous Driving, Computer Vision, Object Detection, Semantic Segmentation, Machine Learning, Data Annotation, Virtual Worlds, Domain Adaptation

1. Introduction

1.1. Mobility, ADAS & Self-Driving Vehicles

The 20th century brought affordable automobiles for the middle classes. This caused a major benefit in terms of personal mobility and sense of freedom. Overall, automotive industry has been a major economic driving force since then. However, the introduced benefits were at the expense of other significant drawbacks which have become worldwide concerns that cannot be ignored. We refer to traffic accidents and environmental contamination; both factors end up in healthy issues in a way or another, as well as in a huge economic cost. Moreover, nowadays half the world's population lives in cities, and this proportion is constantly increasing [1]. The World Health Organization (WHO) predicts 70% by 2050, which strains city traffic more and more. Therefore, during daily life, mobility based on personal vehicles may not provide a sense of freedom and comfort as used to do.

The aforementioned considerations lead us to think that personal automobiles, as they are operating now, do not fit well in the future cities that industrialized countries are designing, *i.e.* the so-called *smart cities*. Indeed, for becoming a smart city, among other issues, municipalities must rethink road usage. Accordingly, in addition to the improvement of massive public transport, we envision a centralized system in the city receiving mobility requests from citizens, both for persons and goods. The city would control a fleet of automated vehicles (*i.e.*, self-driven) that would cooperate between each other and with other elements of the city like infrastructure surveillance cameras, traffic lights, etc., to move safely and comfortably, saving energy, minimizing traffic congestion and pollution, bringing back mobility freedom for the elderly, the temporally or permanent handicapped, the kids, and all citizens in general.

The foreseen intelligent transportation network described

[1] Corresponding author: antonio@cvc.uab.es
This work is supported by the Spanish MICINN projects TRA2014-57088-C2-1-R, by the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya (2014-SGR-1506), and also by the Spanish DGT (SPIP2014-01352). Our research is also kindly supported by NVIDIA Corporation in the form of different GPU hardware.
[2] Contribution done during his PhD at CVC/UAB.
[3] Contribution done during her 6 months visit to CVC.

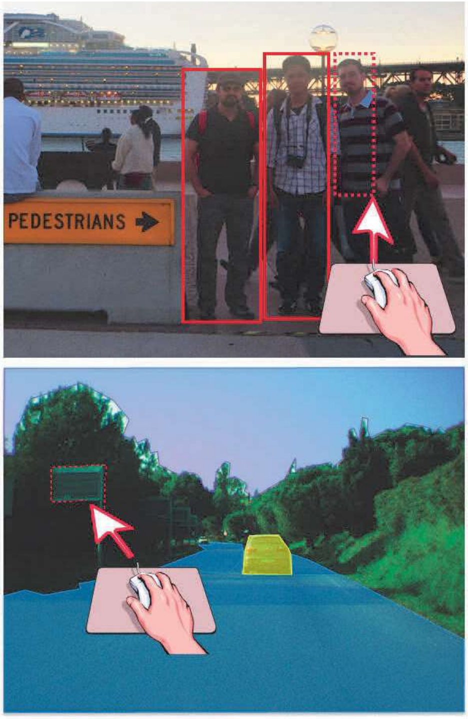


Figure 1: Manual annotation. Top: based on the bounding boxes (BBs) of the object of interest (*e.g.* pedestrians). Bottom: by delineating the silhouettes of the classes of interest (sky, road, car, etc.), which is more precise than BBs, but much more time consuming.

above, cannot be developed in a short time. Research, development, regulations, etc. are required in the step-by-step route towards the final objective. Accordingly, a particularly important question is that, along the way, our current transportation systems need to cohabit with the new introduced paradigms, until the latter ones totally replace the former. Overall, we think that we have to work now towards the described future, but taking into account its gradual implantation cohabiting with the systems of the present. Therefore, the starting point consists of progressively incorporating ADAS to human-driven or driver-in-the-loop vehicles, while working to reliable fully self-driving vehicles which, in turn, would lead to efficient intelligent transportation networks.

1.2. On-board Vision-based Perception

In this context, developing a reliable artificial intelligence able to analyze sensor raw data to achieve *scene understanding* is an essential question for on-board ADAS and self-driving vehicles, since it is the base to perform both assisted and automated safe maneuvers.

Vision-based perception is a core ingredient of on-board scene understanding. Many ADAS can rely only on vision, *e.g.* lane marking detection [2, 3, 4], traffic sign recognition [5, 6, 7], intelligent headlights control [8, 9, 10]. Environment perception for self-driving vehicles will rely on sensor fusion, eventually including active sensors such as LIDAR, RADAR, and GPS/IMU systems [11, 12], undoubtedly incorporating vision too; especially since its performance boost due to deep convolutional neural networks (deep CNNs) [13, 14, 15, 16, 17] and the introduction of powerful embedded super-computers for them [18]. After all, vision is the main sense used by human drivers. In addition, computer vision not only applies to the visual spectrum but also to the far-infrared (FIR) one [19, 20, 21].

Nowadays it is clear that machine learning techniques are essential for developing such a visual perception for driving. In particular, the standard working pipeline consists of: (1) collecting data, *i.e.* on-board images; (2) manually annotating the data, *e.g.* drawing bounding boxes (BBs) of the classes of interest or delineating their silhouette (Fig. 1); (3) learn a discriminative data representation taking advantage of such annotations, *e.g.* a holistic model [22], a deformable part-based model (DPM) [23], a patch-based model [24], an end-to-end hierarchical model [17], even spatio-temporal models [25]; and (4) quantitatively and qualitatively assessing the reliability of the learned representation by using the acquired data.

1.3. Data Annotation, Variability

In the last two decades most of the research effort focused on representation learning. Hence, collecting data and, especially, annotating it, is crucial for learning good representations. While this has been the case from the very beginning, only after the disruptive appearance of deep CNNs it became a serious issue due to their data hungry nature. In this context, the problem is that manual data annotation is a tiresome work prone to errors. For instance, annotating a BB (Fig. 1–Top) requires around 6 seconds in average per human annotator [26], while annotating all the image pixels by silhouette delineation (Fig. 1–Bottom) may require from 30 to 60 minutes per human annotator, depending on the image content and the number of classes of interest [27]. Of course, the annotation time and the inaccuracy of the annotations increase with the time the same person has been working on it. Moreover, just annotating data in a *blindly* repetitive manner does not guarantees diversity, a key point to learn really representative models that can generalize well to operate with previously unseen data.

Accordingly, to face the challenge of collecting visual data annotations, different web-based tools have been proposed under the crowdsourcing paradigm. One popular example is LabelMe [28] which permits human annotators to localize image objects of an established class category by framing them with polygons. Another powerful example is Amazon's Mechanical Turk (MTurk) [29], which allows researchers to define *human intelligence tasks* of different difficulty (*e.g.* from marking points of interest to drawing polygons) to be taken by human online workers. In addition, some approaches [30] even try to transform data annotation into a web-based interactive game. Unfortunately, as it is argued in [31], where these web-based tools and others are analyzed, *it is a fallacy to believe that, because good datasets are big, then big datasets are good.*

In fact, in order to favor variability and reduce the annotation effort, this task has been put in the loop of representation learning, we refer to the *active learning* paradigm [32]. For instance, in [33] and [34] the idea is applied for developing pedestrian and vehicle detectors, respectively. In these cases active learning consists of a stage to obtain an initial object classifier, followed by a loop in which the current classifier is plugged in a detector that is applied to unseen videos, a *human oracle* performs *selective sampling* (*i.e.* annotation of image windows falling into the classifier *ambiguity region*) and then previous and new annotations are used for re-training the classifier. The process is iterated with new videos until a desired performance is achieved. On the other hand, the variability on the annotated data is limited by the actual collected videos and, thus, by the real-world conditions at the moment of acquiring them.

Overall, with a great effort of several research communities different annotated datasets have been collected mainly following the web-based crowdsourcing paradigm. Remarkable examples are the PASCAL VOC Challenge [35] and MS-COCO [36] for object detection and segmentation, and ImageNet [37] for image classification.

It is also worth to mention that for augmenting the variability of the already annotated data, different transformations can be applied. For instance, in [38] a set of training pedestrians is enlarged by transforming their shape (a manually delineated silhouette) and texture, as well as the texture of the background, which is combined with standard jittering and mirroring. In [39], to address automatic license plate recognition, a two-step framework is proposed for specifically generating training plates. First, license plate images are synthesized and, second, image transformations and distortions typically encountered in real images are applied. Nowadays using generic image-transformation-based *data augmentation* is also a common practice for training deep CNNs [13, 40].

Focusing on the application area in which we are mainly interested, *i.e.* ADAS and self-driving vehicles, we can see that there are not publicly available datasets with annotations in the order of magnitude of PASCAL VOC, MS-COCO or ImageNet. Usually, we can find publicly available dataset for a particular task, *i.e.* where data is annotated only with the information required for the task. One of the most popular cases has been pedestrian detection with datasets[4] such as INRIA [22], Daimler [41], ETH [42], Caltech [43], and CVC [44, 45, 46], which contain images of the visual spectrum, or others like [47, 21] with images of the FIR one. We can find also analogous datasets for vehicles [34] and Traffic Signs [48, 6]. In all of these datasets the objects of interest (*i.e.* pedestrians, vehicles, traffic signs) are annotated with BBs.

A remarkable effort for providing publicly available data and standard evaluation protocols is the KITTI benchmark suite [12, 49, 50, 51]. It includes annotations for driving tasks such as object detection (pedestrians, vehicles, cyclists) and road surface segmentation among others. Interestingly, for the task of semantic segmentation different members of the research community were annotating KITTI images. Of course, this was done by manually delineating the silhouette of classes of interest such as pedestrians, vehicles, cyclists, traffic signs, sidewalks, buildings, vegetation, etc. Despite of this effort, at the moment of writing this paper there are only around 550 of such pixelwise annotated images. Another popular example of such kind of annotations is CamVid [52] with 701 images.

Since these are really few annotated images given the complexity of the task, *i.e.* semantic segmentation, new efforts have been done for providing more pixelwise annotated data. For instance, annotating 3D data and back-projecting the annotations to corresponding 2D images has been explored to increase annotation productivity as well as the quality of the annotations [27]. However, no dynamic objects are included in the annotation process (*e.g.* moving pedestrians). Using the traditional manual effort, the Cityscapes Dataset [53] has been recently created by acquiring images in 50 cities, including around 5,000 images with fine silhouette-based annotations, and 20,000 images with coarsely annotated silhouettes. The dataset also contains instance-level annotation, as well as vehicle odometry, stereo and GPS information.

1.4. Learning to See using Virtual Worlds

Given the enormous difficulty of manually annotating visual information, we have to really thank the effort done in the above mentioned works. On the other hand, foreseeing such difficulties, in late 00's we decided to explore a different alternative. Looking at Fig. 2 we can see that manual paintings can evoke real counterpart images even not being exactly equal, and the same happens with digitally generated images and their real counterparts. In the latter case, the advantage is that computers can also automatically generate the desired annotations (*e.g.* the silhouette of the head). Therefore, we decided to explore the synergies between modern Computer Animation and Computer Vision in order to *close the circle*: the Computer Animation community is modeling the real world by building increasingly realistic virtual worlds, thus, *can we now learn our models of interest in such controllable virtual worlds and use them back successfully in the real world?*. Note that modern videogames, simulators and animation films, are gaining photo-realism. In fact, all the ingredients for creating *soft artificial life* are being improved: visual appearance both global (3D shape, pose) and local (texture, where involved Computer Graphics aim at reaching the power spectrum of real images [54]), kinematics, perception, behavior and cognition. For instance, see [55] for the case of animating autonomous pedestrians in crowded scenarios. This means that, eventually, from such virtual worlds we could collect an enormous amount of automatically annotated information in a controlled manner.

[4] The page `www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/` keeps track of different pedestrian detection datasets.

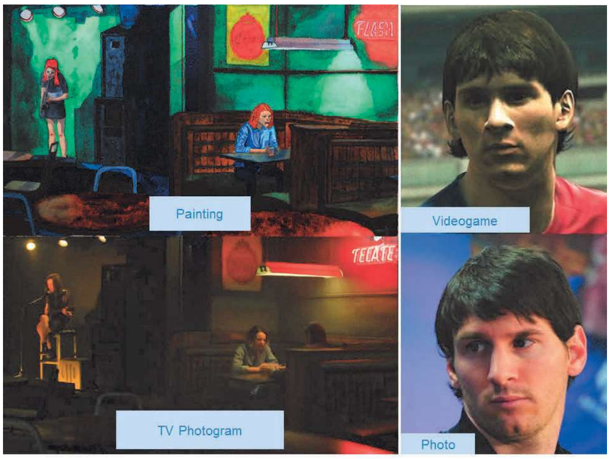


Figure 2: Left: a painting that evokes a real-world TV snapshot, even not really being equal. Right: a videogame character that visually resembles its real counterpart model.

In order to face such a question, and given the fact that our focus is on ADAS and self-driving vehicles, we decided to start by assessing the usefulness of virtual worlds for the challenging task of appearance-based pedestrian detection. We explain this work in Sect. 2. We think that lessons learned in pedestrian detection can be extrapolated to object detection in general (*e.g.* see [56], for the case of *vehicle detection*). In the same application context, as we have mentioned before, another challenging task is semantic segmentation. Therefore, applying again our work strategy of facing first the most challenging problems, after pedestrian detection we focused on semantic segmentation. We draw this work in Sect. 3. During our transition from pedestrian detection to semantic segmentation, the breakthrough in computer vision due to deep CNNs happened; thus, while our work of pedestrian detection was based on traditional object representations, *i.e.* human designed image descriptors with principled classifiers and/or ensembles, our work on semantic segmentation already starts using deep CNNs. As we will see, an especial contribution of our work is to cast as a *domain adaptation* problem the fact of learning visual representations that must operate in real scenarios using virtual worlds.

It is worth mentioning that, during the last years, an increasing interest appeared in the research community for using synthesized data to train discriminative representations, useful for visual perception and artificial intelligence in general. We mention illustrative related works in Sect. 4. Finally we conclude this paper with Sect. 5, where we draw our main conclusions and future work.

2. Pedestrian Detection

Due to its high relevance for ADAS and self-driving vehicles, *pedestrian detection* has been one of the research topics receiving most attention since the beginning of the century [41, 57, 43, 58]. It is a difficult task difficult due to a combination of factors: pedestrians are moving objects with varying morphology, pose and clothes; there is a large variety of (urban) outdoor scenarios; on-board images are acquired from a moving vehicle so that pedestrians can be seen from different viewpoints at a range of distances, under uncontrolled weather and illumination.

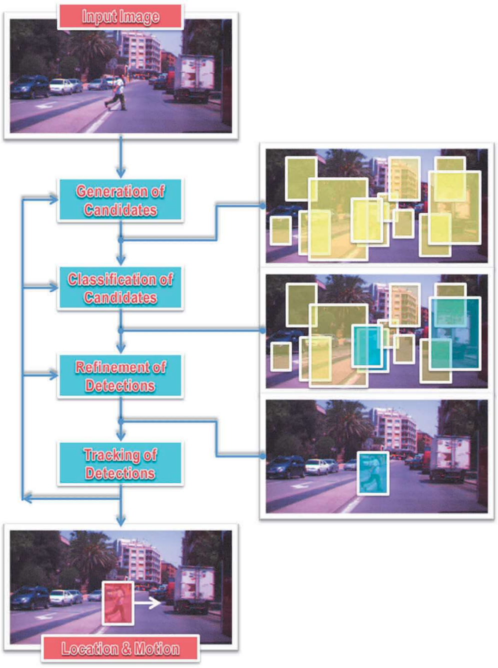


Figure 3: Common pedestrian detection pipeline.

In this context, the most common processing pipeline was the one conceptualized in Fig. 3, which in fact is also valid for other objects such as cyclists and vehicles. Briefly, the *generation of candidates* mostly consists in using a sliding window for the different layers of an image pyramid that accounts for the desired range of detection distances. The *classification of candidates* must determine if each candidate corresponds to a pedestrian or not (*i.e.* classify each candidate window as pedestrian or background). The *refinement of detections* can discard some false positives by incorporating complementary criteria such as temporal coherence (based on the tracking of the detections). In addition, this module applies non-maximum suppression to resume the multiple detections coming from the same pedestrian into just one. Finally, the *tracking of detections* allows to predict the location of pedestrians in next frame and, thus, their moving direction in the scene.

The most critical module is the *classification of candidates*. For performing such a task, an object-discriminative representation must be learned from a set of annotated samples. In this case, collections of images containing pedestrians annotated with BBs are typically used for learning the desired representation. Therefore, as Fig. 4 illustrates, our aim was to learn such a representation using virtual worlds, and then just plugging it in the pedestrian detection pipeline to operate in real scenarios.

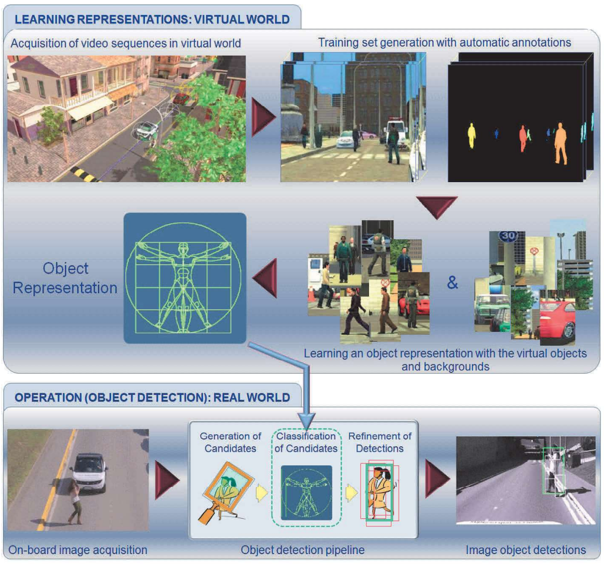

LEARNING REPRESENTATIONS: VIRTUAL WORLD
Acquisition of video sequences in virtual world
Training set generation with automatic annotations
Object Representation
&
Learning an object representation with the virtual objects and backgrounds
OPERATION (OBJECT DETECTION): REAL WORLD
Generation of Candidates
Classification of Candidates
Refinement of Detections
On-board image acquisition
Object detection pipeline
Image object detections

Figure 4: Development and testing of an object (pedestrian) detector. Top: training object representations in virtual worlds to obtain automatic annotations (*ground truth*). Bottom: using the representation to operate in real scenarios.

2.1. Straightforward proof-of-concept

The classification of candidates consists on three main ingredients: (1) a set of designed visual *appearance* (shape and texture) descriptors (*e.g.* Haar, HOG, LBP, Chanel features)–of course, *motion* descriptors are also used but we skip them here for the sake of simplicity–; (2) an overall object *model* (*e.g.* holistic, patch-based, deformable part-based); (3) an object *classifier* learned in the space of the selected visual descriptors (*e.g.* SVM-based, AdaBoost, Random Forest). For a comprehensive review please refer to [59]. In modern end-to-end learning all these ingredients are integrated in the form of deep CNN.

Since visual appearance of virtual and real worlds is correlated but different, thus being the most probable cause for a potential failure of our idea, we started by focusing on appearance descriptors and a simple, but effective, combination of model and classifier. In particular, assuming a holistic model and linear SVM, we used the widespread HOG descriptor, *i.e.* basically the algorithm proposed in [22]. In [60] we elaborated this idea and presented the obtained results according to the so-called Daimler real-world dataset [41]. In Fig. 5 we reproduce the quantitative accuracy curves from which we drew our conclusions[5]. We can see that, following the same training protocol for fair comparison, the obtained average miss rate when testing in the Daimler testing set, is even slightly better if we train the pedestrian classifier using a virtual world than if we train it with the training set of Daimler.

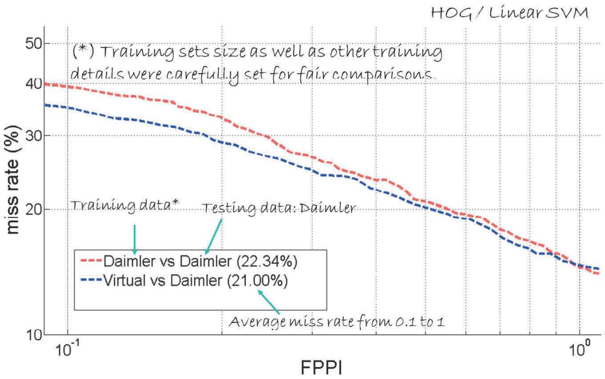


Figure 5: Detection curves plotted according to the false positives per image (FPPI) *vs* miss rate (percentage of non-detected objects). The average miss rate is shown in parenthesis. 'Dataset-1 *vs* Dataset-2' means that we trained the pedestrian classifier with the training set of 'Dataset-1', and tested it with the testing set of 'Dataset-2'.

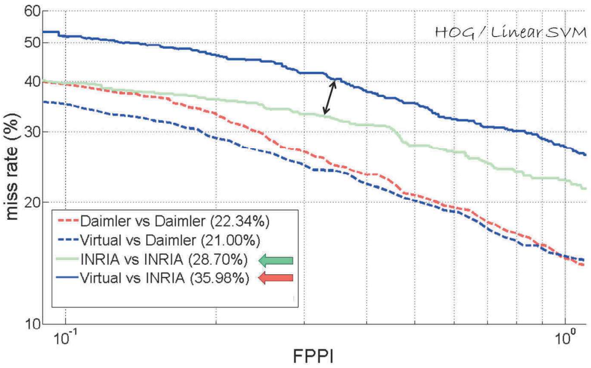


Figure 6: When training with virtual-world pedestrians and testing in INRIA testing set, there is a meaningful loss of detection accuracy with respect to the training based on the INRIA training set.

For these experiments we used the Half-Life 2 video game [61], which allowed us to import pedestrians and city maps created with third-party graphic editors, as well as to add modifications. One of those modifications, done by the company ObjectVideo [62], allowed us to obtain pixelwise annotations for the virtual-world pedestrians (see Fig. 4–(Top,Right)), from which we obtain pedestrian and background BBs automatically.

Overall, for the question *'can we learn pedestrian appearance in virtual worlds and use it successfully in real scenarios for pedestrian detection?'*, the answer was *'yes'*.

2.2. The domain adaptation problem

After this initial proof-of-concept we evaluated the same training-testing paradigm with more datasets. In particular, we used another popular dataset for pedestrian detection at that moment, INRIA [22]. As can be seen in Fig. 6, in this case there was a relatively large drop on the detector accuracy when training with virtual-world data. The first question then was if the gap was due to the fact of using virtual *vs* real data for training and testing, respectively; or if the underlying reason was more generic, *i.e.* if the problem is due to the fact of training and testing with images acquired with really different sensors and scenarios.

[5] These plots are from a posterior improved version of our detector.

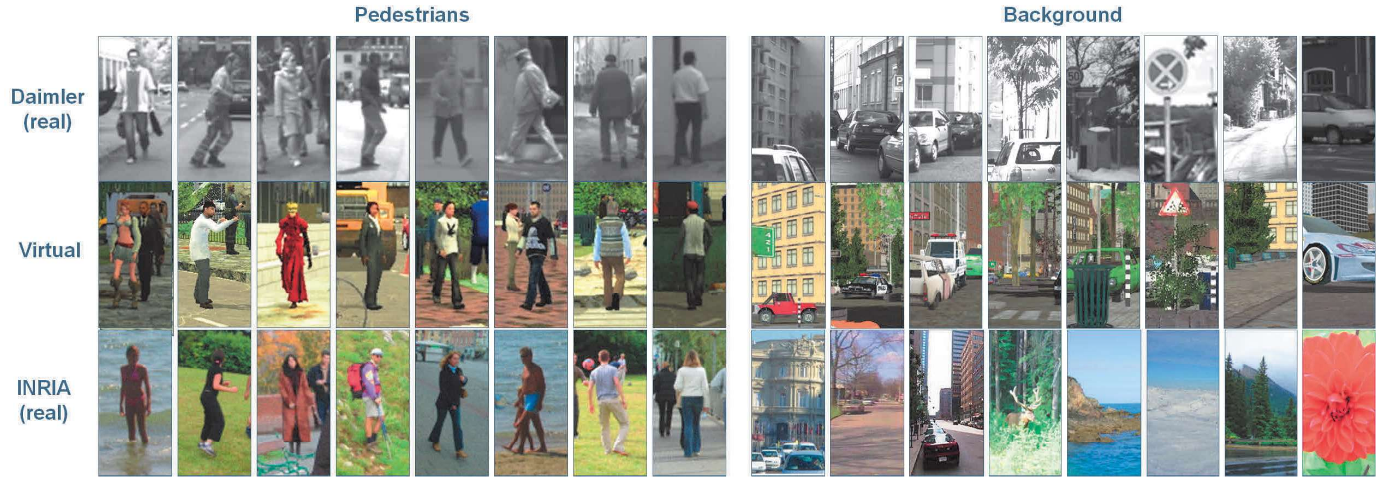


Figure 7: Pedestrian and background windows from Daimler, virtual, and INRIA datasets.

In Fig. 7, we see different pedestrian and background samples from virtual, Daimler and INRIA datasets. The main differences are the following:

- *Environment.* Daimler and virtual-world images are acquired in urban environments, while INRIA consists of photographs done not only in urban environments but also in countryside and beach scenarios.
- *Sensor.* The INRIA dataset consists of color photographs based on a single-sensor Bayer pattern, the Daimler dataset is based on a monochrome camera, and for the virtual-world dataset we have three spatially aligned virtual sensors (R,G,B).
- *Focus.* Since INRIA dataset is based on photographs, each image was focused independently of the others, so there is variability on the focus distance, many times focusing on the persons within the photos; while Daimler dataset is based on a fixed focus distance and virtual-world dataset did not considered distance-dependent blurring at that time.
- *Pedestrians.* The typical pose and viewpoint of the pedestrians is more similar for Daimler and virtual-world datasets than for INRIA; in the latter case pedestrians are sometimes even posing for the photo.

Hence, we hypothesized that the underlying problem is due to the fact of training and testing with images acquired with different sensors and in different scenarios. For validating this hypothesis we investigated the case of training with Daimler training set and testing in INRIA testing set [63]. The results are shown in Fig. 8. Indeed, the drop in accuracy is similar to the case of using virtual-world data for training. Thus, we cast the accuracy drop as a *domain adaptation* problem [64]; *i.e.* we assume that, in general, training and testing with datasets of different origin provokes a drop on detection accuracy. Consequently, training with virtual-world data and testing in real scenarios is just a particular case of such a more general problem.

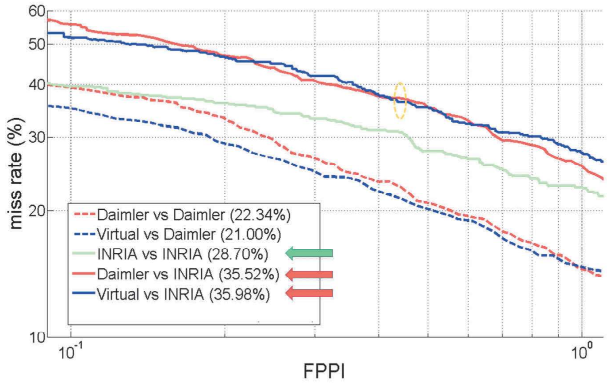


Figure 8: *Domain adaptation* problem: training and testing with datasets of different origin provokes a drop on detection accuracy.

2.3. Formulation of the domain adaptation problem

Let $\mathcal{D}_s$ and $\mathcal{D}_t$ be two different domains from which we take samples. $\mathcal{D}_s$ is the *source* domain and $\mathcal{D}_t$ is the *target* domain. Given a sample $x_t \in \mathcal{D}_t$, we want to know if $x_t \in w_t$, using w_t to denote the samples in $\mathcal{D}_t$ with a particular property in which we are interested in. We want to face this problem by learning a classifier $\mathcal{C}$ able to answer if $x_t \in w_t$. To learn $\mathcal{C}$ we want to follow a discriminative paradigm, *i.e.*, learning from annotated samples. If $x_t \in \mathcal{D}_t$, its corresponding label ℓ_{x_t} equals $+1$ if $x_t \in w_t$ and -1 otherwise. It turns out that we have very few annotated samples drawn from $\mathcal{D}_t$ as to learn a reliable classifier. However, either we have sufficient annotated samples drawn from $\mathcal{D}_s$ or we already learned a classifier $\mathcal{C}_s$ (of the same type than $\mathcal{C}$) using such data. If the distributions of the samples in $\mathcal{D}_s$ and $\mathcal{D}_t$ are uncorrelated, then domain adaptation is not possible. However, if they have a sufficient correlation, then we are facing a problem of *supervised domain adaptation* [65]. Basically, we can apply a *feature-transform-based* method or a *model-transform-based* one to address it. In the former case, we can use the large amount of annotated data from $\mathcal{D}_s$ and a low amount

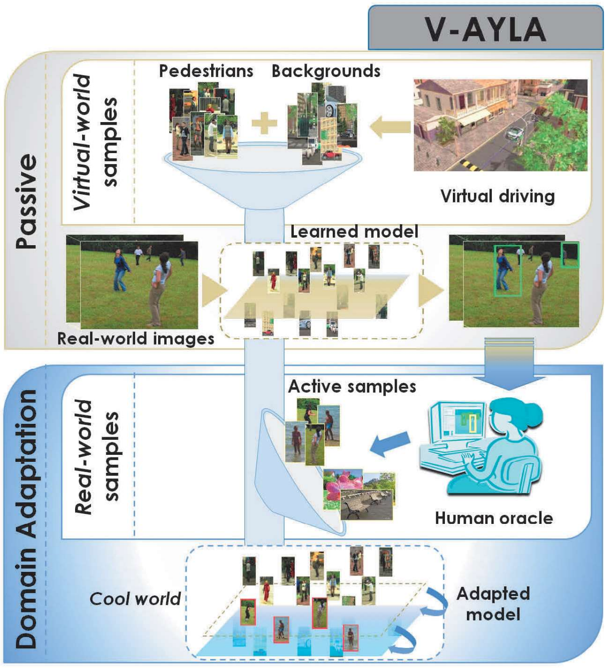


Figure 9: V-AYLA: Virtual-world Annotations, Yet Learning Adaptively. In this case, active learning is applied to collect target-domain informative samples (here from real scenarios) to be used together with the source-domain one (here from virtual world) to train the desired domain-adapted classifier. The *cool world* is the feature space where virtual- and real-world samples are together for the training of the domain-adapted classifier.

of annotated data from $\mathcal{D}_t$ to learn a $\mathcal{C}$ with chances of succeeding in the task of classifying unseen samples from $\mathcal{D}_t$. In the latter case, we use such annotated data from $\mathcal{D}_t$ to adapt $\mathcal{C}_s$ for obtaining the desired $\mathcal{C}$.

Roughly speaking, $\mathcal{D}_s$ is the set of image windows cropped from virtual-world images, and our $\mathcal{D}_t$ the set of image windows cropped from the real-world images in which we want to detect pedestrians. A sample x_t is just the descriptor of an image window (*i.e.*, the *features* in the machine learning language), w_t is the property of imaging a pedestrian (*pedestrian* class), and $\mathcal{C}$ a pedestrian classifier.

2.4. Domain adaptation solutions

2.4.1. Holistic models

We started by using *feature-transform-based* methods for holistic pedestrian representations. In [63, 66, 67] we defined the V-AYLA[6] (*Virtual-world Annotations Yet Learning Adaptively*) framework. It consists of:

- A criterion for manually annotating objects (pedestrian BBs and pedestrian-free-image flags) in the target domain (*i.e.* the real-world scenarios). The aim is that, thanks to the source-domain annotated data (*i.e.* the virtual-world data), we can significantly reduce the quantity of manual annotations required to have a competitive classifier.

- The space of image descriptors (feature space) where the target-adapted classifier is learned, *i.e.* where the virtual- and real-world descriptors are combined. We termed this space as *cool world*[7].

In [66] we presented experiments for HOG, LBP and HOG+LBP image descriptors, using linear SVM.

We tested two different cool worlds. One corresponds to the original (ORG) feature space where samples from both words are just combined; in other words, the SVM learning procedure does not distinguish between real and virtual-world samples. The other cool world is the augmented (AUG) feature space proposed in [68] for natural language processing. Assume that $\mathbf{x}^j \in \Re^n$ denotes the image descriptor of a *j-th* sample (*e.g.* a HOG vector); if the sample comes from the virtual world (source), let us term it as $\mathbf{x}^j_v$, while if it comes from real world (target) we call it $\mathbf{x}^j_r$. Then, in the AUG space $\mathbf{x}^j_v$ is represented by the vector $< \mathbf{x}^j_v, \mathbf{x}^j_v, \mathbf{0} > \in \Re^{3n}$, being $\mathbf{0} \in \Re^n$ a vector of zeros. Analogously, $\mathbf{x}^j_r$ is represented by $< 0, \mathbf{x}^j_r, \mathbf{x}^j_r > \in \Re^{3n}$ in AUG space. The idea of AUG space is to exploit commonalities of source and target domains (note how the sub-space $< \_, \mathbf{x}, \_ >$ is common for real and virtual data) at the same time than their differences ($< \mathbf{x}_v, \_, \_ >$ separated from $< \_, \_, \mathbf{x}_r >$), when learning the SVM classifier.

We tested two criteria for annotating real-world data. One corresponds to doing a prefixed quantity (N) of random annotations in the real-world training images, and marking pedestrian-free images. The other criterion consists in applying *active learning* [32], *i.e.* the human oracle annotates only pedestrians which have not been detected by a previous version of the detector (initially based only on the virtual-world data) and eventually pointing out false positives too (Fig. 9).

Active learning together with the AUG cool world was the best option when $N \sim 10\%$ of the available real-world training pedestrians; *i.e.* domain adaptation is achieved. However, as can be seen in Fig. 10, the combination of random annotations with the ORG cool world also achieves domain adaptation with a relatively low number of annotated pedestrians ($N \lesssim 25\%$). In fact, for higher percentages of target-domain pedestrians, ORG performs better than AUG. We remark here that in these experiments the 100% of real-world training pedestrians approximately matches the number of virtual-world training pedestrians.

[6] *AYLA* wants to evoke the main character (a Cro-Magnon women) of the popular saga *Earth's Children* by *Jean M. Auel*. *Ayla* is an icon of robustness and adaptability. During her childhood she is educated by Neanderthals (*the clan*), the physical appearance of them corresponds to *normal humans* for her. However, somehow, she recognizes Cro-Magnons as humans too first time she met them during her youth. *Ayla* adapts from Neanderthals customs to Cro-Magnons ones, keeping the best of both worlds, which is the spirit with our V-AYLA paradigm. Anecdotally, it turns out that there is a popular videogame that incorporates *Ayla* as character.

[7] *Cool world* term is a tribute to the movie with that title. In it, there is a real world and a cool world, in the latter, real humans and cartoons live together but it is mainly populated by cartoons, *i.e.* as in our experiments.

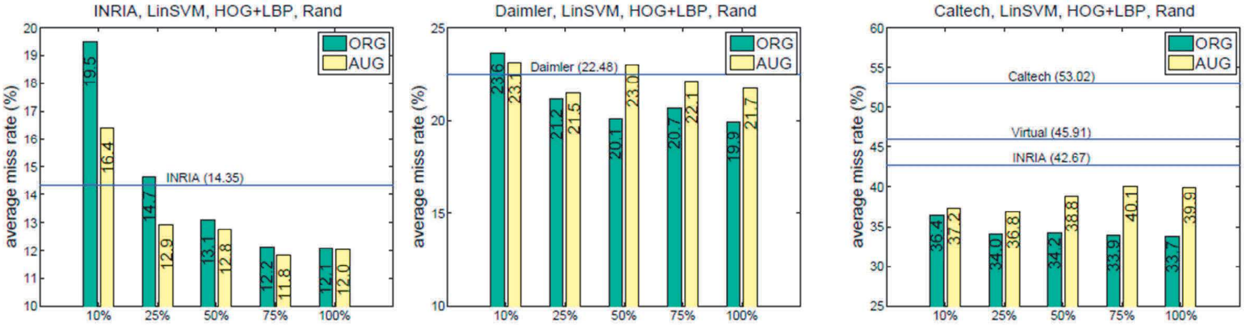


Figure 10: V-AYLA results testing on different datasets (INRIA, Daimler, Caltech) for the case of random annotation of N target-domain pedestrians, focusing on the pedestrian detector based on the HOG+LBP image descriptors (ORG and AUG cool worlds) and linear SVM. We test $N \sim 10\%$, $N \sim 25\%$, $N \sim 50\%$, $N \sim 75\%$ and $N \sim 100\%$, where these percentages are with respect to the total number of available training pedestrians of each target-domain dataset (INRIA, Daimler, Caltech). The vertical line with the average miss rate in parentheses corresponds to using the 100% of the target-domain training pedestrians without taking into account any information from the virtual world. In the case of Caltech we also show with vertical lines the result of training using only the INRIA and only virtual-world datasets.

2.4.2. Part-based models

After obtaining these evidences about the reliability of domain adapting virtual and real worlds for pedestrian detection, the natural following step was two-fold. On the one hand, assessing the use of virtual worlds and domain adaptation for richer object representations. On the other hand assuming more challenging conditions. In particular, we considered the widespread deformable part-based model (DPM) [23], and we restricted ourselves to *model-transform-based* methods, *i.e.* adapting the DPM without visiting again the source-domain data. The latter restriction leads to adaptation methods which are faster because we do not need to revisit the source-domain data (here the virtual one), and more general since some times we may not have access to the original (source-domain) data (*e.g.* due to confidentiality reasons) but we may have access to the model resulting from training with such data.

DPM describes *appearance* and per *component* information of *deformation* (Fig. 11). Components account for different BB aspect ratios of the annotated objects (*e.g.* frontal/rear-viewed pedestrians *vs* left/right-side views) as a proxy for the viewpoint under which such objects were imaged. Thus, each object (*e.g.* pedestrian) sample contributes to the training of one component. Such component-based data clustering allows to learn more accurate models than by just mixing all training samples.

Then, given a component, the DPM training focuses on appearance and deformation parameters. It is assumed that annotations are only available as full object BBs, *i.e.* there are not annotations for the parts (latent information). Thus, the training must apply a coordinate-descent method, *i.e.* first it is assumed that the location of each part is known and only the appearance of the parts must be learned, then the appearance of each part is supposed to be known and the best location matching the appearance is learned per part. Locations are initialized for uniformly covering the full object BB.

Since initialization is key for convergence and such rule

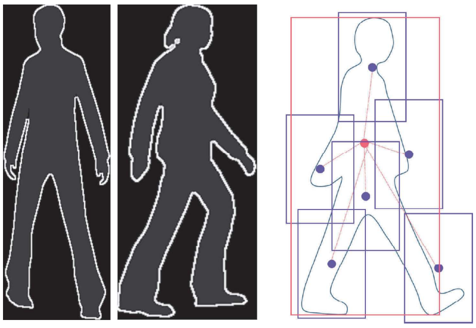


Figure 11: DPM structure. Left: pedestrian BBs with different aspect ratios, so eventually these samples would be used to train different *components* of the DPM. Right: given a component, example of a DPM based on six parts (*i.e.* sub-windows of equal size) plus the root (the whole pedestrian window). The root and each part convey *appearance* information (HOG-inspired) which is learned. DPM is based on a *star* model, so the location of each part is referenced with respect to the root. The plausible locations of the parts are also learned and encoded as *deformation* parameters.

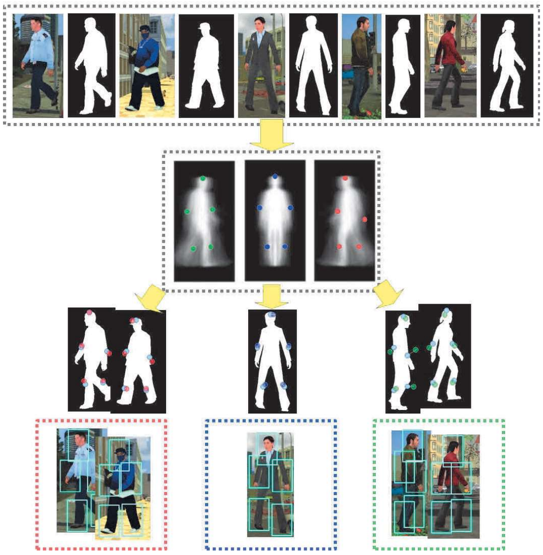


Figure 12: Virtual-world pedestrians can be clustered according to their imaged silhouette and we can have annotated semantic parts.

of thumb can be sub-optimal, in [69] we used virtual-world privileged information to improve this stage. In particular, we relied on the silhouette of the pedestrians (available thanks to the pixelwise nature of the virtual-world annotations) for defining components (measuring silhouette similarity with the Chamfer distance [70]) and providing BBs for semantic parts (Fig. 12). Thus, in terms of the DPM training, there was not latent information. We termed this DPM variant as virtual-world DPM (V-DPM). In [69] we showed that starting with V-DPM and refining it with all the available real-world training data provided a more accurate DPM than just using the real-world training data with the standard initialization. Thus, our learning procedure consisted of a pre-training on virtual-world data with privileged information, followed by a *fine tuning* with real-world data (*i.e.* without revisiting the virtual-world data).

However, we aim at using as few annotated real-world objects (pedestrians) as possible. In particular, when experimenting with different datasets, we set the constrain of using only the $\sim$ 10% of the available training data (*i.e.* simulating the saving of 90% of training annotations) to adapt V-DPM for operating in real world; which implied to design new domain adaptation methods for DPM beyond the mentioned fine-tuning. To address this question we saw a DPM as a particular case of a structural model, where the components and parts define the structure. Therefore, the learning of a DPM can be addressed through the *structural SVM* (SSVM) paradigm with latent variables (during the adaptation) [71, 72, 73].

Accordingly, in [74] we proposed the adaptation procedure drawn in Fig. 13. The figure shows the adaptation of a DPM-based pedestrian detector from a virtual-world (source domain) to real scenarios (target domain). As domain adaptation module we proposed the *adaptive structural* SVM (A-SSVM) and the *structure-aware* A-SSVM (SA-SSVM). A-SSVM is a straightforward extension of the Adaptive SVM (A-SVM) proposed in [75]. A-SSVM does not take into account the inherent structure of the SSVM (*e.g.* either the parts or the components in our case) during the adaptation procedure. SA-SSVM takes this into account and shows a better adaptation than A-SSVM. Both A-SSVM and SA-SSVM require target-domain annotated samples (*e.g.*, a few pedestrians and background) that can be provided by a human oracle. As alternative and/or complement, we proposed a self-annotation strategy inspired by *self-paced learning*. The experiment conducted in different datasets (ETH, Caltech, CVC) showed that the SA-SSVM with an oracle was the best performing method. SA-SSVM improved from 7 to 15 points (percentage units) the V-DPM accuracy; *e.g.*, in the CVC dataset the miss-rate of V-DPM was 45.87%, and its domain adapted version using SA-SSVM was 30.75%. On the other hand, A-SSVM also showed domain adaptation capabilities since it was just around 2 points than SA-SSVM in the different datasets. Similarly, the self-annotation procedure also allowed to do domain adaptation; with it and SA-SSVM the improvement over V-DPM ranged from 4 to 10 points.

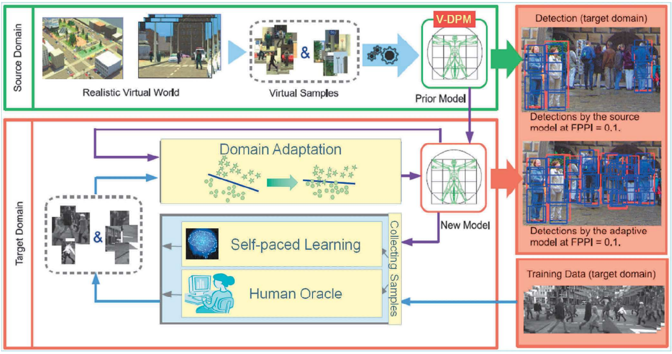


Figure 13: Proposed framework for domain adaptation (DA) of DPMs. The figure shows the adaptation of a DPM-based pedestrian detector from a virtual-world (*source domain*) to real scenarios (*target domain*). As DA module we proposed the *adaptive structural* SVM (A-SSVM) and a the *structure-aware* A-SSVM (SA-SSVM). A-SSVM and SA-SSVM require target-domain labeled samples (*e.g.*, a few pedestrians and background) that can be provided by a human oracle. As alternative and/or complement, an strategy inspired by *self-paced learning* can be used for the automatic labeling of samples in unlabeled or weakly labeled target domains.

Since DPM was the best performing model for object detection before the irruption of deep CNNs and given these positive results, we explored more domain-adaptation ideas for such a model. In particular, we proposed to exploit eventually existing tree-like hierarchical relationships in the target domain [76]. At testing time, each node of the tree contains a DPM, from node to node the weights of the DPM change but not the structure (components and parts). At training time, the intuitive idea consists of using V-DPM as source model for the root of the tree (target model), while, in turn, the DPM of each father node acts as source model for the corresponding children. All node-to-node adaptations are performed simultaneously with A-SSVM, but eventually using different target-domain samples at each node. The new domain adaptation procedure was termed as HA-SSVM. Following HA-SSVM with a resolution-based hierarchy, we won the 1st Pedestrian Detection Challenge of the KITTI benchmark[8]. In order to adapt our V-DPM, we used 200 pedestrians of the KITTI training set, roughly the 11% of the available ones, as well as 2,000 pedestrian-free images of the 7,518 available for training. To the best of our knowledge, this was the first time that an object detector domain adapted from a virtual world won such a challenge.

Encouraged by the good results of HA-SSVM and the above mentioned self-annotation proof-of-concept used with SA-SSVM, in [77] we researched domain adaptation for unlabeled videos. The idea is summarized in Fig. 14 for pedestrian detection. Given a target domain sequence, we first apply the source domain detector (V-DPM) to collect the detected bounding boxes (red boxes). Then, multiple object tracking (MOT) is used to generate trajectories

[8] Results were reported during the Reconstruction Meets Recognition Challenge (RMRC) held in conjunction with the ICCV2013, see `ttic.uchicago.edu/∼rurtasun/rmrc/program.php`

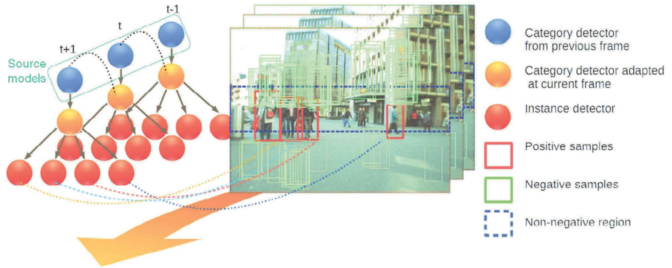


Figure 14: HA-SSVM applied for online domain adaptation, frame by frame, with self-annotated target-domain pedestrians. The domain-adapted DPM at frame t (the one at the root of the three-level tree) is used as source model for the domain adaptation at frame $t + 1$.

and also remove some false positives. After generating the trajectories, our hierarchical model can be learned frame by frame using online A-SSVM. At each frame, the hierarchical model consists of instance models at the leaf nodes (red balls in Fig. 14) and a category model at the second layer (the orange ball). The adaptation is executed in a progressive manner, *i.e.*, the category model (orange ball at time t), is adapted from the previously adapted category model (blue ball at time t, adapted from orange ball at time $t - 1$). At $t = 0$, the category model is initialized by the source domain model (V-DPM). The instance models are adapted concurrently with the category model of the current frame. The hierarchical model is constructed dynamically according to the trajectories at current frame, *i.e.* each instance model corresponds to one trajectory. Each instance model is essentially an exemplar classifier which is trained using only one positive example (red BBs) and many negative examples (green BBs). The negative examples are collected from the same frame, but outside the *non-negative* area. The non-negative region (blue dash rectangle) is defined using prior geometry knowledge, which can avoid accidentally introducing false negatives as background examples during training.

The final domain-adapted pedestrian detector corresponds to the DPM of the root node at the last frame of the training sequence. In fact, if the on-board sequences are acquired with an associated world location (*e.g.* using a consumer-grade GPS), then we could link intermediate adapted models to world areas. Thus, once the car visits again one of such areas we can apply its specifically associated detector.

2.4.3. Patch-based models

In parallel to our work with DPMs, we were developing our own object model. In particular, we proposed a *random forest of local experts* (RF-LE) which demonstrates a very competitive performance [24]. This model consists of a set of binary decision trees that form the forest. Given a candidate window, each tree provides a probability of containing the modeled object or background. The final decision is taken by thresholding the average probability over all the trees. To obtain the probability of a particular tree, the candidate window follows a path of decisions from the root to a leaf node. Leaf nodes encode the object *vs* background probabilities. Node decisions are taken according to a local expert, which decides whether to route the candidate window to the left or the right child. The local expert used in [24] considers only a rectangular sub-window, that we call *patch*, of the whole candidate window (*i.e.* a sort of part), and the decision is taken according to the HOG+LBP descriptor of such a patch and a linear SVM classifier. Therefore, we can think that each leaf encodes the probability of being object or background given a particular configuration of patches, *i.e.* those analyzed in the path to the leaf. Fig. 15 shows a simple visual example. These patches are not manually predefined, instead they are learned at training time; which is also the case for the node SVMs, and each leaf probability distribution.

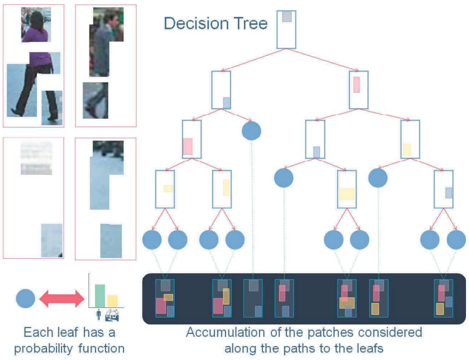

Decision Tree
Each leaf has a probability function
Accumulation of the patches considered along the paths to the leafs

Figure 15: Random Forest (set of binary decision trees) of Local Experts. To determine if a candidate window contains an object (*e.g.* pedestrian) or background, many times is enough to check a subset of well-selected sub-windows (patches). In a given decision tree, the root and each intermediate node focus on a patch, routing the search towards the left or the right child node according to the appearance of the patch (*e.g.* based on HOG+LBP/Linear-SVM). Each leaf node contains a probability distribution to classify the candidate window as object or background. Intuitively, since each leaf is reached trough a path from the root, the classification at the leaf is based on all the visited patches. The location of the patches and their appearance classifier, as well as the probability distributions, are learned. The classification at the Random Forest level is performed by thresholding the average of the probabilities of being object obtained from the decision trees.

Since in [67] we showed that the approach based on the ORG cool world together with active learning was able to domain adapt (from virtual to real worlds) an AdaBoost-based classifier applied to Haar+EOH image descriptors [78], which is similar to a patch-style classifier, we decided to investigate domain adaptation methods for our RF-LE. In [79], we presented three different approaches based on: adapting the individual nodes of each tree, adapting the different root-to-leaf paths of the trees, and adapting the forest itself by changing full trees as as sort of *reforesting* the forest mechanism. The reforesting procedure was the simplest to program, provided the most accurate adapted classifiers, and was the fastest in terms of domain adaptation computing time. The method consists of the following steps. We start with a RF-LE build using only virtual-world data, let us call it V-RF-LE. Then, with the few annotated samples (pedestrians) available from the target domain (real-world) we build another RF-LE of C trees, let us call it R-RF-LE, where C is a hyper-parameter set by a hold-out validation dataset. The reforesting procedure used in [79] consisted on randomly removing C trees from V-RF-LE and replace them by the trees of R-RF-LE. The resulting RF-LE is the domain adapted one. Experimental results (INRIA, KITTI, Daimler) showed that the domain adapted classifier improves from 5 to 10 accuracy points over V-RF-LE, and similarly over R-RF-LE.

2.5. Conclusion

The experiments summarized in this section lead us to conclude that virtual-world data is effective for training vision-based pedestrian detectors which can be adapted to operate in real scenarios. The different adaptation procedures have shown to provide adapted detectors that improve those trained only on virtual data, as well as those trained using only the real-world data available for the adaptation (which we constrained to save a $\sim 90\%$ annotation effort along our experiments). These observations hold for holistic, part-based and patch-based models. Moreover, although we have focused only on pedestrian detection as proof-of-concept, we believe that the results may be extrapolated to vision-based object detection in general. For instance, recently we focused on DPM-based onboard vehicle detection [56] as use-case to investigate the role of photo-realism in the virtual-to-real domain gap. We used KITTI dataset as target domain and different virtual-world datasets as source domains, namely Virtual KITTI [80], GTA-V videogame [81], and our SYNTHIA [82] (see Sect. 3 for more details on SYNTHIA). Virtual KITTI and SYNTHIA show similar degree of photo-realism, being different in the type of rendered environment, where SYNTHIA presents a more urban style. GTA-V shows urban style too, being impressively photo-realist. The conclusions drawn from [56] when using SA-SSVM are the same than the summarized here for pedestrian detection. Moreover, it is shown that the achieved domain adaptation is equally good for SYNTHIA and GTA-V, but worse for Virtual KITTI, suggesting that giving a reasonable degree of photo-realism (as Virtual KITTI and SYNTHIA have) the domain gap depends more on the content of the images than on their photo-realism.

3. Semantic Segmentation

A visual task equally relevant for ADAS and autonomous driving is on-board semantic segmentation; *i.e.* given a set of classes of interest, being able to classify each image pixel as belonging to one semantic class or another. Therefore, we decided to challenge our paradigm (virtual-world data + domain adaptation) by addressing such a very difficult task. Moreover, remind that from the annotation perspective, the benefit of using automatic annotations would be much higher than for BB-based object detection since, to perform semantic segmentation, the required annotations are based on the silhouette of the classes of interest (*i.e.* the class frontiers in the images). Moreover, the best performing semantic segmentation methods nowadays are based on deep CNNs, which are representations learned end-to-end; thus, eventually relying on millions of parameters and so especially demanding in terms of training data.

In order to address this new challenge, we generated a new virtual city from which we obtained more than 320,000 pixelwise automatically annotated images (class ID and instance ID), also with depth information (distance from the virtual camera) attached to each pixel. We have named this city as SYNTHIA[9], which stands for *SYNTHetic collection of Imagery and Annotations* (Fig. 16). SYNTHIA includes static infrastructure (road, sidewalks, buildings, vegetation, poles, fences, traffic signs, lane markings, sky) and dynamic objects (pedestrians, cyclists, vehicles, clouds). Different outdoor illumination is generated as well as the four year seasons. Moreover, all the elements of the scene have a large amount of variability (*e.g.* pedestrian clothes, vehicle colors, etc.). All this information is integrated through the Unity®Pro framework[10].

As first work with SYNTHIA [83] we trained the deep CNN for semantic segmentation described in [82], referred to as T-NET, which combines convolutional and counterpart deconvolutional layers, and it is especially interesting because its relatively low memory footprint (critical for on-board embedded systems). We selected a sequence of 13,400 SYNTHIA images and considered their corresponding pixelwise class annotations, we call this data SYNTHIA-Rand. During the training of T-NET, we employ the Balanced Gradient Contribution (BGC) introduced in [82]. It consists in building batches with images from both domains (virtual and real), given a fixed ratio. Experimentally we found that the ratio of 60% of the images being from the real scenarios and 40% from SYNTHIA-Rand was giving the best results. Note that this does not mean that overall we assume more images coming from the real scenarios than from SYNTHIA-Rand, it is just that the same real-world images eventually will be used in more batches than the images of SYNTHIA-Rand. In fact, in our experiments the number of annotated real-world data is significantly lower than SYNTHIA-Rand (Tab. 1). Note also that here we are mixing the data, following the spirit of the *cool world* (ORG) idea (*i.e.* our starting point also for object detection). In here, however, due to the mentioned 60/40 ratio, real-world images dominate the distribution, while virtual-world images are used as a sophisticated regularization term. Thus, statistics of both domains are considered during the whole procedure, creating a model which pursues to be accurate for both.

We performed semantic segmentation in the outdoor urban scenarios corresponding to the training and valida-

[9] www.synthia-dataset.net

[10] www.unity3d.com

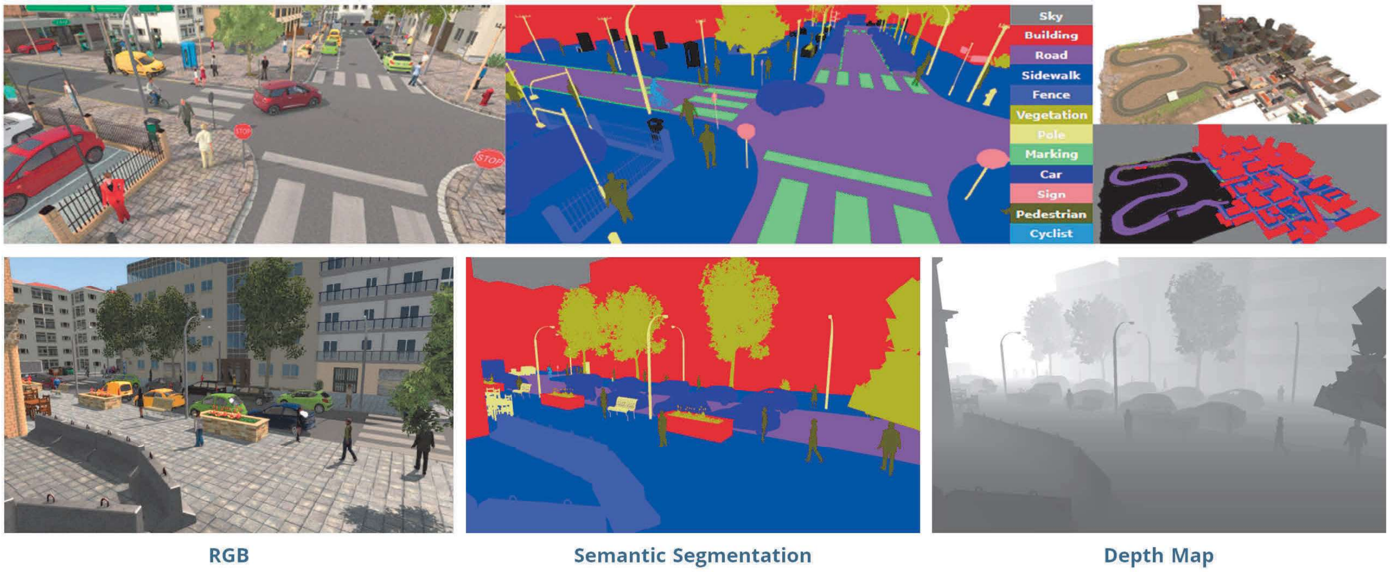


Figure 16: SYNTHIA: a large synthetic collection of imagery and annotations from a virtual city. Top, left-to-right: (1) view of a scene of the city, (2) its pixelwise annotation with the list of the considered classes, (3) general view of the city from a far viewpoint. Bottom: another scene of the city with its pixelwise annotation of both the considered classes and the depth.

Table 1: Datasets used for evaluating semantic segmentation methods, and how we split them for training and validation (testing).

Dataset	Total #Frames	Training	Validation
CamVid [52]	701	300	401
KITTI [12, 49, 50, 51]	547	200	347
Urban LabelMe [28, 82]	942	200	742
CBCL StreetScenes [84, 82]	3547	200	3347
SYNTHIA-Rand [83]	13,400	13,400	0

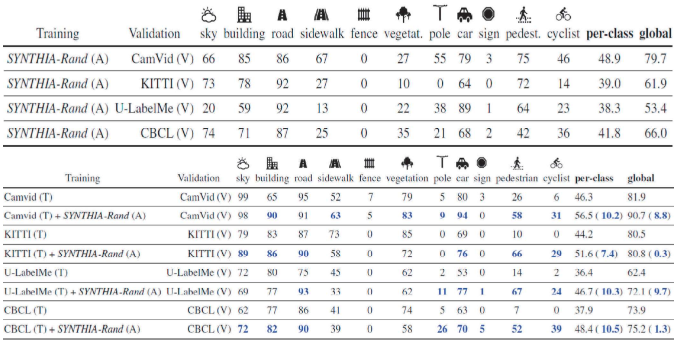
tion splits of the datasets in Tab. 1. The results are shown in Fig. 17. Note that using SYNTHIA-Rand alone produces reasonable results, but when combined with a relatively low number of real-world images (*i.e.* the training ones in Tab. 1), there is a significant gain in performance, not only with respect to the use of SYNTHIA-Rand alone, but also with respect to only using the real-world training images. The gain is more clear in the per-class metric since it is not dominated by the classes that usually have more pixels (*e.g.* sky) but which not necessarily are more relevant. In particular, we can see how there are significant gains in classes such as pedestrians and cyclist since they tend to be underrepresented in the annotated real-world data. It is also worth to mention that we are not fully exploiting all the SYNTHIA possibilities for some classes such as traffic signs and poles. This is due to the fact that in these experiments we performed a hard reduction of the input image resolutions, which is harmful for the segmentation of thin/small objects. In particular, the 960 × 720 pix resolution of the SYNTHIA-Rand images were down sampled to 180 × 120 pix for speeding-up the training process and save memory. In [85], we updated these experiments working with higher resolutions, 512 × 256 pix, and using 20,000 as part of SYNTHIA-Rand. Consequently the absolute accuracy for the same T-NET and real-world dataset was higher, being the conclusions regarding the use of the virtual-world data and domain adaptation the same than in [83]. Moreover, in [85] we also compared the strategy of using the cool world idea against the use of network fine-tuning, *i.e.* training T-NET with SYNTHIA-Rand and then fine-tuning with real-world data. The results show that cool world clearly outperformed fine-tuning. Thus, this indicates to focus research on fine-tuning-like approaches are more convenient because it would not be necessary to revisit the source data for the adaptation; *i.e.* as we did with SA-SSVM for the DPM case (in [56] we have also shown that SA-SSVM is superior to cool-world style for DPM).

Overall, we think these results show that virtual-world data is not only useful for object detection but also for semantic segmentation.

4. Related Work

Overall we think we have been among the pioneers on the use of photo-realistic computer graphics for training object detectors as well as methods for semantic segmentation, with special focus on ADAS and self-driving scenarios. Moreover, as to the best of our knowledge, we were also among the pioneers posing the virtual-to-real gap as a domain adaptation problem. Nowadays, the use of this data is more and more accepted; thus, we would like to briefly summarize here other works worth to follow.

Making use of 3D CAD models of different objects (*i.e.* rendering of backgrounds is not the focus), some computer vision problems were addressed in indoor scenarios (*e.g.* for object pose estimation and localization) and/or in datasets where objects appear as predominant



Training	Validation	sky	building	road	sidewalk	fence	vegetat.	pole	car	sign	pedest.	cyclist	**per-class**	**global**
SYNTHIA-Rand (A)	CamVid (V)	66	85	86	67	0	27	55	79	3	75	46	48.9	79.7
SYNTHIA-Rand (A)	KITTI (V)	73	78	92	27	0	10	0	64	0	72	14	39.0	61.9
SYNTHIA-Rand (A)	U-LabelMe (V)	20	59	92	13	0	22	38	89	1	64	23	38.3	53.4
SYNTHIA-Rand (A)	CBCL (V)	74	71	87	25	0	35	21	68	2	42	36	41.8	66.0

Training	Validation	sky	building	road	sidewalk	fence	vegetation	pole	car	sign	pedestrian	cyclist	**per-class**	**global**
Camvid (T)	CamVid (V)	99	65	95	52	7	79	5	80	3	26	6	46.3	81.9
Camvid (T) + *SYNTHIA-Rand* (A)	CamVid (V)	98	**90**	91	**63**	5	**83**	**9**	**94**	0	**58**	**31**	56.5 (**10.2**)	90.7 (**8.8**)
KITTI (T)	KITTI (V)	79	83	87	73	0	85	0	69	0	10	0	44.2	80.5
KITTI (T) + *SYNTHIA-Rand* (A)	KITTI (V)	**89**	**86**	**90**	58	0	72	0	**76**	0	**66**	**29**	51.6 (**7.4**)	80.8 (**0.3**)
U-LabelMe (T)	U-LabelMe (V)	72	80	75	45	0	62	2	53	0	14	2	36.4	62.4
U-LabelMe (T) + *SYNTHIA-Rand* (A)	U-LabelMe (V)	69	77	**93**	33	0	62	**11**	**77**	**1**	**67**	**24**	46.7 (**10.3**)	72.1 (**9.7**)
CBCL (T)	CBCL (V)	62	77	86	41	0	74	5	63	0	7	0	37.9	73.9
CBCL (T) + *SYNTHIA-Rand* (A)	CBCL (V)	**72**	**82**	**90**	39	0	58	**26**	**70**	**5**	**52**	**39**	48.4 (**10.5**)	75.2 (**1.3**)

Figure 17: Semantic segmentation results using the usual per-class and global accuracy. The icons refer to the considered classes. Top table: using only SYNTHIA-Rand for training. Bottom table: doing domain adaptation. Blue numbers for the individual classes indicate when the use of SYNTHIA and domain adaptation outperforms the use of only the real-world data. For per-class and global metrics, the blue number in parenthesis indicates the improvement.

in the images (*i.e.* similar to a full image classification problem). In [86], rendered 3D pedestrians are used to generate 2D mask projections with an associated skeleton state; this information is used to train a pedestrian pose estimator able to work for real-world pedestrians. The same idea of using rendered data to train pose estimators was used in [87] for hand pose estimation. In [88] a human renderer is used to randomly generate synthetic human poses for training an appearance-based pose recovery system for close human views (mainly showing the upper body). In [89], human body pose estimation in depth images uses training from rendered depth body information too. In [90], web-available 3D rendered models where used to train an stereo-based object recognition systems on-board a home robot (the proof-of-concept was done with mugs and chairs). In [91], 3D CAD models are used to learn part-based object representations, in [92] for training image retrieval systems robust to the viewpoint from which objects are captured, in [93] to teach 3D geometry to DPMs, in [94] to drive indoor scene understanding based on human designed scene descriptors, in [95] to learn view-dependent mid-level visual cues for object category detection (the *chair* class is used as proof-of-concept), and in [96] as support for object labeling with weak supervision. In [97], multicategory 2D object detectors are trained with 3D CAD models and domain adaptation based on decorrelated features. In [98], realistic cluttered room scenes are generated in parallel to the training of a deep CNN for object pose recovery based on RGB-D images. The synthetic data is complemented with a small amount of publicly available annotated RGB-D data and transfer learning is performed with them. In [99], 3D CAD object models are used for supporting the training of deep CNNs for the task of object recognition; in this work, it is investigated the sensitivity of deep CNNs to various low-level sources of variability during training (3D pose, foreground texture and color, background image and color). In [100], it is proposed a deep CNN to bridge the domain gap between textureless CAD models and typical object recognition real-world images. The proposed CNN can generate training images of a more realistic appearance by simultaneously leveraging the object shape from 3D CAD models and structured real-world images of textures. In [101], rendered views of 3D object CAD models and images of real objects are adapted (real-to-virtual in this case) in a deep CNN framework to perform instance and category object detection. In [102] 3D CAD models are used to synthesize features that correspond to novel views of objects. In [103], supported by 3D CAD object models, synthetic images are generated to train deep CNNs for object viewpoint estimation. In fact, viewpoint estimation is also the focus of [104], in this case for cars, using rendered 3D models and domain adaptation. [105] also focuses on estimating the viewpoint of cars using 3D CAD models for training with domain adaptation.

Works using rendered images/objects for training object detectors, image classifiers, and semantic segmentation algorithms can be found too; and even synthesized videos for training action recognition systems. For instance, in [106], where a template-matching-based on-board pedestrian detector is built for far infrared images using rough synthesized textureless pedestrian templates (authors report poor results due to the lack of realism in the synthetic templates). In [107], samples with silhouette delineation are obtained automatically through elaborated 3D human synthetic models and inserted in real-world backgrounds (camera-calibrated human-free driving sequences are required), with such synthetic samples fused with real-world ones, state-of-the-are human detectors were obtained. In [108], 3D RGB-D human models are used to train systems for human depth estimation and human part segmentation in real-world RGB images. Such 3D models are built from MoCap data. In fact, in [109] we can find already the first method for procedural generation of synthetic videos to train deep CNNs for human action recognition in videos. The generated videos are based on realistic physics, scene composition rules, and procedural animation techniques. The human animations are built by leveraging MoCap data, animation blending, and programmatically defined behaviors. In [110], rendered images are used for texture-based material categorization supported by domain adaptation too. In [111], a driving simulator is used to generate camera images, depth and optical flow maps and pixelwise semantic annotations, focusing on highway style roads. This data is used for training a CRF model for semantic segmentation, analyzing the effects of various combinations of features. In [112], synthetic images are used for training a scene-specific and spatially-varying pedestrian appearance model to operate on video surveillance images. In [113], CNN-based object detectors are successfully trained on virtual images for operating on video surveillance images too. In [114], the aim is to detect objects such as drones, planes and cars, especially focusing on synthesizing training images that match those provided by real-world cameras in the presence of noise. More recently we can find new works in line with our work using SYNTHIA (Sect. 3). For instance, [115, 81] demonstrate the great usefulness of synthetic images for CNN-based on-board semantic segmentation using different virtual environments for collecting training data. In [116], the usefulness of synthetic training data is demonstrated for indoor semantic segmentation based on RGB-D images.

The possibility of controlling different parameters when generating synthetic images from virtual scenarios has inspired different works where the robustness and accuracy of different image descriptors are evaluated [117, 118, 119, 120], or where the focus is on gaining understanding of learned image features [118], or robustness with respect to photo-realism is assessed for CNN-based semantic segmentation [121] (being the conclusion that a photo-realism is just needed up to some level, but beyond it domain adaptation is still needed; *i.e.* as we has mentioned before following our study in [56]). Evaluation and training methods for optical flow have been also developed with the help of precise flows obtained from rendered sequences, which is a really relevant issue since groundtruth optical flow is hard to be obtained from real images with uncontrolled multiple motions. In [122], optical flow annotated data is derived from the open source 3D animated short film SINTEL. With respect to previous real-world datasets a bunch of challenging conditions are incorporated, *i.e.* long sequences and motions, motion and defocus blur, specular reflections and atmospheric effects. In [123], scene flow is estimated by a deep CNN that combines optical flow and disparity estimation. To achieve this purpose a synthetic dataset with sufficient realism, variation, and size was used for training the deep CNN. In [124], optical flow is also generated with ground truth using synthetic sequences of driving scenarios. The sequences in the dataset vary in speed, road texture, egomotion and independent moving vehicles in the scene.

Rather than performing classical domain adaptation or transfer learning for solving specific tasks with specific models, a different approach that has attracted increasing interests consists in directly transforming the source images so that they look like the target ones but preserving the ground truth (annotations). Ideally, this approach has the advantage of being task and model agnostic. For instance, [125] follows such an idea by using generative adversarial networks (GANs) and without requiring annotations on the target domain.

Rendered training data has not been used only for image-based object detection. In [126, 127], 3D point clouds are obtained from web-available 3D object models; with them classifiers are trained to operate on 3D laser scans collected by navigating through urban environments. In this case there is also a domain gap, thus, a small set of human annotated 3D laser scans is used to perform domain adaptation.

Using virtual environments for training computer vision algorithms is a very appealing and relevant topic, but we can say the same about using them for debugging during algorithm development and, especially, for testing systems. For instance, by exploiting visually and behaviorally realistic virtual environments, [128, 129] research the realization of a fully autonomous sensor network able to cover large public areas. [62] shows the usefulness of virtual environments to design and validate people trackers for video surveillance, and [130] for evaluating background subtraction and head detection algorithms under video surveillance settings too. Even more interesting, [80] proposes an efficient real-to-virtual world cloning method. Then, the virtual world can be used as proxy to assess the performance of vision-based algorithms under different environment and image acquisition conditions of interest (*e.g.* fog). As proof-of-concept, a new multiobject tracking dataset called Virtual KITTI is created. The conducted experiments suggests that conclusions obtained on this synthetic benchmark transfer to the real world (*i.e.* the real KITTI dataset). On the other hand, the validity of synthesized lower level cues such as optical flow and stereo depth for assessing the respective computation methods has also been under research [131, 132, 133].

Finally, it is also worth to mention that virtual worlds are being used for learning high-level artificial behavior. For instance, in [134] the computer learns to play Atari games using deep and reinforcement learning. In [135], an artificial agent self-learned how to reproduce human-like behaviour while playing a first-person shooter video game (BotPrize competition). In [136], abstract scenarios made with clip art are used to learn *unwritten* common sense. For this purpose, it is argued that image photo-realism is not necessary since common sense is assumed to be encoded on semantic visual information and not at pixel level. In [137], this approach is followed also to discover semantically important features and the relation between the saliency and memorability of objects and their semantic relevance. In our main field of interest, *i.e.* self-driving, deep CNNs are used to learn end-to-end the task of driving (*i.e.* from images to direct vehicle maneuvers commands) by training in a virtual driving environment [138].

5. Conclusions and Future Work

In this paper we have reviewed our pioneering work on learning visual representations to perform computer vision tasks of interest, by using realistic virtual worlds which provide automatic and precise annotations; which are very hard or even impossible to collect by human annotators. We focused on the ADAS and self-driving context. In particular, we have studied this training paradigm for developing different pedestrian models (holistic, patch-based, deformable part-based) relying on human-designed features (HOG, LBP, Haar, EOH) and high performing classifiers (SVM, AdaBoost, Random Forest). Moreover, we have studied also how such virtual-world data can be used to train deep CNNs for the challenging task of pixelwise semantic segmentation. In all cases, the learned models have been operating in real driving scenarios.

By doing this, we realized about the domain gap between virtual and real worlds. However, we have shown that such a gap does exist also between real-world datasets provided that sensor and/or object and environment conditions significantly change from training to testing time. Therefore, we proposed to cast the visual domain gap between virtual and real worlds as a domain adaptation problem. Accordingly, we developed both feature-transform-based and model-transform-based methods for eliminating the gap, always keeping in mind to reduce as much as possible human annotation.

Overall, we can affirm that we have driven a long route of research for training our car to see using virtual worlds. As use to happen, such driving has been just the start of the journey towards a horizon of new possibilities offered by the use of realistic virtual worlds. Many things remain to be done in the rest of the journey. So far, human annotation still can be required for validating computer vision algorithms. Therefore, more research should be done for using the virtual environments for testing; in line with [80], or by using circular evaluation procedures [139]. From the model training perspective, we would like to focus on self-annotating full video sequences with incremental/on-line procedures that rely on initial virtual-world-based visual models. In addition, we are very interested in cross-image modality domain adaptation in line with the proof-of-concept we did in [47] for adapting our virtual-world-based pedestrian detector to operate on FIR images. Learning to act/drive relying on deep learning and reinforcement learning as in [140] is also a topic we would like to research using virtual, but realistic, driving scenarios. Finally, we are obviously interested in researching better domain adaptation methods for deep models as well as image-to-image transformations to convert virtual-world images into real-world ones, *i.e.* as a task and model agnostic domain adaptation procedure.

Overall, the large amount of related works presented in the last years, here reviewed, is a clear sign of the increasing relevance of virtual environments for training and testing intelligent systems.

References

[1] The Institute, IEEE, Smart cities (June 2014).

[2] J. McCall, M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, International Journal of Automotive Technology 7 (2006) 20 – 37.

[3] A. López, J. Serrat, C. Cañero, F. Lumbreras, T. Graf, Robust lane markings detection and road geometry computation, International Journal of Automotive Technology 11 (2010) 395 – 407.

[4] A. Hillel, R. Lerner, D. Levi, G. Raz, Recent progress in road and lane detection: a survey, Machine Vision and Applications 25 (2014) 727 – 745.

[5] A. de la Escalera, J. Armingol, M. Mata, Traffic sign recognition and analysis for intelligent vehicles, Image and Vision Computing 21 (2003) 247 – 258.

[6] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition, Neural Networks 32 (2012) 323 – 332.

[7] F. Zaklouta, B. Stanciulescu, Real-time traffic sign recognition in three stages, Robotics and Autonomous Systems 62 (2014) 16 – 324.

[8] A. López, J. Hilgenstock, A. Busse, R. Baldrich, F. Lumbreras, J. Serrat, Nighttime vehicle detection for intelligent headlight control, in: Proceedings of the Conference on Advanced Concepts for Intelligent Vision Systems, Juan-les-Pins, France, 2008.

[9] J. Rubio, J. Serrat, A. López, D. Ponsa, Multiple-target tracking for intelligent headlights control, IEEE Transactions on Intelligent Transportation Systems 13 (2012) 594 – 605.

[10] S. Eum, H. Jung, Enhancing light blob detection for intelligent headlight control using lane detection, IEEE Transactions on Intelligent Transportation Systems 14 (2013) 1003 – 1011.

[11] M. Buehler, K. Iagnemma, S. Singh (Eds.), The DARPA Urban Challenge – Autonomous Vehicles in City Traffic, 56, STAR: Springer Tracts in Advance Robotics, 2010.

[12] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the KITTI vision benchmark suite, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012.

[13] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Conference on Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 2012.

[14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, OverFeat: integrated recognition, localization and detection using convolutional networks, in: Proceedings of the International Conference on Learning Representations, Banff, Alberta, Canada, 2014.

[15] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015.

[16] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 72 (2016) 48 – 61.

[18] NVIDIA, http://www.nvidia.com/object/drive-px.html.

[19] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, M.-M. Meinecke, Pedestrian detection for driver assistance using multiresolution infrared vision, IEEE Transactions on Vehicular Technology 53 (2004) 1666 – 1678.

[20] S. Krotosky, M.M. Trivedi, On color-, infrared-, and multimodal-stereo approaches to pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 8 (2007) 619 – 629.

[21] S. Hwang, J. Park, N. Kim, Y. Choi, I. So, Multispectral pedestrian detection: Benchmark dataset and baseline, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015.

[22] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005.

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

[24] J. Marin, D. Vázquez, A. López, J. Amores, B. Leibe, Random forests of local experts for pedestrian detection, in: Proceedings of the International Conference on Computer Vision, Sydney, Australia, 2013.

[25] S. Walk, N. Majer, K. Schindler, B. Schiele, New features and insights for pedestrian detection, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Fran-

cisco, CA, USA, 2010.

[26] D. Vázquez, J. Xu, S. Ramos, A. López, D. Ponsa, Weakly supervised automatic annotation of pedestrian bounding boxes, in: Proceedings of the conference on Computer Vision and Pattern Recognition , Workshop on Ground Truth - What is a good dataset?, Portland, OR, USA, 2013.

[27] J. Xie, M. Kiefel, M.-T. Sun, A. Geiger, Semantic instance annotation of street scenes by 3d to 2d label transfer, arXiv:1511.03240 (2016).

[28] B. Russell, A. Torralba, K. Murphy, W. Freeman, LabelMe: a database and web-based tool for image annotation, International Journal of Computer Vision 77 (2008) 157 – 173.

[29] Amazon Mechanical Turk, www.mturk.com.

[30] L. von Ahn, R. Liu, M. Blum, Peekaboom: a game for locating objects in images, in: Proceedings of the ACM SIGCHI Conf. on Human Factors in Computing Systems, Montréal, Québec, Canada, 2006.

[31] T. Berg, A. Sorokin, G. Wang, D. Forsyth, D. Hoeiem, I. Endres, A. Farhadi, It's all about the data, Proceedings of the IEEE 98 (2010) 1434 – 1452.

[32] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Machine Learning 15 (1994) 201 – 221.

[33] Y. Abramson, Y. Freund, SEmi-automatic VIsuaL LEarning (SEVILLE): a tutorial on active learning for visual object recognition, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005.

[34] S. Sivaraman, M.M. Trivedi, A general active-learning framework for on-road vehicle recognition and tracking, IEEE Transactions on Intelligent Transportation Systems 11 (2007) 267 – 276.

[35] M. Everingham, L. V. Gool, C. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) Challenge, International Journal of Computer Vision 88 (2010) 303 – 338.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Zitnick, Microsoft COCO: Common Objects in Context, in: Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 2014.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 2009.

[38] M. Enzweiler, D. Gavrila, A mixed generative-discriminative framework for pedestrian classification, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008.

[39] R. Bala, Y. Zhaoa, A. Burrya, V. Kozitskya, C. Filliona, C. Saundersb, J. Rodríguez-Serrano, Image simulation for automatic license plate recognition, in: Proceedings of the SPIE Conference on Visual Information Processing and Communication, Vol. 8305, Burlingame, California, United States, 2012.

[40] M. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 2014.

[41] M. Enzweiler, D. Gavrila, Monocular pedestrian detection: survey and experiments, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 2179 – 2195.

[42] A. Ess, B. Leibe, L. V. Gool, Depth and appearance for mobile scene analysis, in: Proceedings of the International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007.

[43] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 743 – 761.

[44] D. Gerónimo, A. Sappa, D. Ponsa, A. López, 2D-3D based on-board pedestrian detection system, Computer Vision and Image Understanding 114 (2010) 1239 – 1258.

[45] J. Marin, D. Vázquez, J. A. A.M. López, L. Kuncheva, Occlusion handling via random subspace classifiers for human detection, IEEE Transactions on Cybernetics 44 (2014) 342 – 354.

[46] A. Gonzalez, S. Ramos, D. Vázquez, A. López, J. Amores, Spatiotemporal stacked sequential learning for pedestrian detection, in: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, Spain, 2015.

[47] Y. Socarras, S. Ramos, D. Vázquez, A. López, T. Gevers, Adapting pedestrian detection from synthetic to far infrared images, in: Proceedings of the International Conference on Computer Vision, Workshop on Visual Domain Adaptation and Dataset Bias, 2013.

[48] F. Larsson, M. Felsberg, P.-E. Forssen, Correlating Fourier descriptors of local patches for road sign recognition, IET Computer Vision 5 (2011) 244 – 254.

[49] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, International Journal of Robotics Research 32 (2013) 1231 – 1237.

[50] J. Fritsch, T. Kuehnl, A. Geiger, A new performance measure and evaluation benchmark for road detection algorithms, in: Proceedings of the IEEE International Conference on Intelligent Transportation Systems, The Hague, Netherlands, 2013.

[51] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015.

[52] G. Brostow, J. Fauqueur, R. Cipolla, Semantic object classes in video: A high-definition ground truth database, Pattern Recognition Letters 30 (2009) 88 – 97.

[53] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes dataset for semantic urban scene understanding, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[54] T. Pouli, D. Cunningham, E. Reinhard, Image statistics and their applications in computer graphics, in: Proceedings of the Eurographics, Norrköping, Sweden, 2010.

[55] W. Shao, Animating autonomous pedestrians, Ph.D. thesis, Dept. C. S., Courant Inst. of Mathematical Sciences, New York Univesity (2006).

[56] A. Lopez, J. Xu, J. Gomez, D. Vazquez, G. Ros, From Virtual to Real World Visual Perception using Domain Adaptation - The DPM as Example, Springer Series: Advances in Computer Vision and Pattern Recognition, 2017.

[57] D. Gerónimo, A. López, A. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1239 – 1258.

[58] S. Zhang, R. Benenson, M. Omran, J.H., B. Schiele, How far are we from solving pedestrian detection?, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[59] D. Gerónimo, A. López, Vision-based pedestrian protection systems for intelligent vehicles, SpringerBriefs in Computer Science, 2014.

[60] J. Marin, D. Vázquez, D. Gerónimo, A.M. López, Learning appearance in virtual scenarios for pedestrian detection, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[61] Valve Software Corporation, http://www.valvesoftware.com.

[62] G. Taylor, A. Chosak, P. Brewer, OVVV: Using virtual worlds to design and evaluate surveillance systems, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 2007.

[63] D. Vázquez, A. López, D. Ponsa, J. Marin, Cool world: domain adaptation of virtual and real worlds for human detection using active learning, in: Proceedings of the Conference on Advances in Neural Information Processing Systems, Workshop on Domain Adaptation: Theory and Applications, Granada, Spain, 2011.

[64] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, N. Lawrence (Eds.), Dataset shift in machine learning, Neural Information Processing, The MIT Press, 2008.

[65] J. Jiang, A literature survey on domain adaptation of statistical classifiers, Tech. rep., School of Information Systems, Sin-

gapore Management University (2008).

[66] D. Vazquez, A. López, J. Marín, D. Ponsa, D. Gerónimo, Virtual and real world adaptation for pedestrian detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 797 – 809.

[67] D. Vázquez, A. López, D. Ponsa, D. Gerónimo, Interactive training of human detectors, Springer, 2013, pp. 169 – 184.

[68] H. D. III, Frustratingly easy domain adaptation, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 2007.

[69] J. Xu, D. Vázquez, A. López, J. Marín, D. Ponsa, Learning a part-based pedestrian detector in a virtual world, IEEE Transactions on Intelligent Transportation Systems 15 (2014) 2121 – 2131.

[70] D. Gavrila, A Bayesian, exemplar-based approach to hierarchical shape matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1408 – 1421.

[71] C.-N. J. Yu, T. Joachims, Learning structural SVMs with latent variables, in: Proceedings of the International Conference on Machine Learning, Montreal, Quebec, 2009.

[72] A. Vedaldi, A. Zisserman, Structured output regression for detection with partial truncation, in: Proceedings of the Conference on Advances in Neural Information Processing Systems, Vancouver, Canada, 2009.

[73] L. Zhu, Y. Chen, A. Yuille, W. Freeman, Latent hierarchical structural learning for object detection, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[74] J. Xu, S. Ramos, D. Vázquez, A. López, Domain adaptation of deformable part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 2367 – 2380.

[75] J. Yang, R. Yan, A. Hauptmann, Cross-domain video concept detection using adaptive SVMs, in: ACM Multimedia, Augsburg, Germany, 2007.

[76] J. Xu, S. Ramos, D. Vázquez, A. López, Hierarchical adaptive structural SVM for domain adaptation, International Journal of Computer Vision 119 (2016) 159 – 178.

[77] J. Xu, D. Vázquez, K. Mikolajczyk, A. López, Hierarchical online domain adaptation of deformable part-based models, in: Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 2016.

[78] D. Gerónimo, A. López, D. Ponsa, A. Sappa, Haar wavelets and edge orientation histograms for on board pedestrian detection, in: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Girona, Spain, 2007.

[79] A. Mozafari, D. Vazquez, M. Jamzad, A. Lopez, Node-adapt, path-adapt and tree-adapt: Model-transfer domain adaptation for random forest (2016).
URL `arXiv:1611.02886`

[80] A. Gaidon, Q. Wang, Y. Cabon, E. Vig, Virtual worlds as proxy for multi-object tracking analysis, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[81] S. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: Proceedings of the European Conference on Computer Vision, 2016.

[82] G. Ros, S. Stent, P. Alcantarilla, T. Watanabe, Training constrained deconvolutional networks for road scene semantic segmentation, arXiv:1604.01545 (2016).

[83] G. Ros, L. Sellart, J. Materzyska, D. Vázquez, A. López, The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[84] S. Bileschi, CBCL streetscenes challenge framework (2007).

[85] G. Ros, L. Sellart, G. Villalonga, E. Maidanik, F. Molero, M. Garcia, A. Cedeño, F. Perez, D. Ramirez, E. Escobar, J. Gomez, D. Vazquez, A. Lopez, Semantic Segmentation of Urban Scenes via Domain Adaptation of SYNTHIA, Springer Series: Advances in Computer Vision and Pattern Recognition, 2017.

[86] K. Grauman, G. Shakhnarovich, T. Darrell, Inferring 3d structure with a statistical image-based shape model, in: Proceedings of the International Conference on Computer Vision, Nice, France, 2003.

[87] L. Ballan, A. Taneja, J. Gall, L. Van Gool, M. Pollefeys, Motion capture of hands in action using discriminative salient points, in: Proceedings of the European Conference on Computer Vision, Firenze, Italy, 2012.

[88] A. Agarwal, B. Triggs, A local basis representation for estimating human pose from cluttered images, in: Proceedings of the Asian Conference on Computer Vision, Hyderabad, India, 2006.

[89] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipmanand, A. Blake, Real-time human pose recognition in parts from a single depth image, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011.

[90] W. Wohlkinger, M. Vincze, 3D object classification for mobile robots in home environments using web-data, in: Proceedings of the International Conference on Cognitive System, Zurich, Switzerland, 2010.

[91] S. Satkin, M. Goesele, B. Schiele, Back to the future: Learning shape models from 3D CAD data, in: Proceedings of the British Machine Vision Conference, Aberystwyth, UK, 2010.

[92] J. Schels, J. Liebelt, K. Schertler, R. Lienhart, Synthetically trained multi-view object class and viewpoint detection for advanced image retrieval, 2011.

[93] B. Pepik, M. Stark, P. Gehler, B. Schiele, Teaching 3D geometry to deformable part models, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012.

[94] S. Satkin, J. Lin, M. Hebert, Data-driven scene understanding from 3D models, in: Proceedings of the British Machine Vision Conference, Surrey, UK, 2012.

[95] M. Aubry, D. Maturana, A. Efros, B. Russell, J. Sivic, Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of cad models, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014.

[96] L.-C. Chen, S. Fidler, A. Yuille, R. Urtasun, Beat the MTurkers: Automatic image labeling from weak 3D supervision, in: Proceedings of the conference on Computer Vision and Pattern Recognition, 2014.

[97] B. Sun, K. Saenko, From virtual to reality: fast adaptation of virtual object detectors to real domains, in: Proceedings of the British Machine Vision Conference, Nottingham, UK, 2014.

[98] J. Papon, M. Schoeler, Semantic pose using deep networks trained on synthetic RGB-D, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[99] X. Peng, B. Sun, K. Ali, K. Saenko, Learning deep object detectors from 3D models, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[100] X. Peng, K. Saenko, Synthetic to real adaptation with deep generative correlation alignment networks (2017).
URL `arXiv:1701.05524`

[101] F. Massa, B. Russell, M. Aubry, Deep exemplar 2D-3D detection by adapting from real to rendered views, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016.

[102] H. Su, F. Wang, E. Yi, L. Guibas, 3D-assisted feature synthesis for novel views of an object, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[103] H. Su, C. Qi, Y. Li, L. Guibas, Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[104] P. Panareda, J. Liebelt, J. Gall, Adaptation of synthetic data for coarse-to-fine viewpoint refinement, in: Proceedings of the British Machine Vision Conference, Swansea, UK, 2015.

[105] Y. Movshovitz-Attias, T. Kanade, Y. Sheikh, How useful is photo-realistic rendering for visual learning?, arXiv:1603.08152 (2016).

[106] A. Broggi, A. Fascioli, P. Grisleri, T. Graf, M. Meinecke, Model-based validation approaches and matching techniques for automotive vision based pedestrian detection, in: Proceedings of the conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005.

[107] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormählen, B. Schiele, Learning people detection models from few training samples, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011.

[108] X. M. N. M. M. B. I. L. C. S. G. Varol, J. Romero, Learning from synthetic humans (2017).
URL arXiv:1701.01370

[109] C. de Souza, A. Gaidon, Y. Cabon, A. Lopez, Procedural generation of videos to train deep action recognition networks, in: Proceedings of the conference on Computer Vision and Pattern Recognition, 2017.

[110] W. Li, M. Fritz, Recognizing materials from virtual examples, in: Proceedings of the European Conference on Computer Vision, Firenze, Italy, 2012.

[111] V. Haltakov, C. Unger, S. Ilic, Framework for generation of synthetic ground truth data for driver assistance applications, in: Proceedings of the German Conference on Pattern Recognition, Saarbrücken, Germany, 2013.

[112] H. Hattori, V. Boddeti, K. Kitani, T. Kanade, Learning scene-specific pedestrian detectors without real data, in: Proceedings of the conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015.

[113] E. Bochinski, V. Eiselein, T. Sikora, Training a convolutional neural network for multi-class object detection using solely virtual world data, 2016.

[114] A. Rozantsev, V. Lepetit, P. Fua, On rendering synthetic images for training and object detector, Computer Vision and Image Understanding 137 (2014) 24 – 37.

[115] A. Shafaei, J. Little, M. Schmidt, Play and learn: Using video games to train computer vision models, in: Proceedings of the British Machine Vision Conference, 2016.

[116] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, R. Cipolla, Understanding real world indoor scenes with synthetic data, in: Proceedings of the conference on Computer Vision and Pattern Recognition, 2016.

[117] B. Kaneva, A. Torralba, W. Freeman, Evaluation of image features using a photorealistic virtual world, in: Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 2011.

[118] M. Aubry, B. Russell, Understanding deep features with computer-generated imagery, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[119] V. Veeravasarapu, R. Hota, C. Rothkopf, R. Visvanathan, Simulations for validation of vision systems, arXiv:1512.01030 (2015).

[120] V. Veeravasarapu, R. Hota, C. Rothkopf, R. Visvanathan, Model validation for vision systems via graphics simulation, arXiv:1512.01401 (2015).

[121] V. Veeravasarapu, C. Rothkopf, R. Visvanathan, Model-driven simulations for deep convolutional neural networks, arXiv:1605.09582 (2016).

[122] D. Butler, J. Wulff, G. Stanley, M. Black, A naturalistic open source movie for optical flow evaluation, in: Proceedings of the European Conference on Computer Vision, Firenze, Italy, 2012.

[123] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: Proceedings of the conference on Computer Vision and Pattern Recognition, 2016.

[124] N. Onkarappa, A. Sappa, Synthetic sequences and ground-truth flow field generation for algorithm validation, Multimedia Tools and Applications 74 (2015) 3121 – 3135.

[125] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixellevel domain adaptation with generative adversarial networks (2017).
URL arXiv:1612.05424

[126] K. Lai, D. Fox, 3D laser scan classification using web data and domain adaptation, in: Proceedings of Robotics: Science and Systems, Seattle, WA, USA, 2009.

[127] K. Lai, D. Fox, Object recognition in 3D point clouds using web data and domain adaptation, International Journal of Robotics Research 29 (8) (2010) 1019–1037.

[128] F. Qureshi, D. Terzopoulos, Towards intelligent camera networks: A virtual vision approach, in: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005.

[129] F. Qureshi, D. Terzopoulos, Surveillance in virtual reality: System design and multi-camera control, in: Proceedings of the conference on Computer Vision and Pattern Recognition, 2007.

[130] S. Musse, M. Paravisi, R. Rodrigues, J. Jacques, C. Jung, Using synthetic ground truth data to evaluate computer vision techniques, in: PETS, 2007.

[131] T. Vaudrey, C. Rabe, R. Klette, J. Milburn, Differences between stereo and motion behaviour on synthetic and real-world stereo sequences, in: Int. Conf. on Image and Vision Computing New Zealand, 2008.

[132] S. Meister, D. Kondermann, Real versus realistically rendered scenes for optical flow evaluation, 2011.

[133] R. Haeusler, D. Kondermann, Synthesizing real world stereo challenges, in: Proceedings of the German Conference on Pattern Recognition, 2013.

[134] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with deep reinforcement learning, in: Proceedings of the Conference on Advances in Neural Information Processing Systems, Workshop on Deep Learning, Lake Tahoe, NV, USA, 2013.

[135] J. Llargues, J. Peralta, R. Arrabales, M. González, P. Cortez, A. López, Artificial intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters, Expert Systems With Applications 41 (2014) 7281 – 7290.

[136] R. Vedantam, X. Lin, T. Batra, C. Zitnick, D. Parikh, Learning common sense through visual abstraction, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[137] C. Zitnick, R. Vedantam, D. Parikh, Adopting abstract images for semantic scene understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2016) 627 – 638.

[138] C. Chen, A. Seff, A. Kornhauser, J. Xiao, DeepDriving: Learning affordance for direct perception in autonomous driving, in: Proceedings of the International Conference on Computer Vision, Santiago de Chile, 2015.

[139] L. Bruzzone, M. Marconcini, Domain adaptation problems: a DASVM classification technique and a circular validation strategy, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1239 – 1258.

[140] A. Dosovitskiy, V. Koltun, Learning to act by predicting the future, in: Proceedings of the International Conference on Learning Representations, 2017.