

The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes

German Ros^{†‡}, Laura Sellart[†], Joanna Materzynska[§], David Vazquez[†], Antonio M. Lopez^{†‡}

[†]Computer Vision Center
Edifici O,
Campus UAB
Barcelona, Spain

[‡]Computer Science Dept.
Universitat Autònoma de Barcelona
Campus UAB
Barcelona, Spain

[§] Faculty of Mathematics,
University of Vienna
Oskar-Morgenstern-Platz
Vienna, Austria

{gros, laura.sellart, dvazquez, antonio}@cvc.uab.es, a1248555@unet.univie.ac.at

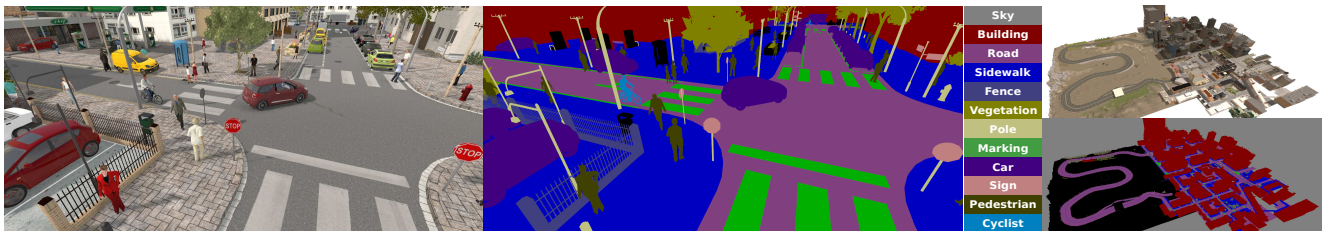


Figure. 1. The SYNTHIA Dataset. A sample frame (Left) with its semantic labels (center) and a general view of the city (right).

Abstract

Vision-based semantic segmentation in urban scenarios is a key functionality for autonomous driving. Recent revolutionary results of deep convolutional neural networks (DCNNs) foreshadow the advent of reliable classifiers to perform such visual tasks. However, DCNNs require learning of many parameters from raw images; thus, having a sufficient amount of diverse images with class annotations is needed. These annotations are obtained via cumbersome, human labour which is particularly challenging for semantic segmentation since pixel-level annotations are required. In this paper, we propose to use a virtual world to automatically generate realistic synthetic images with pixel-level annotations. Then, we address the question of how useful such data can be for semantic segmentation – in particular, when using a DCNN paradigm. In order to answer this question we have generated a synthetic collection of diverse urban images, named SYNTHIA, with automatically generated class annotations. We use SYNTHIA in combination with publicly available real-world urban images with manually provided annotations. Then, we conduct experiments with DCNNs that show how the inclusion of SYNTHIA in the training stage significantly improves performance on the semantic segmentation task.

1. Introduction

Autonomous driving (AD) will be one of the most revolutionary technologies in the near future in terms of the impact on the lives of citizens of the industrialized countries [43]. Nowadays, advanced driver assistance systems (ADAS) are already improving traffic safety. The computer vision community, among others, is contributing to the development of ADAS and AD due to the rapidly increasing performance of vision-based tools such as object detection, recognition of traffic signs, road segmentation, etc. At the core of such functionality are various types of classifiers.

Roughly until the end of the first decade of this century, the design of classifiers for recognizing visual phenomena was viewed as a two-fold problem. First, enormous effort was invested in research of discriminative visual descriptors to be fed as features to classifiers; as a result, descriptors such as Haar wavelets, SIFT, LBP, or HOG, were born and their use became widespread. Second, many different machine learning methods were developed, with dis-

Acknowledgements: Authors want to thank Andrew Bagdanov for his help and proofreading and the next funding bodies: the Spanish MEC Project TRA2014-57088-C2-1-R, the Spanish DGT Project SPIP2014-01352, the People Programme (Marie Curie Actions) FP7/2007-2013 REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIO, the Generalitat de Catalunya Project 2014-SGR-1506 and the NVIDIA Corporation for the generous support in the form of different GPU hardware units.

criminative algorithms such as SVM, AdaBoost, or Random Forests usually reporting the best classification accuracy due to their inherent focus on searching for reliable class boundaries in feature space. Complementarily, in order to make easier the search for accurate class boundaries, methods for transforming feature space were also developed (e.g. PCA, BoW encoding, Kernel mappings) as well as more elaborate class models (e.g. DPM, superpixels).

In practice, even the best visual descriptors, class models, feature encoding methods and discriminative machine learning techniques are not sufficient to produce reliable classifiers if properly annotated datasets with sufficient diversity are not available. Indeed, this is not a minor issue since data annotation remains a cumbersome, human-based labor prone to error; even exploiting crowdsourcing for annotation is a non-trivial task [39]. For instance, for some ADAS and for AD, semantic segmentation is a key issue [33, 18, 29] and it requires pixel-level annotations (*i.e.* obtained by delineating the silhouette of the different classes in urban scenarios, namely pedestrian, vehicle, road, sidewalk, vegetation, building, etc).

In order to ameliorate this problem there are paradigms such as unsupervised learning (no annotations assumed), semi-supervised learning (only a few annotated data), and active learning (to focus on annotating informative data), under the assumption that having annotated data (*e.g.* images) is problematic but data collection is cheap. However, for ADAS and AD such data collection is also an expensive activity since many kilometers must be traveled to obtain sufficient diversity. Moreover, it is well-known that, in general terms, supervised learning (annotations assumed) tends to provide the most accurate classifiers.

Recently, the need for large amounts of accurately annotated data has become even more crucial with the massive adoption of deep convolutional neural networks (DCNNs) by the computer vision community. DCNNs have yielded a significant performance boost for many visual tasks [17, 10, 40, 35]. Overall, DCNNs are based on highly non-linear, end-to-end training (*i.e.* from the raw annotated data to the class labels) which implies the learning of millions of parameters and, accordingly, they require a relatively larger amount of annotated data than methods based on hand-crafted visual descriptors.

As we will review in section 2, the use of visually realistic synthetic images is gaining attention in recent years (*e.g.*, training in virtual worlds [21, 34, 1, 26, 12], synthesizing images with real-world backgrounds and inserted virtual objects [28, 27]) due to the possibility of having diversified samples with automatically generated annotations. In this spirit, in this paper we address the question of *how useful can the use of realistic synthetic images of virtual-world urban scenarios be for the task of semantic segmentation – in particular, when using a DCNN paradigm*. To the best of

our knowledge, this analysis has not been done so far. Note that, in this setting the synthetic training data can not only come with automatically generated class annotations from multiple points of views and simulated lighting conditions (providing diversity), but also with ground truth for depth (simulating stereo rigs and LIDAR is possible), optical flow, object tracks, etc.

Moreover, in the context of ADAS/AD the interest in using virtual scenarios is already increasing for the task of validating functionalities in the Lab, *i.e.* to perform validation in the real world (which is very expensive) only once after extensive and well-designed simulations are passed. Therefore, these virtual worlds can be used for generating synthetic images to training the classifiers involved in environmental perception. In addition, the realism of these virtual worlds is constantly increasing thanks to the continuously growing videogames industry.

To address the above mentioned question, we have generated *SYNTHIA*: a *SYNTHetic* collection of *Imagery* and *Annotations* of urban scenarios.¹ *SYNTHIA* is detailed in section 3 where we highlight its diversity and how we can automatically obtain a large number of images with annotations. On the other hand, it is known that classifiers trained only with virtual images may require domain adaptation to work on real images [42, 44, 37, 25, 45]; however, it has been shown that this is just because virtual and real world cameras are different *sensors*, *i.e.* domain adaptation is also often required when training images and testing images come from different real-world camera sensors [42, 41].

As we will see in section 4, where we explain the DCNN used to perform semantic segmentation, in this work we use a simple domain adaptation strategy which consists of training with the synthetic data and a smaller number of real-world data simultaneously, *i.e.* in the same spirit than [42] for a HOG-LBP/SVM setting. In our case, the data combination is done in the generation of batches during DCNN training. The experiments conducted in section 5 show how *SYNTHIA* successfully complements different datasets (Camvid, KITTI, U-LabelMe, CBCL) for the task of semantic segmentation based on DCNNs, *i.e.* the use of the combined data significantly boosts the performance obtained when using the real-world data alone. The future work that we foresee given these results is pointed out in section 6, together with the conclusions of the paper.

2. Related Work

The generation of semantic segmentation datasets with pixel-level annotations is costly in terms of effort and money, factors that are currently slowing down the development of new large-scale collections like ImageNet [15]. Despite these factors, the community has invested great effort

¹*SYNTHIA* is available at adas.cvc.uab.es/synthia



Figure 2. Dynamic objects catalogue of SYNTHIA. (Top) vehicles examples; (middle) cyclists; (bottom) pedestrians.

to create datasets such as the NYU-Depth V2 [23] (more than 1,449 images densely labelled), the PASCAL-Context Dataset [22] (10,103 images densely labelled over 540 categories), and MS COCO [19] (more than 300,000 images with annotations for 80 object categories). These datasets have definitely contributed to boost research on semantic segmentation of indoor scenes and also on common objects, but they are not suitable for more specific tasks such as those involved in autonomous navigation scenarios.

When semantic segmentation is seen in the context of autonomous vehicles, we find that the amount and variety of annotated images of urban scenarios is much lower in terms of total number of labeled pixels, number of classes and instances. A good example is the CamVid [4] dataset, which consists of a set of monocular images taken in Cambridge, UK. However, only 701 images contain pixel-level annotations over a total of 32 categories (combining objects and architectural scenes), although usually only the 11 largest categories are used. Similarly, Daimler Urban Segmentation dataset [33] contains 500 fully labelled monochrome frames for 5 categories. The more recent KITTI benchmark suite [9] has provided a large amount of images of urban scenes from Karlsruhe, Germany, with ground truth data for several tasks. However, it only contains a total of 430 labelled images for semantic segmentation.

A common limitation of the aforementioned datasets is the bias introduced by the acquisition of images in a specific city. The LabelMe project [32], later refined by [30], corrects this by offering around 1,000 fully annotated images of urban environments around the world and more than 3,000 images with partial (noisy) annotations.

A larger dataset is the CBCL StreetScenes [3], which contains 3,547 images of the streets of Chicago over 9 classes with noisy annotations. This dataset has recently been enhanced in [30], improving the quality of the annotations and adding extra classes. To date, the largest dataset for semantic segmentation is the CityScapes dataset [8], which consists of a collection of images acquired in 50

cities around Germany, Switzerland and France in different seasons, and having 5,000 images with fine annotations and 20,000 with coarse annotations over a total of 30 classes. However, the cost of scaling this sort of project would require a prohibitive economic investment in order to capture images from a larger variety of countries, in different seasons and different traffic conditions. For these reasons, a promising alternative proposed in this work is to use synthetic imagery that simulate real urban scenes in a vast variety of conditions and produce the appropriate annotations.

The use of synthetic data has increased considerably in recent years within the computer vision community for several problems. For instance, in [14], the authors used a virtual world to evaluate the performance of image features under certain types of changes. In the area of object detection, similar approaches have been proposed by different groups [27, 37, 21, 12], making use of CAD models, virtual worlds and studying topics such as the impact of a realistic world on the final accuracy of detectors and the importance of domain adaptation. Synthetic data has also been used for pose estimation [1, 6] to compensate for the lack of precise pose annotations of objects. The problem of semantic segmentation has also begun to benefit from this trend, with the creation of virtual scenes to perform segmentation of indoor environments [11, 26].

The present work proposes a novel synthetic dataset of urban scenes, which we call SYNTHIA. This dataset is a large collection of images with high variability due to changes in illumination, textures, pose of dynamic objects and camera view-points. We also explore the benefits of using SYNTHIA in the context of semantic segmentation of urban environments with DCNNs.

3. The SYNTHIA Dataset

Here we describe our synthetic dataset of urban scenes, which we call the SYNTHetic collection of Imagery and Annotations (SYNTHIA). This dataset has been generated with the purpose of aiding semantic segmentation in the

context of AD problems, but it contains enough information to be useful in additional ADAS and AD-related tasks, such as object recognition, place identification and change detection, among others.

SYNTHIA consists of photo-realistic frames rendered from a virtual city and comes with precise pixel-level semantic annotations for 13 classes, *i.e.*, sky, building, road, sidewalk, fence, vegetation, lane-marking, pole, car, traffic signs, pedestrians, cyclists and miscellaneous (see Fig. 1). Frames are acquired from multiple view-points (up to eight views per location), and each of the frames also contains an associated depth map (though they are not used in this work).

3.1. Virtual World Generator

SYNTHIA has been generated by rendering a virtual city created with the Unity development platform [38]. This city includes the most important elements present on driving environments, such as: street blocks, highways, rural areas, shops, parks and gardens, general vegetation, variety of pavements, lane markings, traffic signs, lamp poles, and people, among others. The virtual environment allows us to freely place any of these elements in the scene and to generate its semantic annotations without additional effort. This enables the creation of new and diverse cities as a simple combination of basic blocks. The basic properties of these blocks, such as textures, colors and shapes can be easily changed to produce new looks and to enhance the visual variety of the data.

The city is populated with realistic models of cars, vans, pedestrians and cyclists (see Fig. 2). In order to extend visual variability, some of these models are modified to generate new and distinctive versions.

We have defined suitable material coefficients for each of the surfaces of the city in order to produce photo realistic outcomes that look as similar as possible to real data. Our virtual world also includes four different seasons with drastic change of appearance, with snow during winter, blooming flowers during spring, etc., (see Fig. 3). Moreover, a dynamic illumination engine serves to produce different illumination conditions, to simulate different moments of the day, including sunny and cloudy days and dusk. Shadows caused by clouds and other objects are dynamically cast on the scene, adding additional realism.

We would like to highlight the potential of this virtual world in terms of extension capabilities. New parts of the cities can be easily generated by adding existing blocks in different setups and additional ground truth can be produced almost effortlessly. Extending the number of classes of the city is also a simple task which consists of assigning a new id to objects. In this way we can generate a broad variety of urban scenarios and situations, which we believe is very useful to help modern classifiers based on deep learning.

3.2. SYNTHIA-Rand and SYNTHIA-Seqs

From our virtual city we have generated two complementary sets of images, referred to as *SYNTHIA-Rand* and *SYNTHIA-Seqs*. Both sequences share standard properties as frame resolution of 960×720 pixels and horizontal field of view of 100 degrees.

SYNTHIA-Rand consists of 13,400 frames of the city taken from a virtual array of cameras moving randomly through the city, with its height limited to the range $[1.5m, 2m]$ from the ground. In each of the camera poses, several frames are acquired changing the type of dynamic objects present in that part of the scene along with the illumination of the scene and the textures of road and sidewalks. We enforce that the separation between camera positions is at least of 10 meters in order to improve visual variability. This collection is oriented to serve as training data for semantic segmentation methods based on DCNNs.

SYNTHIA-Seqs simulates four video sequences of approximately 50,000 frames each one up to a total of 200,000 frames, acquired from a virtual car across different seasons (one sequence per season). The virtual acquisition platform consists of two multi-cameras separated by a baseline $B = 0.8m$ in the x-axis. Each of these multi-cameras consists of four monocular cameras with a common center and orientations varying every 90 degrees, as depicted in Fig. 4. Since all cameras have a field of view of 100 degrees the visual overlapping serves to create an omnidirectional view on demand, as shown in Fig. 5. Each of these cameras also has a virtual depth sensor associated, which works in a range from 1.5 to 50 meters and is perfectly aligned with the camera center, resolution and field of view (Fig. 5, bottom). The virtual vehicle moves through the city interacting with dynamic objects such as pedestrians and cyclists that present dynamic behaviour. This interaction produces changes in the trajectory and speed of the vehicle and leads to variations of each of the individual video sequences. This collection is oriented to provide data to exploit spatio-temporal constraints of the objects.

4. Semantic Segmentation & Synthetic Images

We first define a simple but competitive deep Convolutional Neural Network (CNN) for the task of semantic segmentation of urban scenes, following the description of [30] (section 4.1). This architecture, referred as Target-Net (T-Net) is more suitable for its application to urban scenes due to its reduced number of parameters. As a reference, we also consider the FCN [20], a state-of-the-art architecture for general semantic segmentation. In 4.2 we describe the strategy used to deal with the synthetic domain (virtual data) and the real domain during the training stage.



Figure 3. The same area captured in different seasons and light conditions. Top-left, fall; top-right, winter; bottom-left, spring; bottom-right, summer.

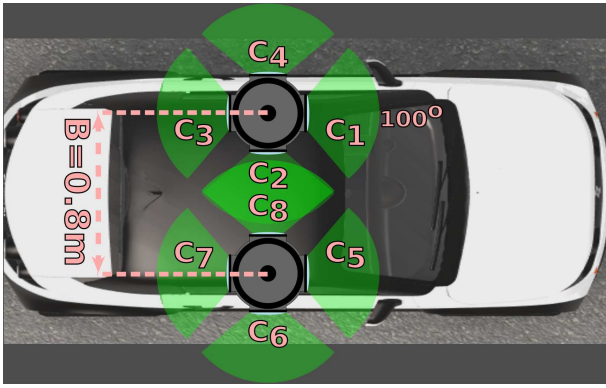


Figure 4. Virtual car setup used for acquisition. Two multi-cameras with four monocular cameras are used. The baseline between the cameras is $0.8m$ and the FOV of the cameras is 100° .

4.1. Architectures Specification

T-Net [30] architecture along with its associated training procedure is drawn from [30], due to its good performance and ease of training. Similar architectures have proven to be very effective in terms of accuracy and efficiency for segmentation of general objects [24] and urban scenes [2]. Fig. 6 shows a graphical schema of T-Net. The architecture is based on a combination of contraction, expansion blocks and a soft-max classifier. Contraction blocks consist of convolutions, batch normalization, ReLU and max-pooling with indices storage. Expansion blocks consist of an unpooling of the blob using the pre-stored indices, convolution, batch normalization and ReLU.

FCN [20] architecture is an extension of VGG-16 [36] with deconvolution modules. Different from T-Net, FCN does not use batch normalization and its upsampling scheme is based on deconvolutions and mixing information across layers.

We use weighted cross-entropy as a loss function for both architectures, where the weights are computed as the inverse frequencies of each of the classes for the training data [2]. This helps prevent problems due to class imbal-

ance. During training the contraction blocks are initialized using VGG-F [7] for T-Net and VGG-16 [36] for FCN, pre-trained on ILSVRC [31]. Kernels are accordingly re-scaled when the original sizes do not match. Expansion blocks are randomly initialized following the method of He et al. [13]. Input data is pre-processed using local contrast normalization of each channel independently to avoid problems with drastic illumination changes.

Networks are trained end-to-end using Adam [16] since the learning rates are automatically adjusted. Using Adam leads the network to converge in a couple of hundred iterations, speeding up the training procedure considerably.

4.2. Training on Real and Synthetic Data

The aim of this work is to show that the use of synthetic data helps to improve semantic segmentation results on real imagery. There exist several ways to exploit synthetic data for this purpose. A trivial option would be to use the synthetic data alone for training a model and then apply it on real images. However, due to domain shift [37, 42] this approach does not usually perform well. An alternative is to train a model on the vast amount of synthetic images and afterwards fine-tuning it on a reduced set of real images. This leads to better results, since the statistics of the real domain are considered during the second stage of training [27].

However, here we employ the Balanced Gradient Contribution (BGC) that was first introduced in [30]. It consists of building batches with images from both domains (synthetic and real), given a fixed ratio. Real images dominate the distribution, while synthetic images are used as a sophisticated regularization term. Thus, statistics of both domains are considered during the whole procedure, creating a model which is accurate for both. In section 5 we show that extending real data with synthetic images using this technique leads to a systematic boost in segmentation accuracy.

5. Experimental Evaluation

We present the evaluation of the DCNNs for semantic segmentation described in section 4, training and evaluating on several state-of-the-art datasets of driving scenes. We test how the new SYNTHIA dataset can be useful both on its own and along with real images to produce accurate segmentation results. For the following experiments we have made use of the 13,400 images of the *SYNTHIA-Rand* collection to favour visual variability while using a moderate number of images.

5.1. Validation Datasets

We selected publicly available urban datasets to study the benefits of SYNTHIA. Table 1 shows the different datasets used in our experiments, along with a definition of the number of images used in our experiments for training (T) and validation (V).

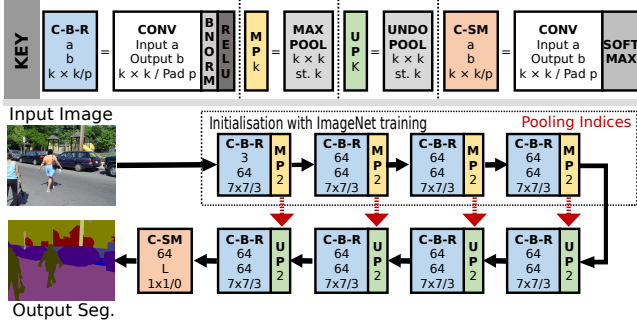


Figure 6. Graphical schema of the semantic segmentation network consisting of a set of contraction (yellow) -expansion (green) blocks. In the legend, for convolutions a and b stand for the input and output number of channels respectively; k is the kernel size and p the padding size. For pooling $st.$ stands for stride.

Table 1. Driving scenes sets for semantic segmentation. We define the number of training images (T), validation (V) and in total (A).

Dataset	# Frames (A)	# Training (T)	# Validation (V)
CamVid [4, 5]	701	300	401
KITTI [9, 18, 29, 30]	547	200	347
Urban LabelMe [32, 30]	942	200	742
CBCL StreetScenes [3, 30]	3547	200	3347
SYNTHIA-Rand	13,400	13,400	0

It is worth highlighting the differences between these datasets. Each of them has been acquired in a different city or cities. CamVid and KITTI datasets have high quality labels and low complexity in terms of variations and atypical scenes. Urban LabelMe (U-LabelMe) is very challenging, since it contains images from different cities and several view-points. It has been annotated by several users and contains some images with partial and noisy annotations. CBCL images are also challenging, and contain many noisy, semi-supervised annotations [30]. Each of the individual splits is designed to include a large number of validation images, keeping enough images for training on each datasets.

5.2. Analysis of Results

The following experiments have been carried out by training DCNNs using low-resolution images. All images are resized to a common resolution of 180×120 . This is done to speed-up the training process and save memory.

However, it has the disadvantage of decreasing the recognition of certain textures present in roads and sidewalks and makes it harder to recognize small categories such as traffic signs and poles. This fact needs to be considered to correctly understand the results of our experiments.

In our first experiment we evaluate the capability of *SYNTHIA-Rand* in terms of the generalization of the trained models on state-of-the-art datasets. To this end we report in Table 2 the accuracy (%) of T-Net and FCN for each of the 11 classes along with their average per-class and global accuracies for each of the validation sets (V).

The networks trained on just synthetic data produce good results recognizing roads, buildings, cars and pedestrians in the presented datasets. Moreover, sidewalks are fairly well recognized in CamVid, probably due to their homogeneity. The high accuracy at segmenting roads, cars and pedestrians in U-LabelMe—one of the most challenging datasets due to the large variety of view-points—is a proof of the high quality of SYNTHIA. Notice also that FCN performs better than T-Net for many of the classes due to the higher capacity of the model, although in practice FCN has the disadvantage of being too large for embedded context such as autonomous driving. It is worth highlighting that the average per-class accuracy of the models trained with SYNTHIA is close or some times even higher (e.g. T-Net: CamVid, U-LabelMe, CBCL; FCN: CamVid, CBCL) than those models trained on real data (see Table 3).

Our second experiment evaluates the true potential of *SYNTHIA* to boost DCNN models trained on real data. To this end we perform several tests combining data from *SYNTHIA-Rand* along with individual real datasets, following the strategy defined in section 4.2. Here, each batch contains 6 images from the real domain and 4 from the synthetic domain. These results are compared against using just real data coming from each respective training split (T). The outcome of this experiment is shown in Table 3. Observe that, for all the datasets and architectures, the inclusion of synthetic data systematically helps to boost the average per-class accuracy. Improvements with respect to the baselines (training only with real data) are highlighted in blue. Notice that for both, T-Net and FCN there are improvements

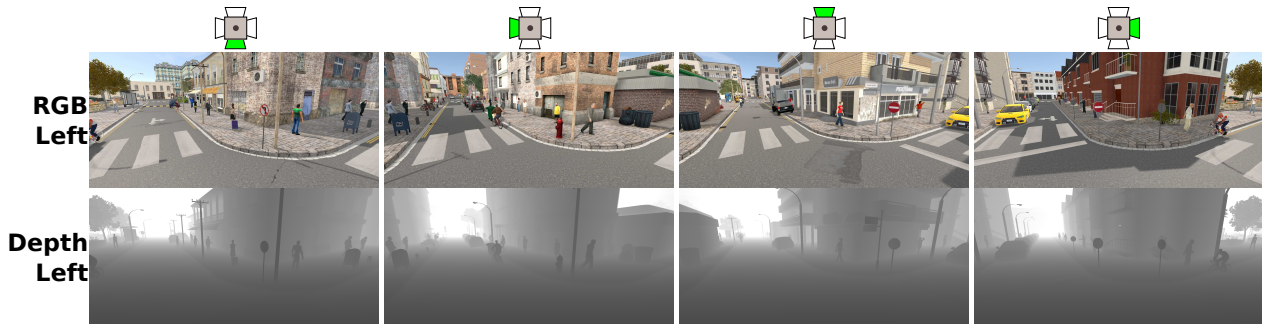


Figure 5. One shot example: the four views from the left multi-camera with its associated depth maps.

Table 2. Results of training a T-Net and a FCN on *SYNTHIA-Rand* and evaluating it on state-of-the-art datasets of driving scenes.























Method	Training	Validation												per-class	global
T-Net [30]	<i>SYNTHIA-Rand</i> (A)	CamVid (V)	66	85	86	67	0	27	55	79	3	75	46	48.9	79.7
	<i>SYNTHIA-Rand</i> (A)	KITTI (V)	73	78	92	27	0	10	0	64	0	72	14	39.0	61.9
	<i>SYNTHIA-Rand</i> (A)	U-LabelMe (V)	20	59	92	13	0	22	38	89	1	64	23	38.3	53.4
	<i>SYNTHIA-Rand</i> (A)	CBCL (V)	74	71	87	25	0	35	21	68	2	42	36	41.8	66.0
FCN [20]	<i>SYNTHIA-Rand</i> (A)	CamVid (V)	78	66	86	72	12	79	17	91	43	78	68	62.5	74.9
	<i>SYNTHIA-Rand</i> (A)	KITTI (V)	56	65	59	26	17	65	32	52	42	73	40	47.1	62.7
	<i>SYNTHIA-Rand</i> (A)	U-LabelMe (V)	31	63	68	40	23	65	39	85	18	71	46	50.0	59.1
	<i>SYNTHIA-Rand</i> (A)	CBCL (V)	71	59	73	32	26	81	40	78	31	63	72	56.9	68.2

Table 3. Comparison of training a T-Net and FCN on real images only and the effect of extending training sets with *SYNTHIA-Rand*.

Method	Training	Validation												per-class	global
T-Net [30]	Camvid (T)	CamVid (V)	99	65	95	52	7	79	5	80	3	26	6	46.3	81.9
	Camvid (T) + <i>SYNTHIA-Rand</i> (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) + <i>SYNTHIA-Rand</i> (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) + <i>SYNTHIA-Rand</i> (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) + <i>SYNTHIA-Rand</i> (A)	CBCL (V)	72	82	90	39	0	58	26	70	5	52	39	48.4 (10.5)	75.2 (1.3)
FCN [20]	Camvid (T)	CamVid (V)	99	65	98	45	27	54	16	77	11	34	25	52.8	78.4
	Camvid (T) + <i>SYNTHIA-Rand</i> (A)	CamVid (V)	97	70	98	66	39	88	41	88	53	75	79	72.1 (18.3)	83.6 (5.2)
	KITTI (T)	KITTI (V)	75	77	77	64	47	84	18	78	5	1	1	51.5	82.3
	KITTI (T) + <i>SYNTHIA-Rand</i> (A)	KITTI (V)	84	81	82	71	60	86	43	83	24	7	32	59.4 (7.9)	80.8 (-1.5)
	U-LabelMe (T)	U-LabelMe (V)	93	81	83	57	2	79	41	72	20	71	63	60.1	79.4
	U-LabelMe (T) + <i>SYNTHIA-Rand</i> (A)	U-LabelMe (V)	93	72	81	63	10	76	46	79	49	76	64	64.4 (4.3)	76.2 (-3.2)
	CBCL (T)	CBCL (V)	90	77	90	41	2	80	37	84	10	47	31	53.4	79.7
	CBCL (T) + <i>SYNTHIA-Rand</i> (A)	CBCL (V)	82	78	74	56	1	80	20	78	8	77	35	53.5 (0.2)	75.2 (-4.5)

of more than 10 points (up to 18.3 points) in per-class accuracy. We believe that the decrement of global accuracy for FCN may be related to the combination of early and late layers during the upsampling process and the use of BGC, but further investigation is still required. The classes that most benefit from the addition of synthetic data are pedestrian, car and cyclist (dynamic objects), which is due to the lack of enough instances of these classes in the original datasets. On the other hand, signs and poles are very hard to segment as a consequence of the low-resolution images.

Fig. 7 shows qualitative results of the previous experiments. Observe how the training on synthetic data is good enough to recognize pedestrians, roads, cars and some cyclists. Then the combination of real and synthetic data (right column) produces smooth and very accurate results for both objects and architectural elements, even predicting thin objects like poles. We consider the results of these experi-

ments an important milestone for the use of synthetic data as the main information source for semantic segmentation.

6. Conclusions

We presented SYNTHIA, a new dataset for semantic segmentation of driving scenes with more than 213,400 synthetic images including both, random snapshots and video sequences in a virtual city. Images are generated simulating different seasons, weather and illumination conditions from multiple view-points. Frames include pixel-level semantic annotations and depth. SYNTHIA was used to train DCNNs for the semantic segmentation of 11 common classes in driving scenes. Our experiments showed that SYNTHIA is good enough to produce good segmentations by itself on real datasets, dramatically boosting accuracy in combination with real data. We believe that SYNTHIA will help to boost semantic segmentation research.

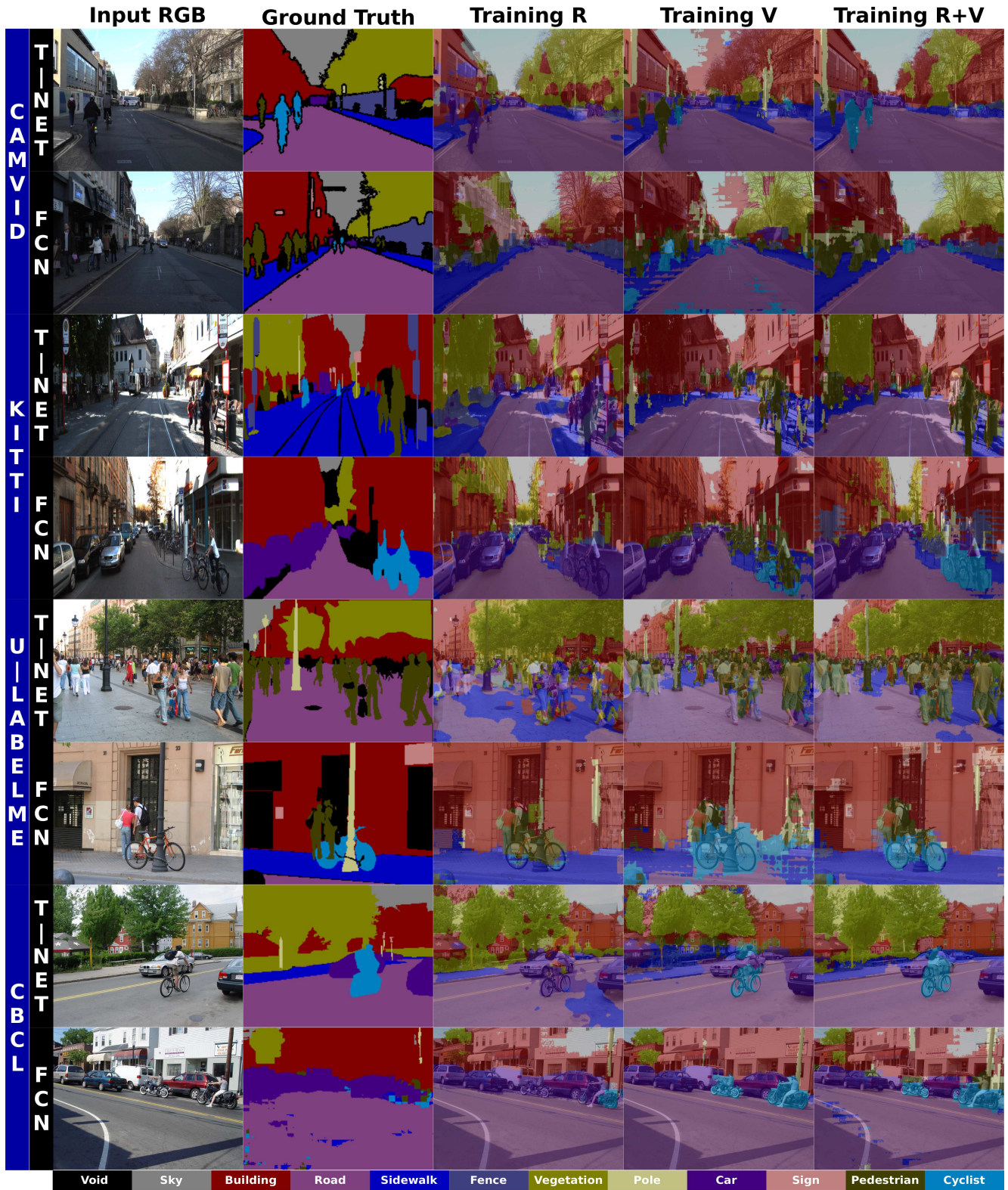


Figure 7. Qualitative results for different testing datasets and architectures (T-Net and FCN). First column shows the **RGB** testing frame; second column is the **ground truth**; **Training R** is the result of training with the real dataset; **Training V** is the result of training with SYNTHIA; **Training R+V** is the result of training with the real and SYNTHIA-Rand collection. Including SYNTHIA for training considerably improves the results.

References

- [1] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [2] V. Badrinarayanan, A. Handa, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint abs/1505.07293*, 2015.
- [3] S. Bileschi. CBCL StreetScenes challenge framework, 2007.
- [4] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009.
- [5] G. J. Brostow, J. Shotton, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Eur. Conf. on Computer Vision (ECCV)*, 2008.
- [6] P. P. Busto, J. Liebelt, and J. Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In *British Machine Vision Conf. (BMVC)*, 2015.
- [7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional networks. In *British Machine Vision Conf. (BMVC)*, 2014.
- [8] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *CVPR, Workshop*, 2015.
- [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *Intl. J. of Robotics Research*, 2013.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [11] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Synthcam3d: Semantic understanding with synthetic indoor scenes. *arXiv preprint abs/1505.00171*, 2015.
- [12] H. Hattori, V. Naresh Boddeti, K. M. Kitani, and T. Kanade. Learning scene-specific pedestrian detectors without real data. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. 2015.
- [14] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluating image features using a photorealistic virtual world. In *Intl. Conf. on Computer Vision (ICCV)*, 2011.
- [15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
- [18] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *Eur. Conf. on Computer Vision (ECCV)*, 2014.
- [19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Eur. Conf. on Computer Vision (ECCV)*, 2014.
- [20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [21] J. Marin, D. Vazquez, D. Geronimo, and A. Lopez. Learning appearance in virtual scenarios for pedestrian detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [22] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [23] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Eur. Conf. on Computer Vision (ECCV)*, 2012.
- [24] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *Intl. Conf. on Computer Vision (ICCV)*, 2015.
- [25] P. Panareda, J. Liebelt, and J. Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In *British Machine Vision Conf. (BMVC)*, 2015.
- [26] J. Papon and M. Schoeler. Semantic pose using deep networks trained on synthetic RGB-D. In *Intl. Conf. on Computer Vision (ICCV)*, 2015.
- [27] X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3D models. In *Intl. Conf. on Computer Vision (ICCV)*, 2015.
- [28] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele. Articulated people detection and pose estimation: reshaping the future. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [29] G. Ros, S. Ramos, M. Granados, A. Bakhtiyar, D. Vázquez, and A. M. López. Vision-based offline-online perception paradigm for autonomous driving. In *Winter Conference on Applications of Computer Vision (WACV)*, 2015.
- [30] G. Ros, S. Stent, P. F. Alcantarilla, and T. Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *arXiv preprint abs/1604.01545*, 2016.
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *Intl. J. of Computer Vision*, 2015.
- [32] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *Intl. J. of Computer Vision*, 2008.
- [33] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. Efficient multi-cue scene segmentation. In *Pattern Recognition*, 2013.
- [34] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose

- recognition in parts from a single depth image. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *arXiv preprint abs/1406.2199*, 2014.
 - [36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
 - [37] B. Sun and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *British Machine Vision Conf. (BMVC)*, 2014.
 - [38] U. Technologies. Unity Development Platform.
 - [39] T.L. Berg, A. Sorokin, G. Wang, D.A. Forsyth, D. Hoeiem, I. Endres, and A. Farhadi. It’s all about the data. *Proceedings of the IEEE*, 2010.
 - [40] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, 2014.
 - [41] A. Torralba and A. Efros. Unbiased look at dataset bias. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.
 - [42] D. Vázquez, A. López, J. Marín, D. Ponsa, and D. Gerónimo. Virtual and real world adaptation for pedestrian detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 2014.
 - [43] L. Woensel and G. Archer. Ten technologies which could change our lives. Technical report, EPRS - European Parliamentary Research Service, January 2015.
 - [44] J. Xu, S. Ramos, D. Vazquez, and A. Lopez. Domain adaptation of deformable part-based models. *IEEE Trans. Pattern Anal. Machine Intell.*, 2014.
 - [45] J. Xu, S. Ramos, D. Vazquez, and A. M. Lopez. Hierarchical adaptive structural SVM for domain adaptation. 2016.