A generative model for the creation of a synthetic dataset for semantic segmentation

Mladen Vidović*, Nebojša Nešić*, Ivan Radosavljević*, Aleksandra Mitrović*, Đorđe Obradović*
* Singidunum University, Belgrade, Serbia
mvidovic@singidunum.ac.rs, nnesic@singidunum.ac.rs, iradosavljevic@singidunum.ac.rs,
amitrovic@singidunum.ac.rs, djobradovic@singidunum.ac.rs

***Abstract*—The acquisition of large, annotated image datasets, required for the training of semantic segmentation models, is often an arduous task. This is because of the time-consuming, complicated and error-prone nature of the process of manual image labelling. This process also often requires specialized software and domain knowledge. These problems can be circumvented by utilizing a generative model to create synthetic automatically labelled datasets. In this paper, we propose a generative model in the form of a 3D scene, representing an urban environment. A virtual camera setup is used to acquire labelled images from the virtual urban environment. Each image is stored as a multi-channel EXR file, containing RGB data as well as an additional channel for each object class. These channels contain binary values which indicate whether a pixel belongs to the target class. These images are used to form a dataset for the training of semantic segmentation models. The viability of the generated dataset is evaluated by testing the trained semantic segmentation model on real world manually annotated images.**

I. INTRODUCTION

Semantic segmentation is the process of assigning class labels to individual pixels in an image or video frame. It has several applications, such as autonomous driving [1, 2], industrial inspection [3, 4], medical imaging analysis [5, 6], classification of terrain visible in satellite imagery [7, 8] etc. The main challenge of training supervised semantic segmentation models is the lack of openly available labelled data, due to the error-prone and time-consuming nature of manual annotation. Additionally, some of the available datasets are only applicable in certain scenarios such as semantic segmentation of urban environments. This is due to the recent interest in autonomous driving, as well as the comparative simplicity of acquisition of such data. Some examples of manually labelled datasets commonly used for urban environment semantic segmentation include the Cityscapes Dataset [9], KITTI [10], COCO [11], and CamVid [12, 13]. The main issues of these datasets are the lack of a sufficient number of fine annotations, a lack of visual variety, such as depictions of scenes in different weather and lighting conditions, as well as class imbalance, inherent to manually acquired data. One proposed solution that could alleviate the aforementioned shortcomings is the generation and utilization of synthetic datasets. In order to generate synthetic datasets, a generative model, which is the virtual representation of a real-world scenario, needs to be constructed. The advantages of this approach include the simple and fast acquisition of large amounts of labelled data, the complete control over the generated data parameters, allowing for the creation of diverse and balanced datasets.

In this paper, we propose a generative model for the creation of a synthetic dataset for semantic segmentation. The generative model represents an urban environment, populated with static objects such as road networks, buildings, roadside objects and vehicles. The data generated with this model are photorealistic, automatically labelled images. Along with the images, the model outputs metadata containing additional information about lighting and weather conditions present in the rendered scene, which can be useful for the study of the impact of these conditions on the performance of semantic segmentation models.

In the second chapter, we provide an overview of existing research in this field, as well as an analysis of similar solutions. In III, we present the construction of our generative model, and the process of dataset generation as well as a statistical overview of objects present in the virtual world. In IV, samples of the generated dataset, evaluation results, and statistical data of the generated dataset are shown. Finally in V, we conclude with a discussion of the achieved results and possible future improvements and research.

II. RELATED WORK

There are several papers related to the generation of synthetic datasets for semantic segmentation. One notable example is Synthia [14], a dataset generated from a virtual world created in the Unity[1] game engine. The virtual word is constructed by manually placing predefined blocks of assets representing elements of an urban environment, such as buildings, roads, vegetation, lamp posts etc. The virtual world is also populated by dynamic objects such as cars, cyclists, and pedestrians. The simulation of different seasons and weather conditions is supported in order to provide greater visual variability without the need for the construction of different locations or assets. Additionally, a dynamic illumination engine is utilized to produce different illumination conditions, such as sunny or overcast weather. This also allows for the simulation of dynamic shadows cast by objects within the world. The authors present two distinct datasets generated by utilizing the described virtual world. The first dataset consists of images of the city taken by virtual cameras placed randomly throughout the city. The second dataset consists of four video sequences which were created by placing virtual cameras on a virtual car which is driving through the city and interacting with other dynamic objects. Pixel-level semantic segmentation annotations are created automatically at the time of rendering by using unique identifiers assigned to each object type. The authors evaluate the quality of the acquired data by comparing the

[1] https://unity.com/

performance of models trained with only synthetic data and validated with real world data to models trained with a combination of synthetic and real data and validated with real world data. From this they conclude that the use of synthetic data in combination with real data provides better results than the usage of real or synthetic data alone. The boost in performance is most significant on classes which rarely appear in the real-world data, but are more evenly distributed in the Synthia dataset, thus exploiting the inherent imbalance of manually acquired real-world datasets. The metadata provided in the Synthia datasets consists of only the semantic segmentation labels and does not include any data describing the scene conditions such as weather effects or time of day, thus limiting the possibility of analyzing the impact of these properties on semantic segmentation quality.

Another example using the Unity game engine is the Virtual KITTI dataset [15]. It is based on five videos available in the original KITTI dataset. These videos were used to reconstruct the real world in the Unity game engine. Most of the construction is automated by extracting positions of objects relative to the position of the camera in the source videos. Manual adjustments were necessary in the case of a change of the width of the main road, as the road was generated by placing a fixed width road along the trajectory of the vehicle. Side roads, as well as background objects lacking positional data were placed manually. After the scenes were created, they were used to recreate the source videos in the virtual world as well as generate new videos by altering weather and lighting conditions. The position, number, speed, size and trajectory of vehicles can also be randomized to increase visual variability. The camera position, orientation and path can also be modified. Each moment of the scene is rendered four times. First, an RGB image is rendered using the Unity rendering engine. The second rendering pass is used to generate a depth map. The third pass produces an image containing ground truth for category and instance level segmentation. Finally, in the fourth pass, the optical flow for image is calculated. The generated dataset was evaluated by comparing the MOTA scores of a model trained only on the original KITTI dataset, only on the Virtual KITTI dataset, and a model trained on the Virtual KITTI dataset and then fine-tuned with real KITTI data. The results of the evaluation show that, while training only on virtual data is inferior to training on real data, combining real and virtual data results in the best performance. In a follow-up paper [16], the authors present an improved version of the Virtual KITTI dataset, that utilizes a stereo camera setup and contains backward and forward optical and scene flow as well as metadata containing camera parameters, vehicle colour, pose and bounding boxes.

In paper [17], the authors present the Virtual Environment for Instance Segmentation (VEIS) and a dataset generated from this environment. The environment is constructed in the Unity game engine by manually placing 3D objects available in the Unity asset store. Two types of scenes were generated: a multi-class scene representing an urban environment populated with objects from different classes, and a single-class scene containing one or more instances of the same object in different poses with a varying background. The images rendered from these scenes are then automatically annotated during rendering. Both instance and class-level annotations are generated. Along with the dataset, the authors propose the separation of foreground and background objects during training. They argue that this approach yields better results as the background objects in synthetic datasets have more realistic textures whilst foreground objects tend to have more realistic geometry but textures lacking in detail. In order to evaluate their proposed method, the authors train their model on different synthetic datasets and validate the model on the CamVid dataset. They show that their method outperforms training a standard semantic segmentation network using synthetic data and state-of-the-art domain adaptation techniques.

In [18], the authors present ProcSy, a synthetic dataset with the purpose of studying influence factors on of semantic segmentation models. The authors generate a 3D scene from real world data obtained from OpenStreetMap[2], municipal databases containing the heights of buildings. This data is complemented by satellite imagery which is used as a reference for manual edits. Lane-level data, such as number of lanes, street or lane width or heights of overpasses and highways, was not openly available so the authors manually restructured road networks to account for the missing details. The generated 3D environment was then populated with static vehicle and pedestrian models and exported to Unreal Engine 4[3] in order to utilize its rendering capabilities. The open-source driving simulation platform CARLA[4] was utilized to streamline the process of dataset generation. The dataset was generated by randomly placing a camera in the static 3D scene. Due to the random placement of the camera, it was possible to obtain frames with collision issues such as the camera being placed partially or fully inside static objects, thus obstructing the view. This issue was solved by manual inspection and filtering of the generated images. The annotations for instance and class-level semantic segmentation were automatically generated at the time of rendering. The instance level labels are provided only for vehicles and not for other objects in the scene. As the purpose of the paper is studying the effect of different weather conditions on the performance of semantic segmentation models, the option to simulate rain, clouds and puddles in the images was implemented.

Richter et al. [19] present a solution utilizing an existing commercial video game. They used a virtual world provided by GTAV, an open world sandbox game, to extract images that are later manually labelled. The images are extracted using a wrapper library that collects every 40th frame as well as information about resources being used to render that frame. The collected data contains camera effects and UI elements which must be filtered out, as they are specific to the video game and do not have any real-world counterparts. In order to uniquely and consistently identify resources in the frame the hash value of a resource in the GPU memory is calculated and used as an identifier. These identifiers are then associated to pixels in a rendered frame and are used as coarse labels that are then manually transformed into semantic segmentation labels. The transformation process is carried out using a purpose-built interface which allows users to associate resource IDs with semantic segmentation labels. Additionally, this interface provides means to propagate labels on consecutive frames and to extract rules with which it can automatically label previously unseen frames,

2 https://www.openstreetmap.org/

3 https://www.unrealengine.com/

4 https://carla.org/

significantly decreasing the time it takes to define semantic segmentation labels for an entire dataset. A dataset which was created and annotated in the described manner, was evaluated by comparing results of semantic segmentation models trained on real-world data, data acquired from the video game, and a combination of the two. The evaluation shows that the performance of the models was worse when trained on synthetic data alone, but a combination of synthetic and real-world data proved to be the best performing. However, this can be attributed to the instances-per-class distribution imbalance of the real-world datasets.

An example of an approach that does not rely on video game engines is shown in [20]. The authors utilize a custom rendering engine that allows for the rendering of photorealistic images. The photorealism is mostly achieved by creating and using a sophisticated lighting setup that mimics real-world lighting conditions more accurately than the ones usually provided in video game engines. The scenes that are used as a generative model in the rendering process are procedurally generated based on several input parameters, specifying: road materials, road width, number of pedestrians and cars, time of day, weather conditions, of which only sunny and overcast are supported, etc. Along with the images, this approach generates class and instance-level segmentation as well as object detection labels, and metadata for each rendered image. The annotations for class and instance-level segmentation are based on the classes defined in the Cityscapes dataset, whilst the metadata contains information about the camera setup, scenario parameters, and instance metadata, including 2D and 3D bounding boxes and classes associated with these bounding boxes. A model trained on a combination of the Synscapes and the Cityscapes dataset proved to outperform a model trained only on Cityscapes or only on Synscapes data. The authors also provide a study of the impact of scenario parameters on the performance of semantic segmentation models, in which they show that the time of day, as well as motion blur have the strongest impact on performance.

III. METHODOLOGY

A. *Scene setup*

The generative model, used to generate the synthetic data, was created as a 3D scene in Blender[5], an open-source 3D modelling software. The scene represents an urban environment, populated with static assets. All assets, excluding cars, were created specifically for the scene, some manually and others procedurally. The car assets were downloaded from various public domain sources and modified to fit our use case. The manually created assets were modelled using standard hard surface modelling. The procedurally generated assets were created using particle systems. They include horizontal vegetation, foliage, and trees, with the tree trunks being generated using parametrized curves. A mix of physically based rendering (PBR) textures and procedural textures was used to create shaders for the scene. The shaders themselves are parametrized for easy manipulation allowing them to play an important part in simulating differing weather conditions as well as a day/night cycle. HDRi environment maps were utilized to light the scene. The HDRi maps were grouped into categories, simulating lighting conditions for different times of day. Additionally, secondary lighting is generated from some of the assets themselves, such as windows, car headlights, and streetlights. The lights from those assets are toggled on or off, based on the selected time of day. Different weather conditions are implemented in the form of rain and fog. The rain effect is procedurally implemented through the shaders themselves as well as post render compositing. A procedural Musgrave texture is added to the road shader to drive the roughness values and simulate wetness of the road surface. The Musgrave parameters are randomized with each render to avoid repetition. Ripples in the generated puddles are procedural as well, created by subtracting two Voronoi textures with differing size attributes to generate the procedural ring texture which is then added on top of the already setup roughness. The rain droplets are added in post processing and are procedurally generated and randomized on every render. Fog is simulated by using Blender's internal mist pass which outputs depth parameters relative to the distance from the camera. These parameters are then used in post processing to create a fog mask that is overlaid over the final render. The fact that the weather conditions are added in post processing allows for the generation of images with different weather conditions without the need to render each image separately, thus reducing the time needed to generate a large and diverse dataset. The scene is rendered in two passes. The first pass produces an RGB image. The second pass generates labels for semantic segmentation in the form of binary images, one for each class in the scene. Each binary image contains pixels of objects belonging to the same class. In order to associate objects with classes, a unique class ID parameter is assigned to each object that is represented in the scene. To be compliant with Cityscapes segmentation labels, holes in the generated labels had to be avoided. This was achieved by enveloping objects containing holes, such as fences, foliage etc., with a modified convex hull. The hull was setup with a transparent shader, allowing it to be present on the mask pass without showing in the colour image render. In order for the transparent convex hulls to be visible in the final labels, all the objects in the scene are overridden with a non-transparent shader for the second rendering pass.

B. *Camera setup*

The synthetic images used to create the final dataset are captured using a virtual camera setup. The camera is placed alongside a predefined set of paths following the road network of the virtual urban environment, in such a manner as to avoid overlap with static assets. The camera can be rotated left and right and tilted clockwise and counterclockwise randomly, to generate a more diverse dataset.

C. *Rendering setup*

In order to realistically simulate physically accurate lighting conditions, the Cycles rendering engine, Blender's raytracing engine, is used for the rendering of the final image. The engine was setup to render images in 1920 x 1080. To optimize render times, whilst retaining a sufficient level of photorealism, the sample size was limited to 24, and the number of light bounces to 4. To eliminate noise in the rendered images, the Optix denoiser was used, as it is the fastest denoiser available in Blender.

[5] https://www.blender.org/

Optix is only supported for Nvidia GPUs, limiting the choice of hardware on which the dataset can be generated.

D. *Output generation*

The final output of the generation process is a set of Multilayer EXR files, each file containing all of the generated labels as well as the colour image. Additionally, metadata for each image is stored in a separate JSON file. The metadata consists of the name of the image, camera position, rotation and tilt, weather conditions, time of day, the utilized HDRi, and classes present in the image.

IV. RESULTS

The generative model described in the previous chapter was used to generate a dataset containing 5016 images. Examples of RGB images as well as annotations are shown in Fig. 1. In Table 1, we show the total number of annotated pixels per class and the average number of pixels per class in a single image. The uneven distribution of pixels per class can be explained by the different apparent sizes of objects, as larger and closer objects will have more annotated pixels. In Table 2, we present the distribution of images with respect to lighting and weather conditions. As is evident from the table, our generative model allows for a balanced distribution of weather effects across all images.

In order to evaluate the dataset, a semantic segmentation model [21] was trained with the generated synthetic dataset, the Cityscapes dataset, as well as a combination of both. The results of the evaluation are shown in Table 3. The results show that the model trained with the combined dataset performs better than the models trained with only artificial or only real data. The lack of performance improvement for some classes is attributed to the absence of those classes in the generated dataset.

TABLE I. NUMBER OF ANNOTATED PIXELS PER CLASS

	Pixels	
	Total	Average per image
Buildings	3418105844	680355
Road	2452824088	488221
Curb	1789823644	356254
Cars	1193801296	237619
Sky	516782872	102862
Trees	396978344	79016
Poles	274463228	54630
Void	153080356	30469
Road marking	105015012	20902
Crosswalk	40088876	7979
Traffic lights	38786820	7720
Vegetation horizontal	23298524	4637
Vegetation vertical	10264416	2043
Road signs	4453080	886

TABLE II. DISTRIBUTION OF IMAGES WITH RESPECT TO LIGHTING AND WEATHER CONDITIONS

	Day	Night	Total
Clear	418	418	836
Fog	418	418	836
Rain	418	418	836
Rain & fog	418	418	836
Wet	418	418	836
Wet & fog	418	418	836
Total	2508	2508	5016

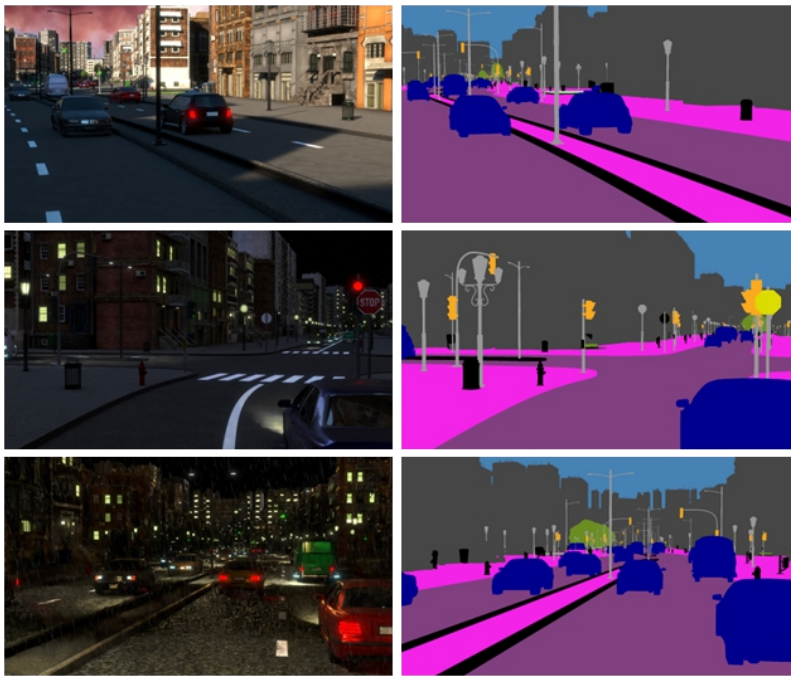

Figure 1. Rendered RGB images (left) and semantic segmentation labels (right)

TABLE III. COMPARISON OF THE PERFORMANCE OF THE MODEL TRAINED ON CITYSCAPES DATASET, THE GENERATED DATASET AND A COMBINATION OF THE TWO

	Cityscapes	Generated dataset	Cityscapes & generated dataset
Road	96.92%	97.76%	97.47%
Sidewalk	76.39%	95.53%	79.95%
Car	91.20%	94.74%	91.10%
Sky	92.30%	94.20%	93.15%
Building	88.17%	93.94%	88.97%
Vegetation	89.32%	76.36%	90.18%
Pole	45.48%	58.99%	52.20%
Traffic light	37.94%	44.48%	45.53%
Traffic sign	56.44%	41.71%	60.18%
Wall	46.83%	0.00%	33.20%
Fence	43.26%	0.00%	48.18%
Terrain	52.04%	0.00%	56.36%
Person	66.96%	0.00%	70.30%
Rider	46.19%	0.00%	48.51%
Truck	64.79%	0.00%	60.41%
Bus	62.35%	0.00%	56.89%
Train	40.25%	0.00%	22.16%
Motorcycle	28.21%	0.00%	52.74%
Bicycle	63.47%	0.00%	63.75%
mIoU	62.55%	77.52%[a]	63.75%

[a] Missing classes were excluded from the calculation.

V. CONCLUSION

In this paper we presented a generative model created in Blender and utilized for the generation of a synthetic image dataset. The generative model is a 3D scene representing an urban environment inspired by the Cityscapes dataset. The model allows for the generation of photorealistic images with varying lighting and weather conditions, increasing the visual variety of the dataset. The images are automatically labelled for semantic segmentation during rendering. We showed that this model generates an evenly balanced dataset in terms of weather and lighting conditions without a severe impact on render times.

The generative model can easily be expanded to include additional objects and object classes by adding new assets and assigning them a new ID, which is one of the planned future improvements. Improvements of the metadata extraction process could allow for other types of annotations to be generated and exported, making the dataset usable for the training of models for different purposes. Additionally, the generation of road networks and the placement of roadside objects can be automated in order to create a fully procedural generative model.

ACKNOWLEDGMENT

The authors acknowledge funding provided by the Science Fund of the 604 Republic of Serbia #GRANT No. 6524105, AI—ATLAS.

REFERENCES

[1] M. Treml et al., Speeding up Semantic Segmentation for Autonomous Driving. 2016.

[2] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for Semantic Segmentation in Street Scenes," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, Jun. 2018, pp. 3684–3692. doi: 10.1109/CVPR.2018.00388.

[3] A. Gheorghe, C. G. Amza, and D. Popescu, "IMAGE SEGMENTATION FOR INDUSTRIAL QUALITY INSPECTION," Fiability & Durability/Fiabilitate si Durabilitate, vol. 1 supliment, May 2012.

[4] J. Huang, L. Guixiong, and B. He, "Fast semantic segmentation method for machine vision inspection based on a fewer-parameters atrous convolution neural network," PLOS ONE, vol. 16, no. 2, p. e0246093, Feb. 2021, doi: 10.1371/journal.pone.0246093.

[5] R. Yang and Y. Yu, "Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis," *Front. Oncol.*, vol. 11, 2021, doi: 10.3389/fonc.2021.638182.

[6] B. V, "BIOMEDICAL IMAGE ANALYSIS USING SEMANTIC SEGMENTATION," JIIP, vol. 1, no. 02, pp. 91–101, Dec. 2019, doi: 10.36548/jiip.2019.2.004.

[7] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 150, pp. 59–69, Apr. 2019, doi: 10.1016/j.isprsjprs.2019.02.006.

[8] G. Bahl, L. Daniel, M. Moretti, and F. Lafarge, "Low-Power Neural Networks for Semantic Segmentation of Satellite Images," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), Oct. 2019, pp. 2469–2476. doi: 10.1109/ICCVW.2019.00302.

[9] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," arXiv:1604.01685 [cs], Apr. 2016, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1604.01685

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, doi: 10.1177/0278364913491297.

[11] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," arXiv:1405.0312 [cs], Feb. 2015, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1405.0312

[12] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and Recognition Using Structure from Motion Point Clouds," in Computer Vision – ECCV 2008, Berlin, Heidelberg, 2008, pp. 44–57. doi: 10.1007/978-3-540-88682-2_5.

[13] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," Pattern Recogn. Lett., vol. 30, no. 2, pp. 88–97, Jan. 2009, doi: 10.1016/j.patrec.2008.04.005.

[14] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 3234–3243. doi: 10.1109/CVPR.2016.352.

[15] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual Worlds as Proxy for Multi-Object Tracking Analysis," arXiv:1605.06457 [cs, stat], May 2016, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1605.06457

[16] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," arXiv:2001.10773 [cs, eess], Jan. 2020, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/2001.10773

[17] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, “Effective Use of Synthetic Data for Urban Scene Semantic Segmentation,” in Computer Vision – ECCV 2018, vol. 11206, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 86–103. doi: 10.1007/978-3-030-01216-8_6.

[18] S. Khan, B. Phan, R. Salay, and K. Czarnecki, “ProcSy: Procedural Synthetic Dataset Generation Towards Influence Factor Studies Of Semantic Segmentation Networks,” p. 9.

[19] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for Data: Ground Truth from Computer Games,” arXiv:1608.02192 [cs], Aug. 2016, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1608.02192

[20] M. Wrenninge and J. Unger, “Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing,” arXiv:1810.08705 [cs], Oct. 2018, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1810.08705

[21] Y. Nirkin, L. Wolf, and T. Hassner, “HyperSeg: Patch-wise Hypernetwork for Real-time Semantic Segmentation,” arXiv:2012.11582 [cs], Apr. 2021, Accessed: May 31, 2021. [Online]. Available: http://arxiv.org/abs/2012.11582