Learning from THEODORE: A Synthetic Omnidirectional Top-View Indoor Dataset for Deep Transfer Learning

Tobias Scheck*, Roman Seidel*, Gangolf Hirtz
Chemnitz University of Technology
Faculty of Electrical Engineering and Information Technology
09126 Chemnitz, Germany
tobias.scheck,roman.seidel,g.hirtz@etit.tu-chemnitz.de

Abstract

Recent work about synthetic indoor datasets from perspective views has shown significant improvements of object detection results with Convolutional Neural Networks (CNNs). In this paper, we introduce THEODORE: a novel, large-scale indoor dataset containing 100,000 high-resolution diversified fisheye images with 16 classes. To this end, we create 3D virtual environments of living rooms, different human characters and interior textures. Beside capturing fisheye images from virtual environments we create annotations for semantic segmentation, instance masks and bounding boxes for object detection. We compare our synthetic dataset to state of the art real-world datasets for omnidirectional images. Based on MS COCO weights, we show that our dataset is well suited for fine-tuning CNNs for object detection and semantic segmentation. Through a high generalization of our models by means of image synthesis and domain randomization we reach a AP up to 0.90 for class person on our own annotated fisheye evaluation suite (FES). Additionally, the evaluation of six classes was done through object detection and semantic segmentation on FES. The segmentation task on FES leads to 0.36 mIoU on all classes and to a mAP of 0.61 for the object detection.

1. Introduction

Synthetic images and labels from modeled 3D indoor scenes has been an increasing research field in computer vision in the last few years. In contrast to manually labeled indoor front-view perspective images for action recognition [28, 37, 39] that are widely explored, image data from top-view indoor scenes of omnidirectional images are rarely available. Invariance against the perspective of objects, e.g. missing images from top-view scenes makes common datasets not adaptable to computer vision tasks on omnidirectional cameras. The most widely used projection of fisheye images is the equirectangular camera model, where all image points are mapped to the inside of a lower half sphere through elevation and azimuth. This projection formulates the distortion of omnidirectional images and leads to a high variation of the shape of objects depending on their location in the image.

In this paper we introduce *THEODORE - a synTHEtic tOp-view inDoOR scEnes* dataset that contains diversified rendered fisheye images of indoor environments with instance segmentation masks and bounding boxes. The indoor world was created with the game engine Unity3D and rendered images were captured with a camera that follows the omnidirectional projection. To bridge the gap between real-world and synthetic images we perform domain randomization with different rooms, persons, objects and camera positions. With THEODORE we release a dataset that improves the accuracy of state-of-the-art CNNs on omnidirectional images in indoor environments. A few application fields are navigation of autonomous systems through visual odometry, personal security in public transportation services or in virtual reality. We expect a strong growth of the research field on omnidirectional images in computer vision.

Our contribution is twofold:

- Generating THEODORE: a dataset with diversified omnidirectional images and labels for indoor scenes
- Improvement of accuracy of state-of-the-art CNNs for object detection in omnidirectional images

The paper is structured as follows. Following this introduction in chapter 2 we treat related works to synthetic image data and one- and two-stage object detection. In chapter 3 we describe the data generation process and its properties. In chapter 4 we describe the behaviour of state of the art single- and two-stage object de-

*Authors contributed equally

tectors in terms of THEODORE. The evaluation on publicly available databases for omnidirectional images is shown in Chapter 5. We summarize our results and give future research directions in Chapter 6. Our dataset can be found at `https://www.tu-chemnitz.de/etit/dst/forschung/comp_vision/theodore`.

2. Related Work

Synthetic data for omnidirectional images isn't well explored and data for training CNNs are sparsely available. In this section the most relevant indoor datasets and CNN architectures for object detection are introduced.

Synthetic Data Synthetic data of persons in perspective views was widely studied [47] for tasks like object detection, segmentation or human pose estimation [21, 22]. For the analysis of multi-object tracking Gaidon *et al.* created the Virtual KITTI dataset [16], including different environment conditions, camera position and instance-level segmentation ground truth. A couple of 3D model repositories for indoor scenes [18, 25] in perspective views with focus on depth, physical based rendering and volumetric ground truth, namely the SUNCG dataset [41] and the Matterport dataset [6], were released. Based on these datasets the work of [42] generates a RGB-D panorama dataset for different camera configurations, but without different camera models and top-view images. While multisensory models for goal-directed navigation in complex indoor environments from ego-perspective MINOS [38] was published, extensive research in terms of semantic descriptions, acoustics and multi agent support from 3D visual renderings led to HoME [5]. With the goal to create houshold activities in virtual homes the work of [33] delivers instance and semantic label annotation, depth, pose and optical flow. The novelty of this approach is the formulation of the automatic generation of program episodes from text and creatable avatar videos. Our approach differs from this work in terms of camera geometry, domain randomization and viewing angle. House3D [48] provides 3D scenes of visually realistic houses that are equipped with a diverse set of fully labeled 3D objects and textures based on the SUNCG dataset including RGB images, depth, segmentation masks and top-down 2D map views. In terms of selecting the viewing angle of indoor scenes automatically, the work of [17] uses per class statistics to find the best viewing angle for semantic segmentation.

Existing datasets of omnidirectional image data ([7, 11, 12, 13, 15]) have low in-class variance, missing ground truth labels or contain less variations of scenes [26].

Object Detection One-stage object detectors [29, 35] that treat object detection as a simple regression task learn class probabilities and bounding box coordinates. Two-stage detectors such as [36] and [10] generate regions of interest (ROIs) by a Region Proposal Network in the first and forward these ROIs to the object classification and bounding box regression pipeline.

Object detection in distorted fisheye images is not widely explored. The authors of [9] and [8] adapt the network architecture of CNNs to spherical representations of the regular convolution operations. However, [9] wraps the sampled locations of convolutional filters to the sphere and effectively reverses the distortions of the omnidirectional camera model. [8] avoids translational weight sharing and creates building blocks that satisfy a generalized Fourier theorem, to detect patterns independently from their location on the sphere.

Current frameworks with AI agents ([23, 48]) concentrate on embodied question answering or navigation (PointGoal, ObjectGoal and RoomGoal). Images and corresponding labels (segmentation masks, surface normals, object IDs, depth) can be created, but are missing for omnidirectional camera model. In contrast to our work, viewing angle of AI agent frameworks is front-view from an ego-perspective.

Taylor et al. presents with a virtual worlds environment the possibility to create foreground masks, bounding boxes and target centroids in top-view omnidirectional images [44].

3. THEODORE Dataset

In this section all relevant steps for the generation of our synthetic dataset THEODORE are presented. We show the properties of the dataset in terms of distribution and variations of viewing angle.

3.1. Data Generation

Game Engine An advantage of the usage of a game engine, compared to rendering software like Blender, is the opportunity to generate data in a less time-consuming manner. In this work we are generating synthetic data using the game engine Unity3D. To configure the walking path of the characters in the virtual environment, Unity3D provides a NavMesh component that allows avoiding obstacles by approximating the walkable areas.

The generated virtual environments consist of indoor scenes, where typical objects like tables, sofas and chairs are placed at fixed positions. Human 3D characters are generated using the Skinned Multi-Person Linear Model (SMPL) [30]. SMPL is a model of the human body with focus of realism based on thousands of 3D body scans. Human characters are able to move randomly in the area determined to be valid by NavMesh. Each character moves from a random start position to a randomly selected object position as destination.

We need to capture the whole scene from an omnidirectional camera placed on the ceiling of the room. However, Unity3D only provides a camera model for perspective and

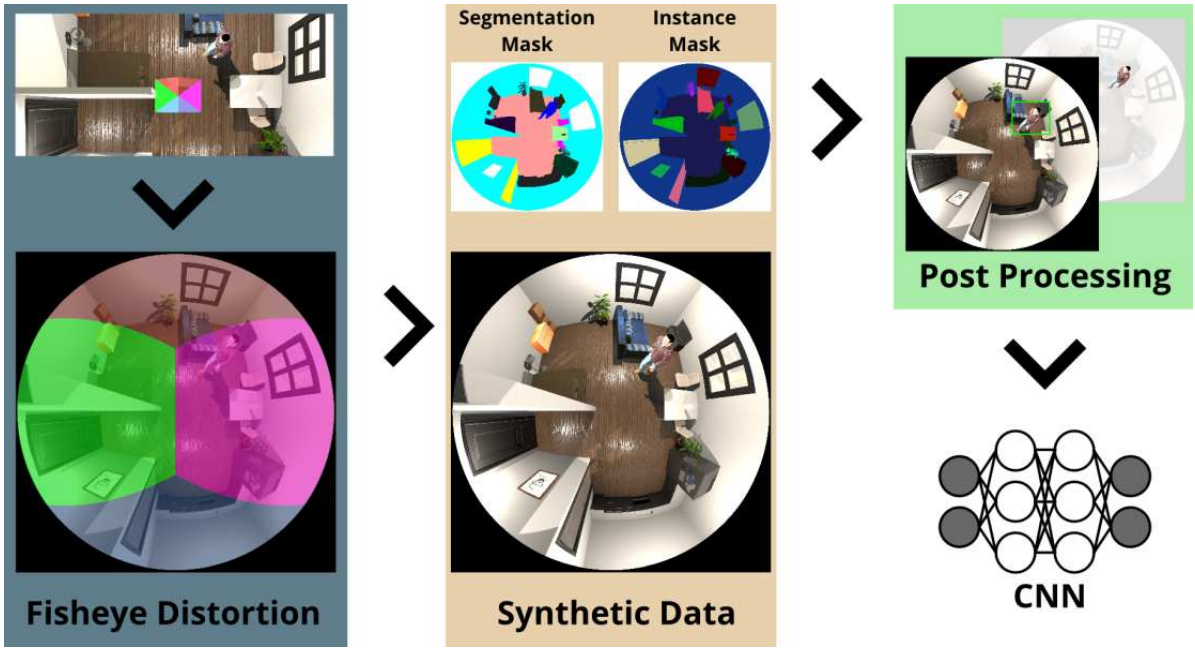


Figure 1: Pipeline showing the image generation for THEODORE. Starting with four cameras pointing in top, left, right and bottom direction with a field of view of 90 degrees per camera, a fisheye distorted image is generated. In addition to the rendered RGB image, instance and segmentation masks are created, distorted and saved as training data. In post-processing, bounding boxes are extracted and converted into common dataset formats such as TFRecord.

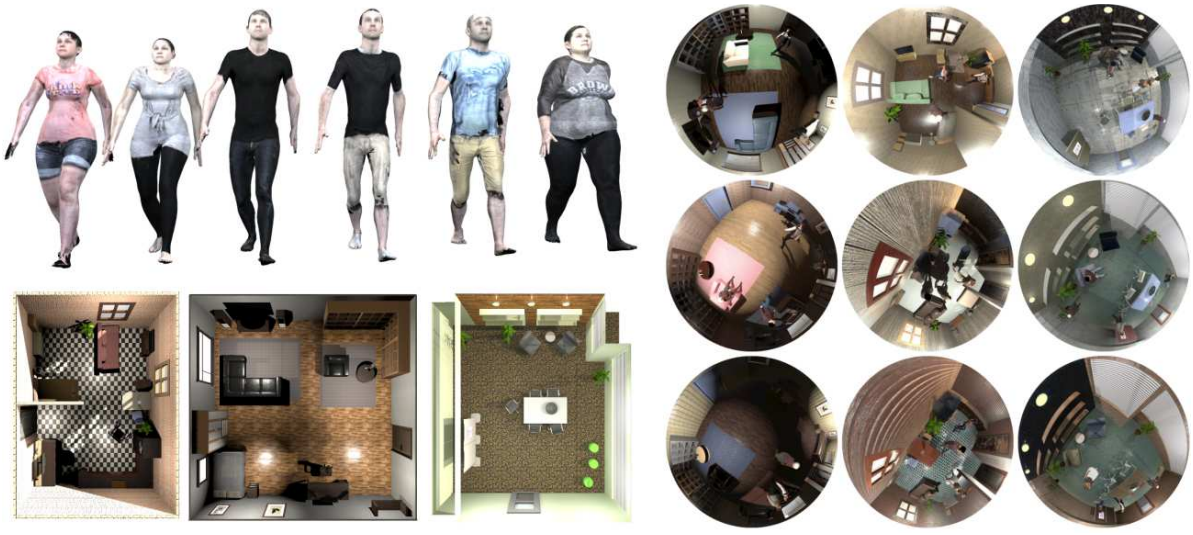


Figure 2: Overview of random characters and room floor plans used in THEODORE. Additionally, some sample images with the applied domain randomization are depicted. For each room three random textured scenes with random camera positions are selected.

orthographic projection. This limitation can be overcome by combining four perspective cameras in order to generate an omnidirectional image as described in the following.

Fisheye Projection Real omnidirectional images can be obtained by using fisheye lenses which results in a barrel distortion. Inside Unity3D fisheye images can be generated

using the approach described by Bourke *et al.* [3, 4].

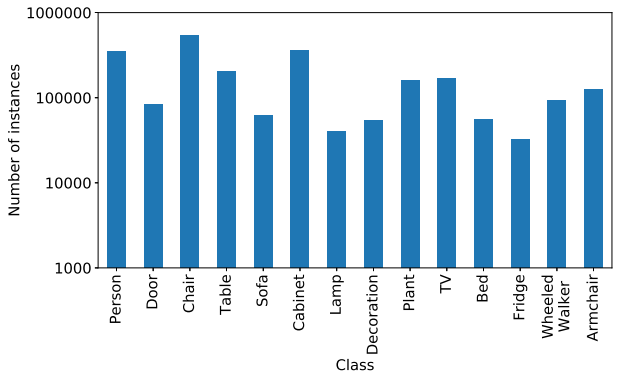
This method is based on a modified cube map rendering (see Figure 1) using 4 of the cube faces to form a fisheye distorted image. Each face is the result of a rendered image captured by a camera with a field of view of 90 °. As shown in Figure 2, the final result is created by warping and combining these images on four meshes, whose texture coordinates model a fisheye projection. Afterwards the generated and distorted fisheye image is captured with an orthographic camera and rendered to the display.

For THEODORE we are using a resolution of 1024×1024 pixels which allows us, in combination with a native plugin, to reach an output of 15 FPS on an Intel i7-7700 and a Nvidia GTX 1080. The native plugin allows us to manage the transfer of the textures from GPU to the CPU memory in a faster way than the conventional methods available in Unity3D.

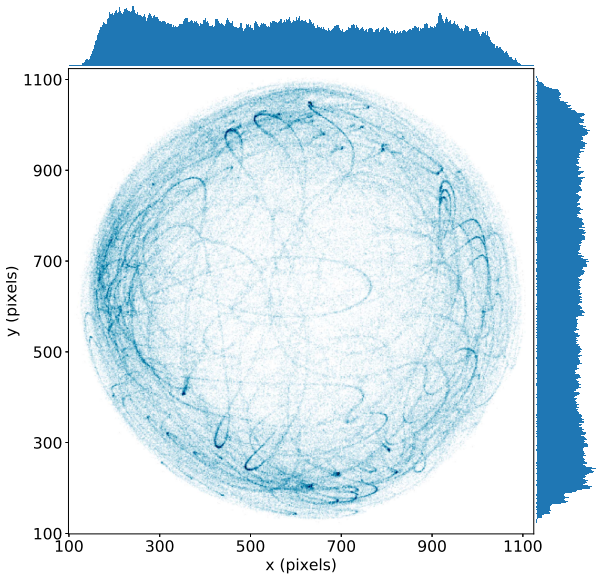
Image Synthesis In addition to a rendered image we are generating segmentation- and instance masks. This is done by cloning the virtual omnidirectional camera setup and replacing the assigned shaders of each object with a unique color shader. In the case of segmentation, the colors are assigned according to the object classes. For instance masks the color is selected based on the unique object ID. With these modifications, the approach presented in [1] fits the previously described fisheye projection. In this case the shader replacement is performed for all four perspective cameras before generating the final segmentation and instance masks.

Domain Randomization An approach for bridging the gap between synthetic and real images is domain randomization [45, 46] that we also apply in our implementation. Every room changes after 25 seconds, which we call a level change. With each level change a new room is selected and object textures are randomly replaced. Furthermore, human characters are generated with random parameters (like height or weight) and textures, using the texture set from [47]. However, the replacement occurs inside a predefined texture set (e.g. wood, concrete, cotton, etc.), to prevent inappropriate texture assignments. Additionally the camera position is changed over time in order to create different points of view. The trajectory of the camera follows a Lissajous curve. Light sources are defined as point lights with a fixed range and intensity. The number of enabled light sources in each room is selected to ensure a well illuminated scene. To create different lighting situations with each level change some randomly selected light sources are disabled, however with the restriction that at least one light source remains active.

Post Processing The final image, the segmentation and instance masks are combined in order to extract the necessary bounding boxes for the CNN training. By segmenting per color on the instance mask, a binary mask for each object is generated. Then these masks are applied on the segmentation mask to identify each object with its corresponding label and the bounding box coordinates x_{min}, x_{max}, y_{min} and y_{max} are calculated. Finally the fisheye images together with the extracted bounding boxes and their corresponding labels are used to perform a conversion into common dataset formats (e.g. TFRecords [20] or PASCAL VOC2012 [14]).



(a) Instances per class



(b) Distribution of centroid location of a person

Figure 3: In Figure 3a we show the number of annotations per class. In Figure 3b the distribution of centroid location of a person over all images is illustrated. See text for details.

3.2. Dataset Analysis

The creation pipeline of our synthetic omnidirectional data is visualized in Figure 1. Apart from the final image we extract the segmentation and label mask from the rendering process. Based on these masks we are able to select specific objects to calculate the bounding box. For THEODORE

we have exported 100k images and bounding boxes for the classes person, chair, table, armchair, wheeled walker, tv generated. The dataset contains different rooms with randomly selected textures, as described in section 3. For this approach we downloaded and categorized 120 textures[1], so that each textured 3D object can theoretically choose one of them. We recorded the scene with 8 frames per second. In combination with a level change parameter of 25 seconds and the texture randomization, the dataset contains about 500 various textured indoor scenes.

An example for three randomized men and women with varying body shape and height, wearing different clothes and additional attributes of our simulation is depicted in Figure 2. The amount of instances per class is visualized in Figure 3(a) and the statistical distribution from the center point of persons bounding box in Figure 3(b). Through camera movement and random selection of destinations for a person we ensure well distributed positions over all fisheye images.

4. Approach

In this section we describe the functionality of three meta-architectures of CNNs for object detection and two semantic segmentation networks and show the corresponding training setup. We train the architectures with our synthetic data using an open source framework for object detection [20] and a own implementation for the segmentation task. For object detection task we choose one- and two-stage object detectors and for segmentation we use pixel-wise classifiers as following described:

Faster R-CNN The Faster R-CNN architecture uses two stages for the detection. The first stage, the region proposal network (RPN), is used to predict and extract box proposals. For this stage a feature extractor is used to extract the features of an image at various intermediate maps. In the second stage proposed boxes are cropped from an intermediate feature map and fed to the remainder of the feature extractor to refine and predict classes of the box proposals. As feature extractor we use ResNet50 [19].
R-FCN R-FCN is similar to R-CNN. The difference is in the cropping approach. An R-FCN only crops the result of the last layer while a Faster R-CNN crops the features from layers where the region proposal is predicted. This reduces the pro-region computation because cropping happens only at the end of the network and results in a faster run time. Here, ResNet101 [19] is used as feature extractor.
Single Shot Detector The Single Shot Detector (SSD) uses a single feed-forward convolutional network to predict box anchors and classes directly without using a second stage per-proposal classification. The final detections are the results of a non-maximum suppression step applied to the prior predictions. For our approach we use the feature pyramid network (FPN) [27] implementation of ResNet50.
SegNet SegNet [2] is a CNN architecture for semantic pixel-wise segmentation. It consists of an encoder network which topology is identical to VGG-16 network [40]. However, fully connected layers were removed to improve the size of the network and the training process. The decoder network of SegNet restores the gradually reduced spatial dimension from the encoder network. To realise this, SegNet uses pooling indices of the corresponding encoder for a non-linear upsampling.
PSPNet Pyramid Scene Parsing Network (PSPNet) [49] is a scene parsing framework that uses a pyramid pooling module to aggregate different regional contexts. This modules is appended to an pre-trained ResNet network, in our case a ResNet101. In addition, ResNet was modified to use dilated convolutions to enlarge the field of view.
Training All selected object detection networks are pre-trained on MS COCO [28]. As configuration for each architecture we use the proposed settings from the framework [20]. Adjustments are made on the training settings. For all experiments, a value of 0.9 for the momentum optimizer [34] is selected. We apply cosine decay [31] as learning rate strategy for the SSD meta architecture. As parameters we select a learning rate and warmup learning rate of $3\mathrm{e}{-5}$ over $20,000$ steps. The training for the R-FCN and Faster R-CNN is manually stopped if the performance on the validation set begins to saturate. As learning rate strategy we reduce the learning rate by a factor of 10 every $20,000$ steps. The input dimensions for all networks change to a 3-channel RGB image with a fixed resolution of 640×640 pixels and batch size is set to 16. For a better generalization of the fine-tuned model data augmentation methods are applied ([24, 43, 45]). We select random brightness, random contrast, random crop, random Gaussian noise and horizontal flip for all meta-architectures during the training. For the semantic segmentation approach we use pre-trained ImageNet weights for PSPNet to fine-tune the architectures. SegNet is trained from scratch without the usage of pre-trained weights. As in our object detection setup we use the momentum optimizer with an momentum of 0.9 and a learning rate of 0.001. Furthermore, the training uses a batch size of 4 and is done for 150 epochs. The data augmentation methods random noise, horizontal flip and brightness are applied during training.

5. Evaluation

With fine-tuning of CNNs with THEODORE we evaluate on labeled real world images by meta-architectures for object detection and segmentation as described in section 4. We choose publicly available real world datasets

[1] https://www.cc0textures.com

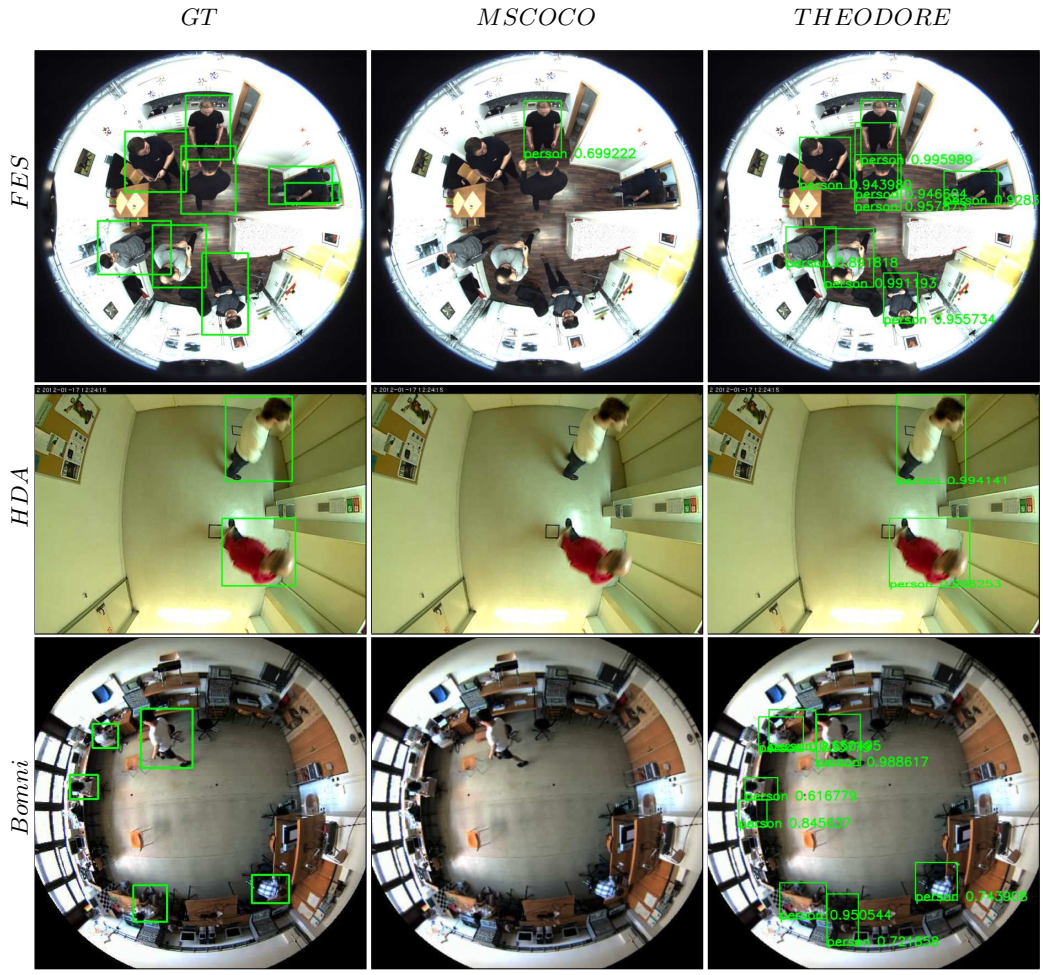


Figure 4: Example of detection results on HDA, Bomni and FES dataset using SSD meta-architecture for person class. The first column contains the ground truth bounding boxes. In the second column the prediction results for the pre-trained CNN with MS COCO weights is shown. The last column indicates the detection boxes which we reach while fine-tuning on THEODORE. Statistics of the dataset and the evaluation through object detection is provided in the supplementary material.

such as High-Definition Analytics (HDA) [32] and Boğaziçi University Multi-Omnidirectional (Bomni) [12] for object detection and an own annotated dataset Fisheye Evaluation Suite (FES) for semantic segmentation (see subsection 5.2) to validate the meta-architectures fine-tuned on THEODORE.

As metric for evaluation the average precision (AP) [14] per class and mean average precision (mAP) is reported for all classes. Detections will be judged to be true positive, if the intersection over union (IoU) between the detected and ground truth bounding box is at least 0.5. Our evaluation results shows exemplary bounding box detections in the first row of Figure 5. For the evaluation on semantic segmentation we choose the mean intersection over union (mIoU) for the classes *armchair*, *chair*, *person*, *table*, *tv* and *wheeled walker*. Exemplary results for semantic segmentation can be found in the second row of Figure 5.

5.1. Number of images

In order to evaluate the amount of images that are relevant for THEODORE, we measured the mAP for the SSD meta-architecture validated on FES and summarize the results in Figure 6. From the generated images we choose 12k, 25k, 50k and 100k images and train the SSD for $20,000$ steps each. With the parameters described in section 3 the 100k images contain 500 differently textured scenes in the training set.

In general, we carried out two approaches to reduce the number of images. First, we sub-sample the 100k images to keep the number of differently textured scenes constant. Second, the number of images is halved and consequently the number of textured scenes. We observe that a increas-

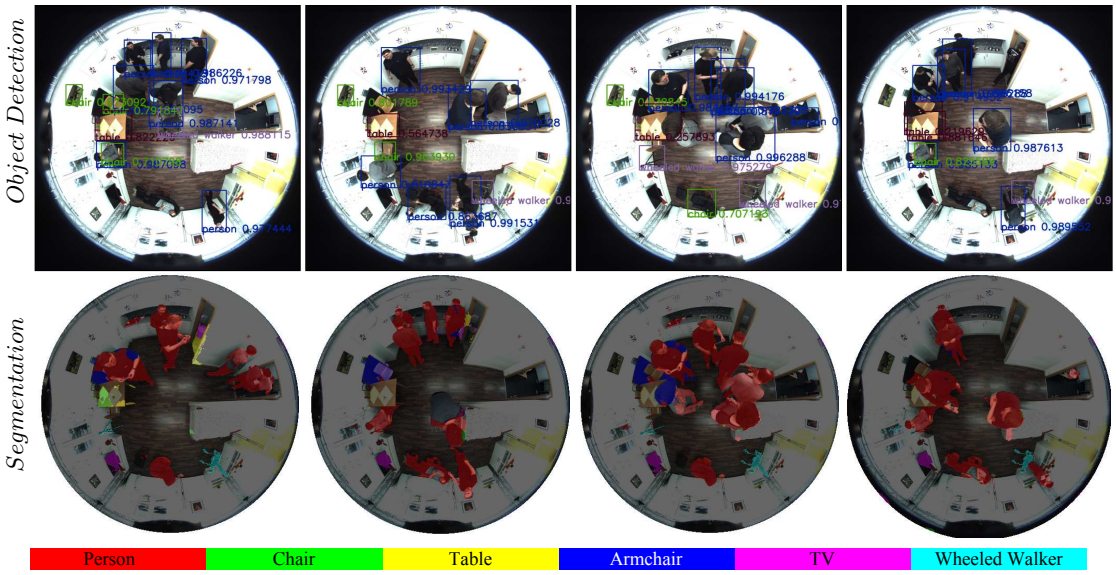


Figure 5: FES evaluation samples of object detection and segmentation architectures trained on THEODORE. The first row shows detection results from SSD, the second row the segmentation results of SegNet. SSD is trained on MS COCO and fine-tuned on THEODORE, while SegNet is trained from scratch on THEODORE.

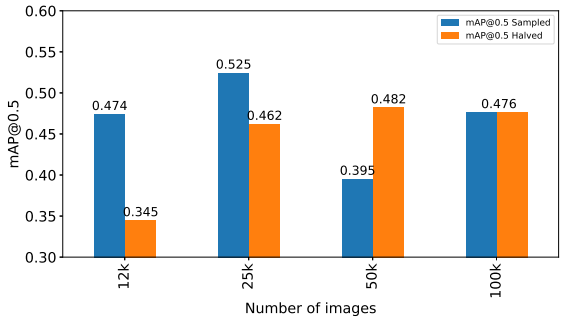


Figure 6: Number of images in the training set with corresponding mAP@0.5 trained on the SSD meta-architecture. Sampled: constant number of textures with variable number of images; halved: halved number of images and halved number of textures

ing number of images not necessarily leads to better results. The sub-sampled approach with 25k images results to the highest mAP. This experiment shows that the number of scene variations have a higher impact on mAP than the absolute number of images for training. For further experiments we use a subset of THEODORE with 25k images and 500 scenes.

5.2. Object Detection

In this section we describe the evaluation of THEODORE on three real-world datasets, the HDA dataset, Bomni and FES. The HDA and Bomni dataset only contains labels for the person class, so the evaluation was done on person class, as long as there are no other classes in the datasets publicly available. The evaluation on our own dataset (FES) was done on six classes.

Validation on HDA The HDA dataset [32] contains images captured with multiple cameras. The dataset was created for the research on high-definition surveillance. For our evaluation we use the 1388 labeled images from *Cam 02*. These images, with a resolution of 640×480 pixel were captured at 5 Hz from the top-view position with a full 140° field of view. The images of the HDA dataset are barrel distorted, which makes them more comparable to omnidirectional images.

Validation on Bomni Bomni Video Tracking Database contains video frames with a resolution of 640×480 pixel from an omnidirectional camera in a single room. The dataset was created in the context of human tracking and action recognition. For our evaluation we use all frames from top-view cameras of scenario #1 and crop them to a resolution of 480×480 pixel to remove most of the black borders.

Validation on FES To the best of our knowledge FES is the first dataset with real world fisheye top-view images. The dataset contains of 301 images and six class labels (*person*, *armchair*, *chair*, *table*, *tv* and *wheeled walker*) which were annotated manually. All images have a resolution of 1680×1680 pixel with overlapping persons. The images of the dataset, segmentation masks and bounding boxes are available at `https://www.tu-chemnitz.`

de/etit/dst/forschung/comp_vision/fes.

Table 1: Quantitative evaluation of THEODORE for person class based on AP@0.5

Person AP@0.5	MS COCO			MS COCO + THEODORE		
	HDA	Bomni	DST	HDA	Bomni	DST
SSD	0.586	0.052	0.484	**0.802**	0.579	**0.904**
R-FCN	0.303	0.069	0.525	0.694	0.675	0.849
Faster R-CNN	0.627	0.067	0.630	0.704	**0.740**	0.873

For the evaluation of THEODORE we report the AP for person class for the HDA, Bomni and FES datasets in Table 1. As baseline, we choose MS COCO in the left three columns, while the right three colums indicate the APs with fine-tuning on THEODORE. We achieve in all three meta-architectures for person class a significant improvement with THEODORE with respect to the baseline.

Table 2: Quantitative evaluation of THEODORE for all classes based on mAP@0.5

Class AP@0.5	Armchair	Chair	Person	Table	TV	Wheeled Walker	mAP
SSD	0.021	**0.231**	**0.904**	0.824	0.545	0.623	0.525
R-FCN	**0.262**	0.039	0.849	0.859	0.000	**0.640**	0.441
Faster R-CNN	0.148	0.141	0.873	**0.980**	**0.943**	0.596	**0.613**

The mAP of experiments on FES with six classes are shown in Table 2. With 0.613 the highest mAP is reached with the Faster R-CNN. The per-class winners are highlighted bold in Table 2. The classes person and table have the highest APs which can explained through a good representation in the training data, i.e. various viewing angle and texture. Improvements needs to be done in the classes armchair, chair and TV. The low AP values can have different reasons. First, the objects in the training data have too less variations in terms of illumination, texture and viewing angle. Second, the training data doesn't fit well to the test data, which ends up with the creation of a more generalized model for these classes. Another effect we observed through the evaluation is the non-detection of the class TV with R-FCN. For this we suspect a too high shrinking of the image as input for the net, so the filter sizes are to big for the whole image to detect small objects with the R-FCN.

5.3. Semantic Segmentation

Beside object detection we show that THEODORE is eglible for training segmentation networks. Due to the lack of publicy available top-view fisheye label masks for evaluation of THEODORE we annotate own data, namely FES. The report of the class IoU and mIoU on two state of the art architectures for segmentation, SegNet and PSPNet, is shown in Table 3.

In Table 3 we evaluate THEODORE by fine-tuned SegNet and PSPNet on the FES. We observe class IoUs of 0.67 for person and 0.53 for TV. The mIoU lies at 0.36 for the SegNet and 0.23 for the PSPNet. Both segmentation architectures are realtively good in the class person, while classes like chair, armchair and table needs further investigations. We belief that the texture of the synthetically generated furniture is different from the real-world furniture texture.

Table 3: Quantitative evaluation of THEODORE by fine-tuning CNN meta-architectures for semantic segmentation.

Class IoU	Armchair	Chair	Person	Table	TV	Wheeled Walker	mIoU
SegNet	0.009	0.016	**0.674**	0.012	0.53	0.33	**0.359**
PSPNet	0.005	0.023	0.434	0.003	0.195	0.034	0.229

6. Conclusion

In this paper we introduce *THEODORE - a synTHEtic tOp-view inDoOR scEnes* dataset with omnidirectional images. This dataset contains 100,000 rendered images of diversified indoor environments, segmentation and instance masks for 16 classes and bounding boxes for the person class. Additionally, we have shown that the usage of synthetically generated images could compensate the lack of real omnidirectional images during training of CNNs. We have addressed the task of object detection and semantic segmentation for evaluating the performance of state-of-the-art CNNs trained on THEODORE. The evaluation process of our dataset works as follows: the training baseline is MS COCO, which contains front-views of perspective images. We fine-tune three meta-architectures for object detection, namely SSD, R-FCN and Faster R-CNN for the person class on THEODORE. In addition we train two meta-architectures for semantic segmenation, the SegNet and PSPNet for six classes in an indoor environment. Both object detectors and segmentation approaches were evaluated on our own annotated fisheye evaluation suite dataset (FES), that contains segmentation and object detection ground truth for six classes. With this we have shown the adaptation of the front-view to the top-view by fine-tuning CNNs with our generated data. While labels for fixed objects are not available in public real world databases, we use six classes for evaluation of THEODORE, which leads to significant improvement of the AP and mIoU over the baselines in all tested meta-architectures.

Future research will address the balancing of the classes of THEODORE. While the FES evaluation dataset only contains one scenario, we plan to add more real world indoor scenes. Beyond the segmentation and detection masks we intend to create omnidirectional depth, skeletons and optical flow ground truth from rendered scenes.

References

[1] Unity-technologies: Image synthesis for machine learning - bitbucket. https://bitbucket.org/Unity-Technologies/ml-imagesynthesis. (Accessed on 02/26/2019). 4

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 5

[3] P. Bourke. idome: Immersive gaming with the unity3d game engine. volume 9, pages 265–272, 2009. 4

[4] P. D. Bourke and D. Q. Felinto. Blender and immersive gaming in a hemispherical dome. volume 10, pages 280–284, 2010. 4

[5] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville. Home: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, 2017. 2

[6] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2

[7] I. Cinaroglu and Y. Bastanlar. A direct approach for human detection with catadioptric omnidirectional cameras. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 2275–2279. IEEE, 2014. 2

[8] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling. Spherical cnns. In *International Conference on Learning Representations (ICLR)*, April 2018. 2

[9] B. Coors, A. Paul Condurache, and A. Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[10] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 379–387. Curran Associates, Inc., 2016. 2

[11] C. R. del Blanco and P. Carballeira. The piropo database (people in indoor rooms with perspective and omnidirectional cameras). https://sites.google.com/site/piropodatabase/, unpublished dataset, 2016. 2

[12] B. E. Demiröz, İ. Ari, O. Eroğlu, A. A. Salah, and L. Akarun. Feature-based tracking on a multi-omnidirectional camera dataset. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–5. IEEE, 2012. 2, 6

[13] A. Eichenseer and A. Kaup. A data set providing synthetic and real-world fisheye video sequences. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 1541–1545. IEEE, 2016. 2

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. 4, 6

[15] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino. The hda+ data set for research on fully automated re-identification systems. In *European Conference on Computer Vision*, pages 241–255. Springer, 2014. 2

[16] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 2

[17] K. Genova, M. Savva, A. X. Chang, and T. Funkhouser. Learning where to look: Data-driven viewpoint set selection for 3d scenes. *arXiv preprint arXiv:1704.02393*, 2017. 2

[18] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016. 2

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. 5

[20] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3296–3297, July 2017. 4, 5

[21] C. Ionescu, F. Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2220–2227. IEEE, 2011. 2

[22] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014. 2

[23] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv*, 2017. 2

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5

[25] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018. 2

[26] D. Liciotti, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti. *Person Re-identification Dataset with RGB-D Camera in a Top-View Configuration*, pages 1–11. Springer International Publishing, Cham, 2017. 2

[27] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017. 5

[28] 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 D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1, 5

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detec-

tor. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. 2

[30] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2

[31] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[32] A. Nambiar, M. Taiana, D. Figueira, J. Nascimento, and A. Bernardino. A multi-camera video dataset for research on high-definition surveillance. *International Journal of Machine Intelligence and Sensory Signal Processing*, 1(3):267–286, 2014. 6, 7

[33] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[34] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999. 5

[35] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 2

[37] M. R. Ronchi and P. Perona. Describing common human visual actions in images. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 52.1–52.12. BMVA Press, September 2015. 1

[38] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv:1712.03931*, 2017. 2

[39] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3-Volume 03*, pages 32–36. IEEE Computer Society, 2004. 1

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[41] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[42] S. Song, A. Zeng, A. X. Chang, M. Savva, S. Savarese, and T. Funkhouser. Im2pano3d: Extrapolating 360 structure and semantics beyond the field of view. *Proceedings of 31th IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[43] R. Takahashi, T. Matsubara, and K. Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 786–798. PMLR, 14–16 Nov 2018. 5

[44] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 2

[45] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, Sep. 2017. 4, 5

[46] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, June 2018. 4

[47] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 2, 4

[48] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 2

[49] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5