JOURNAL OF
EMERGING INVESTIGATORS

Article

Effects of different synthetic training data on real test data for semantic segmentation

William Zhang[1], Stephan Lemmer[2]
[1] St. Andrew's College, Aurora, Ontario, Canada
[2] Robotics Institute, University of Michigan, Ann Arbor, Michigan

SUMMARY

Semantic segmentation - labelling each pixel in an image to a specific class- models require large amounts of manually labeled and collected data to train. These huge datasets often are lacking in many fields such as specific weather events and climates. They also lack variety as most datasets are of city and street scenes as opposed to more rural scenes. Synthetic-to-real domain adaptation can be used to fill those gaps. Although there has been prior work on how to generate virtual data, in this paper we present insight on how features of the training data can affect the model and ultimately, the outcome. They can also provide insight on the dataset selection process. We hypothesized that different synthetic training data on real test data will affect the outcome of semantic segmentation. By testing real-world data on two iterations of a model trained by two separate virtual datasets using Python and deep learning models, we compared the results and analyzed the differences and similarities the datasets yielded for semantic segmentation. We unearthed critical insights that shed light on the dataset selection process, enabling researchers and practitioners to make more informed decisions when choosing the appropriate dataset type for semantic segmentation tasks. We contributed valuable findings, unveiling the limitations of substituting real datasets with virtual counterparts and offering guidance for dataset selection.

INTRODUCTION

Deep learning techniques in computer vision have progressed over the past few years and have shown huge success in many fields. To do so they usually are trained with datasets from the real world. There are numerous drawbacks to this, as real-world datasets usually are troubled with insufficient diversity of training material. Examples include the lack of everyday objects such as a bus in a driving dataset or the lack of rain or nighttime scenes in a weather dataset (1). Another issue is that for real-life data, the ground truth – information that is known to be true, in this case the "correct answer" for the segmentation – must be manually annotated by a human expert rater (1). This process can be extremely tedious, time-consuming, and expensive. To tackle these issues, more and more researchers have been creating and training on virtual datasets which can be produced using more cost-effective and faster methods (2).

We focused on semantic segmentation, the task of labelling each pixel in an image to a specific class (type of object). Methods of segmentation include variations of thresholding, such as: K-means clustering (grouping points by finding distances between them), compression-based (compressing image into foreground and background), and histogram-based methods (using a histogram to threshold the image into foreground and background) (3). Deep learning-based approaches usually involve some sort of weakly/semi-supervised (small amounts of labeled data with large amounts of unlabelled data) deep convolutional neural networks (CNN), while novel methods such as layered general adversarial networks (GAN) have been utilizing rendering in conjugation with CNNs (3,4). Methods in semantic segmentation (treating multiple objects within a single category as one entity) include U-Net, an approach using convolutional networks specifically for biomedical images, SegNet, a deep CNN method that focuses on pixel-wise segmentation and DeepLab, an improvement on previous methods that implements Atrous convolutions and lowers computational costs significantly (5-7).

Several studies have investigated training using out-of-domain data or training with virtual data. Toldo et al. proposed numerous unsupervised domain adaptation methods, while Tian et al. proposed the Parallel Vision framework for generating virtual images with accurate annotations (1,2). On the contrary, there is limited information regarding the differences in accuracy when training on different virtual datasets and testing using a real-world dataset.

For deep learning algorithms to work, with a focus on semantic segmentation, they must be trained with large amounts of data. Sufficient training data is not available for many domains and tasks regarding semantic segmentation. When that occurs one often uses a synthetic or virtual dataset. The virtual datasets used in this experiment are the GTAV and the Synthetic collection of Imagery, and Annotations (SYNTHIA) (8,9). The real-world dataset is the Cityscapes Dataset (10).

The Cityscapes Dataset is a real dataset that contains 5,000 images with fine annotations (more detailed) and 20,000 with coarse annotations (less detailed) that span across 50 cities and 30 classes. The images are urban street scenes with both vehicles and people. The images are diverse, covering several months and seasons, and a multitude of weather conditions. The dataset directory contains predetermined groups for testing and training. Along with the raw street image data it also contains human-annotated ground truth (correct answer) segmentations data.

The GTAV Dataset contains 24,966 densely labeled images that are generated inside the Grand Theft Auto Five video game. The annotations are generated by a program – bypassing the human aspect – and was completed in 49 hours, roughly three orders of magnitude faster than that of other datasets (8). The nature of the images is similar to that of the Cityscapes Dataset, with street view images of various conditions and information including both vehicles and people. The images are in png format, and the labels are the "ground truth" for which to compare results. Most GTAV and Cityscapes classes overlap, except for some. This leads to some classes being discarded in the final calculations.

The SYNTHIA dataset contains 9400 pixel-level labeled images generated from a European-style virtual city through the Unity development platform. The dataset contains 13 different classes that overlap with Cityscapes. The images contain multiple seasons and variable lighting and weather conditions, including day/night modes and rain (10). The data is separated into classes such as road, sidewalk, building, wall, fence, pole, light, sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle. There is overlap in all classes except terrain, truck, train, and fence.

We focused on the effects that differences in virtual datasets bring to synthetic-to-real domain adaptation in the form of semantic segmentation, a computer vision task that tries to label each pixel in an image to a corresponding class (3,11). We hypothesized that different synthetic training data on real test data will affect the outcome of semantic segmentation. We present a comparison of two virtual datasets when the models trained with them are tested with a dataset from real life. In addition, we analyzed the differences and features of the training datasets – especially their image content and perspective – to provide insights on how the training data might affect the task of synthetic-to-real domain adaptation. We hypothesized that training with different synthetic training data on real test data will affect the outcome of semantic segmentation. Our results show that GTAV has a slightly higher adjusted mean IoU but a much lower pixel accuracy than SYNTHIA.

RESULTS

The goal of this study was to test and compare the virtual datasets SYNTHIA and GTAV by testing on the real Cityscapes dataset. Our study consists of two experiments each using pre-trained models from pytorch-auto-drive and testing them to find the IoU (intersection over union) - as well as pixel accuracy (12). In the first experiment, we took a DeepLab 2 model trained on GTAV images and tested it with Cityscapes data using the PyTorch implementation. In the second experiment we took a DeepLab 2 model trained on SYNTHIA images and tested it with Cityscapes data using the PyTorch implementation. Our results show that the model trained on GTAV has a slightly higher adjusted mean IoU (38.33 vs 37.18) but a much lower pixel accuracy (66.05 vs 70.27) than SYNTHIA. (**Table 1 and 2**).

We present features that can be considered when choosing a dataset to perform domain adaptation by comparing SYNTHIA vs GTAV and training the same model, in this case, DeepLab2, and displayed the results when tested with Cityscapes. We found that the classes in both datasets have overlap, meaning that they have the same classes in both datasets. This ensures that the datasets have the ability to perform domain adaptation. We also found that features such as variable conditions and textures provide better results. A note of caution is that different perspectives (what position the photo/image is taken from) lead to better results in some classes and worse in others, classes that are more prone to perspective shifts will lead to worse results.

Class	Road	Sidewalk	Building	Wall	Fence	Pole	Light	Sign	Vegetation	
IoU(%)	42.01	18.87	46.80	14.01	21.69	30.15	36.13	15.31	83.01	
Pixel Acc(%)	43.99	30.97	94.55	17.47	35.52	34.05	39.23	15.44	91.65	
Class	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
IoU(%)	20.22	62.75	62.16	18.12	79.36	26.27	21.01	0.58	27.47	17.86
Pixel Acc(%)	64.99	71.76	72.01	21.25	89.17	36.29	26.53	0.6	31.83	18.39
Mean IoU: 33.88		Adjusted Mean IoU: 38.33			Mean Pixel Accuracy:66.05			Adjusted Mean Pixel Acc: 74.50		

Table 1. Individual class results of GTAV training set on Cityscapes. Data captured from python terminal. Means adjusted by removing classes with values of 0. Measured using Mean Pixel Accuracy and Intersection Over Union (IoU).

Class	Road	Sidewalk	Building	Wall	Fence	Pole	Light	Sign	Vegetation	
IoU(%)	52.04	25.33	56.47	5.42	0.01	35.49	19.6	15.83	77.66	
Pixel Acc(%)	54.44	65.83	94.46	5.75	0.01	47.29	20.5	16.42	84.27	
Class	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
IoU(%)	0	78.54	59.78	11.18	72.35	0	18.98	0	10.57	18.62
Pixel Acc(%)	0	90.63	83.97	12.46	85.87	0	31.71	0	10.85	18.96
Mean IoU: 33.88		Adjusted Mean IoU: 38.33			Mean Pixel Accuracy:66.05			Adjusted Mean Pixel Acc: 74.50		

Table 2. Individual class results of SYNTHIA training set on Cityscapes. Data captured from python terminal. Means adjusted by removing classes with values of 0. Measured using Mean Pixel Accuracy and Intersection Over Union (IOU).

For example, you can see the hood of the car in the last image reflects the sun and street (**Figure 1**). Both datasets have taken day/night, weather, and variable conditions into account, but GTAV has done so in a more photorealistic manner. This is in comparison to SYNTHIA's data which has almost uniform lighting conditions throughout, except for the presence of shadows (**Figure 2**).

Training a model with GTAV performed more accurately than SYNTHIA on the IoU metric, but worse on the pixel accuracy metric. We found that SYNTHIA performed better on classes that are larger - this caused the higher pixel accuracy. We also found that GTAV performed better dealing with classes that are typically smaller in the image.

DISCUSSION

Aside from the performance difference in each class that the two datasets had, we found that in testing the DeepLab2 models with Cityscapes, there were four classes unaccounted for while training with SYNTHIA and one class unaccounted for while training with GTAV. This gives GTAV 3 more classes of overlap with Cityscapes than SYNTHIA. The unaccounted-for classes do not exist within their respective datasets. We calculated the adjusted results with the unincluded classes (Terrain, Truck, Train, and Fence) omitted for both datasets (**Table 1 and 2**).

GTAV has a slightly higher adjusted mean IoU but a much lower pixel accuracy. This could be due to the GTAV dataset focusing more on images taken from inside a car and having reflections with more sophisticated lighting. SYNTHIA,

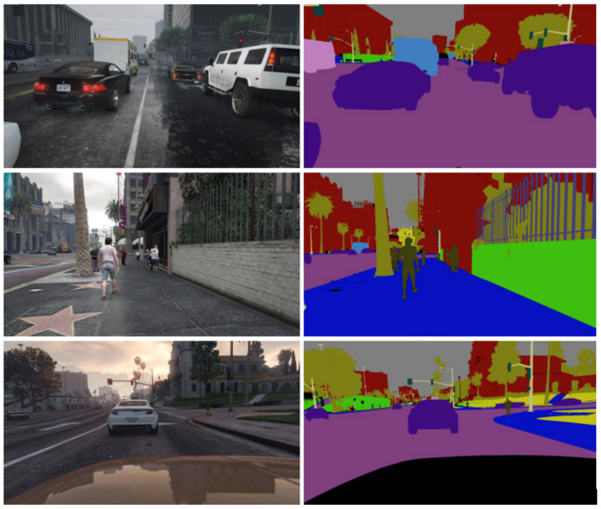

Figure 1. Representative screenshots from the GTAV Dataset. Ground truth masks are shown on the right side (8).

however, has more variety of perspectives taken and has less sophisticated textures and reflections (9). SYNTHIA also has a more saturated appearance when compared with GTAV. SYNTHIA performed significantly better on the larger classes of road, sidewalk, and sky. This could be that the “car perspective” that GTAV uses obscures many details of those larger tasks. This is also where the high pixel accuracy comes in. Because most of the pictures in SYNTHIA are made up of the road, sidewalk, and sky, an IoU of 25.33 of the sidewalks can get a pixel accuracy of 65.83. This class imbalance is why pixel accuracy might be misleading when used as a metric in semantic segmentation. GTAV performs better when dealing with the smaller, more detailed objects, but worse in the larger objects that take up most of the image. This causes the pixel accuracy disparity, even though the mean IoU values are relatively close, the pixel accuracy differs about 17%.

These results could be due to SYNTHIA being simpler, with clear outlines and boundaries. SYNTHIA also uses monotonous textures throughout the dataset. For example, the road texture inside one picture is the same throughout. GTAV textures are variable, with image distortion and discoloring on the reflections. (**Figure 1 and 2**).

Due to GTAV having the higher adjusted mean IoU, it is more accurate to use it to train for the Cityscapes dataset, as pixel accuracy can be deceptive. This finding, however, cannot be generalized to all datasets. Limitations in this study mainly relate to the limited access to computing power needed to run models. This makes it infeasible to train models, which is why pre-trained models are utilized in this study. Other limitations are that this paper only covers datasets involving street view, but there are countless datasets available. Depending on the intended usage case, one might opt for a specific dataset even if another is more accurate. With the different use cases for IoU and pixel accuracy, the ideal metric and dataset are dependent on the task of interest.

The results and their implications demonstrate the importance of how the training data affects synthetic-to-real domain adaptation, thus validating our hypothesis that different synthetic training data will affect the outcome of semantic segmentation. This paper provides new information and insight on dealing with data selection, often the first step of starting a project. As this field is relatively new, there has not been a lot of research in this area dealing with how differences in the training data affect the outcome. We can conduct further studies to analyze the effects of virtual data when combined with real data in the training process. Further studies might analyze the plausibility of training with multiple datasets or the benefits of training with a mix of virtual and real data. The innovations of Generative Adversarial Nets (GANS) by Goodfellow et al. and video game engines such as Unity make it easier than ever to create virtual datasets (13,14). There are severe drawbacks to human-annotated datasets, but they are the current standard due to semantic segmentation often being applied in the real world. Therefore, there needs to be a way to maximize efficiency by either finding the right balance of virtual versus real data or by creating virtual data that is indistinguishable from real data. Overall, the hypothesis that different synthetic training data on real test data will affect the outcome of semantic segmentation is validated. These results can lay the groundwork for future work to eventually generate guidelines to improve semantic segmentation dataset generation and selection.

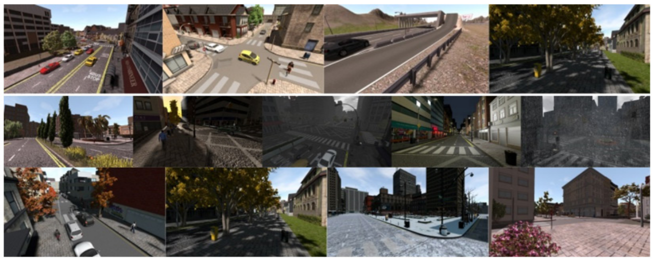

Figure 2. Representative screenshots from the SYNTHIA dataset (9).

MATERIALS AND METHODS

Model Used

A single model was used to perform all semantic segmentation tasks. The codebase, which was used to implement said task, is the pytorch-auto-drive, which includes trained semantic segmentation models and support for multiple datasets (12). The model used to implement said tasks is a PyTorch adaptation of Google’s DeepLab 2 model, a model that can be utilized to perform semantic segmentation, instance segmentation, depth estimation, and video panoptic segmentation tasks. DeepLab 2 is a deep neural network that builds on DeepLab with an atrous spatial pyramid pooling scheme (7). This removes the fixed-size input image and allows it to accept images of any size.

Measurement Metrics

The IoU also known as the Jaccard Index—metric was utilized to evaluate the performance of the semantic segmentation models. The IoU is the area of overlap between the prediction your model generates, and the ground truth divided by the area of union between the 2 images per class. The value ranges from 0% to 100% and is often displayed as 0 to 1. 1 symbolizes perfect overlap, while 0 symbolizes no overlap (15,16).

Another measurement metric for semantic segmentation is pixel accuracy. It is the percentage of pixels in your image that matches up with the ground truth. However, pixel accuracy often gives a misleading result when class imbalance is taken into account. For example, a 95%-pixel accuracy can be an entirely black image if the segmentation target is a small class that only populates 5% of the initial image. Due to this IoU is considered more accurate in some use cases and for the purposes of this paper we considered both metrics and tried to identify what causes the difference in results related to measurement metrics.

Received: March 28, 2022
Accepted: September 16, 2022
Published: June 22, 2023

REFERENCES

1. Tian, Yonglin, et al. “Training and Testing Object Detectors with Virtual Images.” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, Mar. 2018, pp. 539–46. doi:10.1109/JAS.2017.7510841.
2. Toldo, Marco, et al. “Unsupervised Domain Adaptation in Semantic Segmentation: A Review.” ArXiv:2005.10876 [Cs, Eess], May 2020. doi:10.48550/arXiv.2005.10876.
3. Yuheng, Song, and Yan Hao. *Image Segmentation Algorithms Overview*. arXiv:1707.02051, arXiv, 7 July 2017. *arXiv.org*, doi:10.48550/arXiv.1707.02051.
4. Yang, Yu, et al. “Learning Foreground-Background Segmentation from Improved Layered GANs.” ArXiv:2104.00483 [Cs], Dec. 2021. arxiv.org/abs/2104.00483.
5. Ronneberger, Olaf, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” ArXiv:1505.04597 [Cs], May 2015. arxiv.org/abs/1505.04597.
6. Badrinarayanan, Vijay, et al. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.” ArXiv:1511.00561 [Cs], Oct. 2016. arxiv.org/abs/1511.00561.
7. Chen, Liang-Chieh, et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs.” ArXiv:1606.00915 [Cs], May 2017. arxiv.org/abs/1606.00915.
8. Richter, Stephan R., et al. “Playing for Data: Ground Truth from Computer Games.” ArXiv:1608.02192 [Cs], 1, Aug. 2016. arxiv.org/abs/1608.02192.
9. Ros, German, et al. “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 3234–43. DOI.org (Crossref), doi:10.1109/CVPR.2016.352.
10. Cordts, Marius, et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 3213–23. DOI.org (Crossref), doi:10.1109/CVPR.2016.350.
11. “Image Segmentation” | Types Of Image Segmentation.” Analytics Vidhya, 1 Apr. 2019, analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/.
12. Feng, Zhengyang. Codebase for Deep Self-Driving Perception. 2019. 16 Feb. 2022. GitHub, github.com/voldemortX/pytorch-auto-drive/blob/5ac8e15bec0a486af1abf74df12e1e5fb0b5dfc5/docs/datasets/PASCALVOC.md.
13. Goodfellow, Ian J., et al. “Generative Adversarial Networks.” ArXiv:1406.2661 [Cs, Stat], June 2014. arxiv.org/abs/1406.2661.
14. Borkman, Steve, et al. “Unity Perception: Generate Synthetic Data for Computer Vision.” ArXiv:2107.04259 [Cs], July 2021. arxiv.org/abs/2107.04259.
15. Zhou, Dingfu, et al. “IoU Loss for 2D/3D Object Detection.” ArXiv:1908.03851 [Cs], Aug. 2019. arxiv.org/abs/1908.03851.
16. Ekin Tiu. Metrics to Evaluate Your Semantic Segmentation Model | by Ekin Tiu | Towards Data Science. towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2. Accessed 15 Feb. 2022.