ForestSim: A Simulated Dataset for Forest Scene Understanding

Pragat Wagle, Zheng Chen, and Lantao Liu

Luddy School of Informatics, Computing, and Engineering,
Indiana University, Bloomington, IN 47408, USA.
{pwagle, zc11, lantao}@iu.edu

Abstract. Semantic segmentation, a critical process in computer vision, involves classifying each pixel in an image into specific categories. This process is essential for machines to comprehend the visual world. The creation of large-scale datasets with pixel-level labels, necessary for training semantic segmentation models, is a challenging and labor-intensive endeavor, requiring meticulous annotation by humans, which can be time-consuming and expensive. While the field has seen the development of numerous datasets for structured environments, such as urban scenes with clear delineations and regular patterns, there is a noticeable scarcity of datasets for unstructured, off-road environments. These environments, such as forests, present unique challenges due to their irregular and complex nature. This scarcity is a significant hurdle for advancing computer vision applications in off-road autonomy, where understanding the natural, unstructured world is crucial for navigation and decision-making. To enrich the data sources for models in off-road environments, we opt to generate pixel-accurate semantic label maps from images acquired from a high-fidelity simulator. We present the resulting dataset, named ForestSim, which comprises finely annotated images obtained using a simulated ground vehicle. Over 10,000 images were collected and processed from 25 diverse environments, and 2094 of them were included in the final dataset, due to environmental diversity. Gathering and processing the data presented various challenges, for which we provide clear solutions and approaches. We evaluate ForestSim using state-of-the-art benchmark techniques and make our dataset, our processing pipeline, and associated code publicly available for further research and development.
Website: https://vailforestsim.github.io

Keywords: Dataset, Semantic Segmentation, Visual Navigation, Off-road, Forest

1 Introduction

The advancement of computer vision systems is intricately linked with the utilization of models trained on large image datasets. Image segmentation datasets play a vital role in contextual understanding within semantic segmentation tasks. It is recognized that data volume emerges as a decisive factor shaping the capability of deep learning models. With a burgeoning availability of data, the

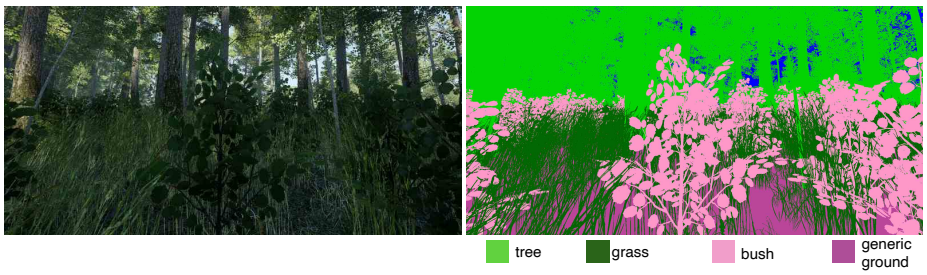

Fig. 1: An example image in ForestSim dataset. The size, shapes, and ratios are similar to those in the real world and demonstrate characteristics of an unstructured environment where the edges of objects are challenging to discern and almost blend into one another.

design and construction of more adaptable models and diverse architectures become feasible. Established large datasets have not only served as benchmarks but have also laid a robust foundation for advancing object detection, classification, and semantic segmentation, thereby fostering the development of benchmark datasets [1–4], particularly effective when objects in the data are well-represented within their collected environments [5].

The classification of large datasets varies depending on their intrinsic characteristics. One popular distinction scheme categorizes image datasets into structured and unstructured datasets. Structured datasets, reminiscent of urban environments with clear boundaries, vehicular traffic, and common shaped buildings [6–10], have been crucial in constructing high-performing semantic segmentation models. Furthermore, semantic segmentation datasets are employed to augment existing models, enhancing the performance of semantic segmentation tasks. On the other hand, unstructured datasets exhibit significant variations in geometry, terrain, and appearance, presenting challenges such as navigational ambiguities in rough terrain and tall grass [11–15].

The creation of large pixelwise semantic labels poses substantial challenges, necessitating human intervention to ensure accuracy and quality. Despite meticulous annotation efforts, generating pixel-accurate annotations often yields smaller datasets, particularly notable in high-quality semantic segmentation datasets. Beyond annotation challenges, navigating unstructured off-road environments presents additional hurdles, including resource scarcity and navigational complexities.

This paper explores the utilization of commercial environments, available publicly in the Unreal Store, built using Unreal Engine to generate pixel-accurate ground truth data for training semantic segmentation models [16]. Leveraging the abundance of environments available within Unreal Engine, ranging in scale, coupled with the Microsoft-owned open-source tool AirSim, allows for the collection of unprocessed semantic segmentation images with persistent random labels across different instances. The ForestSim dataset, introduced herein, aims

to enhance the accuracy of semantic segmentation models across various terrain types, leveraging diverse environments representative of different seasons. The dataset comprises 2094 RGB images with corresponding pixel-wise ground truth annotations, extracted from 25 different high-quality, realistic environments. An example RGB image and ground truth pixel-level semantic segmentation image produced from the data collected from Unreal Engine are illustrated in Fig. 1. Benchmarks, employing up-to-date methodologies such as Mean Intersection-Over-Union and Pixel Accuracy metrics [12,15], validate the efficacy of the proposed dataset.

After over a year of work and development, we introduce this dataset and processing pipeline with demonstrated positive results, with the goal of providing a high quality dataset to augment the capabilities of existing machine learning models.After collecting over 10,000 images we carefully selected 2094 higher quality images that maintained a diversity in the number of objects within an image. By leveraging this dataset to complement other existing simulated and real world datasets, we envision autonomous field robots excelling in a myriad of tasks, including timber sorting, harvesting operations, agricultural field tasks, and surveillance in challenging, unstructured environments.

2 Segmentation Datasets

Semantic segmentation datasets serve as foundational resources for partitioning images into meaningful parts through pixel-wise annotation. This segmentation task is fundamental for various applications in computer vision, facilitating tasks such as object detection, scene understanding, and autonomous navigation [17]. These pipelines include utilizing an encoder to create hierarchical representations of an image using a backbone such as ResNet with a decoder that upsamples features through convolutional layers, converting low dimensional features to original resolutions, which are the feature maps that are eventually used for pixel wise prediction.

2.1 Structured Datasets

Structured semantic segmentation datasets represent environments with well-defined boundaries and organized elements. This category encompasses a plethora of datasets, including COCO-Stuff [6], Pascal VOC [8], ADE20K [9], Pascal Context [10], Audi Structured [18], Cityscapes [7], KITTI [19], Mapillary [20], and ApolloScape [21]. Among numerous selections, for example, Mapillary provides a benchmark dataset specifically tailored for traffic sign classification [20], while KITTI focuses on common urban objects like buildings, trees, cars, and roads [19]. ApolloScape offers a diverse range of data captured from various cities and different times of the day, integrating camera videos, consumer-grade motion sensors, and 3D semantic maps [21]. These datasets are often collected by involving ground vehicles equipped with multiple sensors, capturing rich data from urban environments [7,19,21].

2.2 Unstructured Datasets

In contrast, unstructured semantic segmentation datasets capture environments with complex and varied characteristics, including rugged terrain, dense vegetation, and irregular structures. The RUGD dataset [12] serves as a benchmark for unstructured environments near creeks, vegetation, water bodies, trails, and villages. TAS500 [13] focuses on discerning traversable regions from non-traversable ones, categorizing 44 different objects into nine groups. The Rellis dataset [15] comprises synchronized sensor data collected using a mobile robotic platform, featuring diverse terrains like runways, aprons, and lakes. These datasets play a crucial role in advancing the robustness and adaptability of semantic segmentation models in challenging real-world scenarios.

However, compared to the structured datasets, the number of unstructured datasets is significantly lower. This scarcity stresses the importance of creating more datasets in this category to provide a more comprehensive representation of diverse and challenging environments. In our work, we provide a very accessible and reusable process to improve upon this scarcity through the use of simulated environments.

3 Relevant Uses in Autonomy

Understanding the characteristics of an environment is instrumental in various autonomous applications, particularly in supporting robot path estimation and navigation tasks. Leveraging both 3D terrain information and visual features collectively yields superior results compared to relying on either resource alone [22]. Models can be devised to generate color images and assign traversability costs to different regions based on their geometric attributes and visual appearance, contributing to more informed decision-making processes [23]. Texture-based features derived from onboard sensors such as inertial measuring unit (IMU), motor current, and bumper switches aid in binary segmentation of terrain traversability, enhancing the vehicle's ability to navigate challenging terrains [24]. Additionally, learning approaches that utilize models trained on data collected at different time points can improve nearsightedness by referencing past trajectory data [24].

Relevant to usage, domain adaptation emerges as a critical technique for addressing disparities between source and target domains within semantic segmentation datasets. Unsupervised Domain Adaptation (UDA) methods alleviate the arduous and time-consuming process of manually labeling target environments [25]. These models are trained utilizing labeled source data alongside unlabeled target data, with the objective of minimizing the domain gap between the two domains to achieve optimal performance. Methodologies for UDA include aligning latent representations in feature space to harmonize domains [26, 27], reducing visual disparities between domains through input-level adaptation [28, 29], transferring images between domains to refine segmentation models [30], and incorporating discriminator layers to refine predictions from both domains [31, 32]. By simplifying data preparation for training, domain adaptation techniques facilitate the seamless integration of diverse datasets into

segmentation models [33], serving as a powerful tool for improving the adaptability, robustness, and safety of autonomous vehicles in complex and dynamic environments.

4 ForestSim Data Collection and Preparation

Synthetically generated data has emerged as a powerful tool for enhancing the performance of deep neural networks in image segmentation tasks. Sim2Real, simulation to reality, focuses on developing algorithms that generalize from a virtual setting to a real world setting using techniques such as domain randomization, transfer learning by pre-training models in simulation and then tuning with real world data, and using a combination of simulated and real-world data providing rich and diverse data to train on [34]. Using the aforementioned techniques, benchmark evaluations conducted using synthetic datasets have demonstrated comparable accuracy to real-world data in image segmentation tasks. Moreover, with the application of domain adaptation techniques using labeled real data with synthetic data, results can not only mimic, but can outperform those of just real-world datasets, thereby broadening the scope of dataset applications [35].

4.1 ForestSim Environments

ForestSim aims to provide a level of diversity not generally common in existing datasets. The proposed ForestSim dataset includes a diverse array of environments, ranging from mountains, forests, and hills to jungles and marshes. As depicted in Fig. 2, example images showcase the varied terrain types present in the dataset. This diversity extends to the geometry and size of objects within the environments, facilitating the development of more adaptable models compared to datasets with limited diversity. Notably, the environments utilized in our ForestSim are meticulously crafted from commercial purposes, with high-fidelity realism in appearance, proportions, lighting, textures, and object placement. The Unreal Engine provides photorealistic environments with various illuminations and changing light conditions [16]. This fidelity enhances the dataset's utility, particularly when combined with existing techniques such as synthetic image generation and Generative Adversarial Networks (GANs) for domain transfer between synthetic and real-world datasets, thereby augmenting the capabilities of segmentation models. [36]. Our aim in this dataset was to provide high quality, diverse images within an environment where each and every image was examined to avoid over representation of similar images. Further more we ensure diversity in shape and size of classes between environments as these images were collected from 25 different environments. The processing was meticulous for each environment ensuring the high quality of the ground truth images.

Fig. 2: Example RGB images of seasonal environments. These pictures demonstrate the unstructured, off-road, and forested characteristics of unstructured environments.

4.2 Hardware and Software

Data collection involves a combination of manual intervention and automation, presenting unique challenges. Similar to the TartanAir dataset [37], our approach integrated various modalities, including RGB images and segmentation data.

The data collection system operates on hardware comprising an Intel NUC NUC11PHKi7 11th Gen Core i7-1165G7 Quad-Core processor, 32GB DDR4 RAM, 1TB PCIe NVMe SSD, and GeForce RTX 2060 6GB GDDR6 Graphics, running the Windows 11 OS. Both Unreal Engine and AirSim [38] offer robust support for Windows and macOS environments. Notably, hardware limitations

can impact performance, emphasizing the importance of optimizing system configurations.

The Epic Games Launcher is leveraged to install Unreal Engine and access environments, while simulation within Unreal Engine is facilitated by AirSim, a powerful plugin. AirSim enables interaction with ground or air vehicles programmatically, offering functionalities such as image retrieval, state querying, and vehicle control. Interactions with the AirSim API are orchestrated using Python 3.7, ensuring seamless integration and flexibility in data collection processes.

Fig. 3: More dense environments, an example is seen on the left, required manual control. On the right, data was able to be collected programmatically with no manual control.

4.3 Data Acquisition

Data collection attempts to rely on automated procedures driven by Python scripts, but almost all attempts required manual interventions. At five-second intervals, RGB and segmentation images are captured by a simulated ground vehicle outfitted with three cameras—front left, front center, and front right—using AirSim.

The vehicle follows predefined paths, synchronized with time intervals, optimizing efficiency in its given operating environments. However, navigation poses challenges in congested areas where small, impassable objects increase collision risks, occasionally necessitating manual control to resolve navigation issues. Fig. 3 illustrates a scenario of such challenges encountered during data acquisition. This was a major challenge and limited automated data collection, requiring constant supervision.

4.4 Data Processing and Statistics

Data processing primarily centers on segmentation images collected via AirSim, aiming at developing and annotating pixel-wise ground truth labels. AirSim

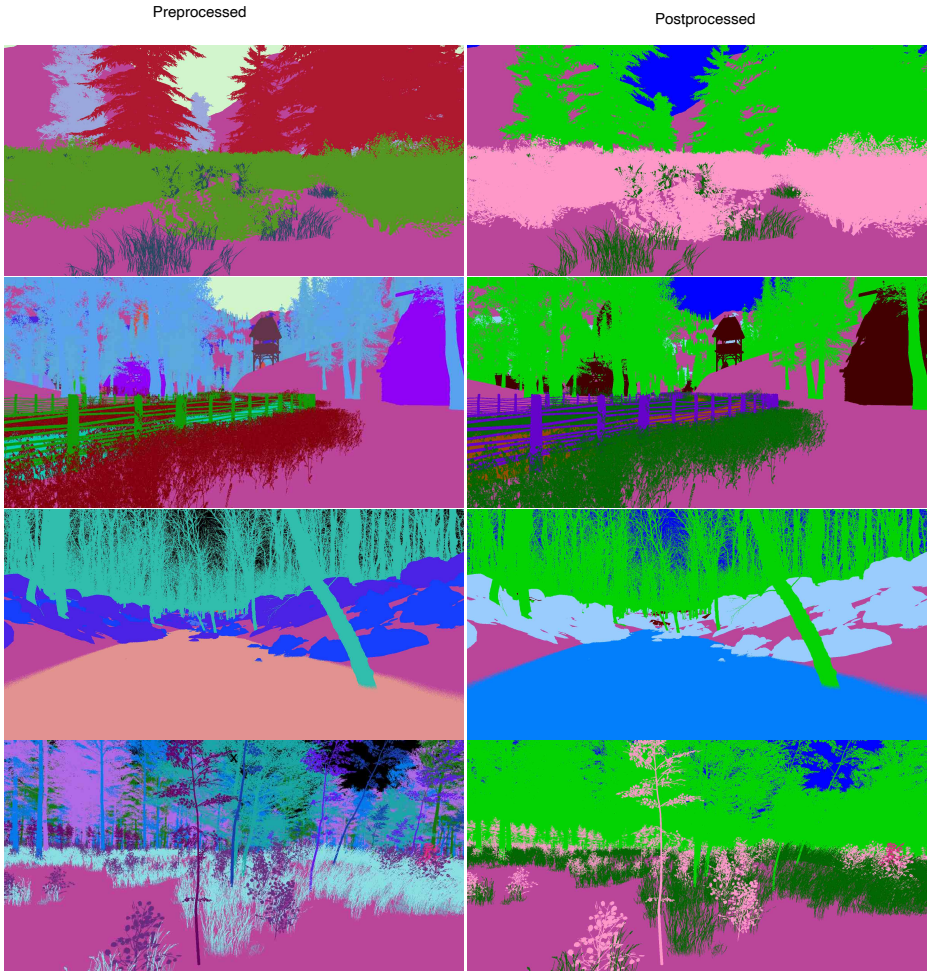

Fig. 4: On the left are examples of segmentation images captured using AirSim and on the right are processed ground truth images. This highlights the nature and challenge of AirSim, showing the original images collected from AirSim, which randomly assigns different RGB values to same class of object. Each image in our dataset had to be carefully examined to ensure that the RGB value was accounted for. Though AirSim does provide the base labeled images, without the processing pipeline to create the ground truth labels, it provides little value in the context of supervised learning.

assigns a unique ID to each static mesh, mapping it to an RGB value from a predefined palette of 255 RGB values. However, many inconsistencies arose in object labeling and color assignments across different environments. To address this, each environment underwent manual curation, where each image was

examined to establish mappings to correctly label similar classes or the same class of object to a predesignated RGB assignment. The segmentation images collected from AirSim provide little value without the extensive processing to create ground truth labels. The work here, in addition to a high quality dataset that is ready to be used, provides a process to, from low value images provided by AirSim, create a consolidated set of useful unstructured ground truth labels.

The semantic labels for our dataset are established through meticulous mapping and reconciliation, eliminating redundancy and harmonizing object representations across diverse environments. Fig. 4 showcases examples of original and converted segmentation images, highlighting the efficacy of our data processing pipeline. Our pipeline required mapping for each unique environment. The time-intensive process was creating the mapping, which demanded manually examining all of the uniquely colored pixels within the collected segmentation images. After mapping was complete, the consolidation process of relabeling the individual pixels was automated.

5 Annotation Statistics and Ontology

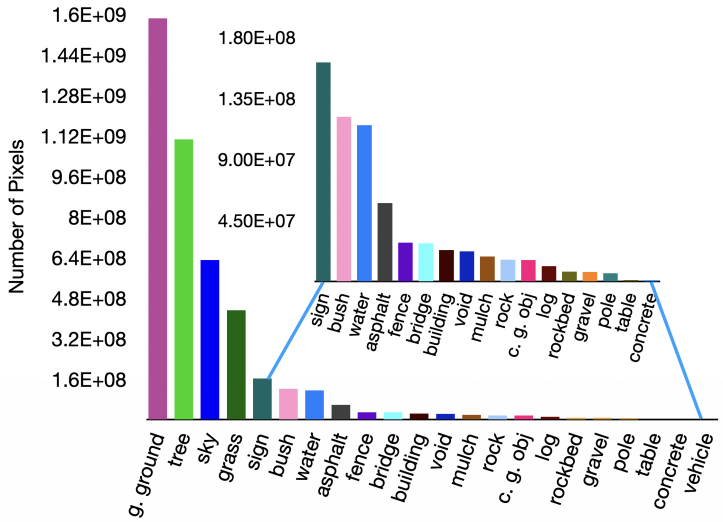

Fig. 5: Numbers of total pixels per class in the dataset in descending order.

Our ForestSim dataset consists of a diverse array of common classes generally found in almost all unstructured environments, including grass, trees, poles, water bodies, sky, vehicles, containers, asphalt, gravel, mulch, rock beds, logs, bushes, signs, rocks, bridges, concrete structures, buildings, void regions, and generic ground. Fig. 5 provides an overview of the distribution of pixels across these classes within the dataset. Fig. 6 presents the finalized mappings of object classes to RGB values, with trees serving as an illustrative example.

Each of the 20 distinct object classes is assigned a unique RGB value for identification and labeling. The category of "generic ground" comprises all traversable ground surfaces. In AirSim, flat ground in certain environments was labeled a specific RGB value, most likely because no static mesh was used for it during development. These are traversable, flat regions. Moreover, the category of "generic container objects" includes a variety of miscellaneous objects that may pose collision risks or influence navigation. These include benches, trash cans, playground equipment (such as slides and swings), water containers, log containers, and similar items.

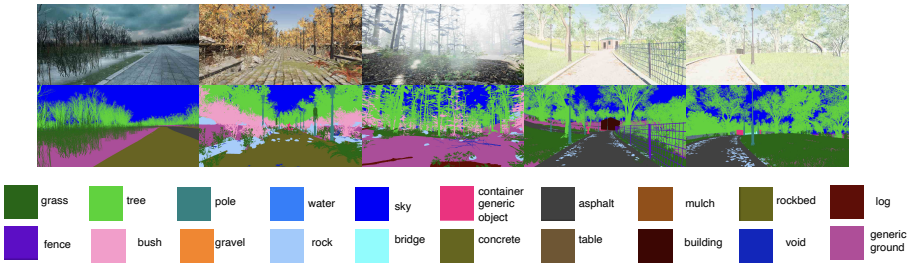

Fig. 6: Examples of ground truth annotations from the ForestSim dataset. The first row is the photo-realistic RGB image collected from the environment, and the second row is the corresponding semantic segmentation ground truth images used to train the models to evaluate the images. The examples demonstrate the wide array of environments the images were collected from, ranging from a flat winter environment to a forested summer environment full of elevations. Please note that these are the pixel-wise, true semantic segmentation images after consolidation and labeling.

6 Benchmarks for Domain Adaptive Segmentation

6.1 Baselines and Experimental Setups

The benchmarking process for ForestSim leverages a unified framework implemented using MMSegmentation [39]. All of the combinations of models used here were built referencing MMSegmenation and are primarily used to support and strengthen the case of our high quality dataset. Our goal was not to provide another model but to use existing models to show the quality of our data. Models are structured following an encoder-decoder pattern, with various configurations explored to optimize segmentation performance.

One approach utilizes a pretrained ResNet50v1c model as the encoder, coupled with a Pyramid Scene Parsing Network (PSPNet) decoder to aggregate and consider multi-scale context, employing Cross Entropy Loss with a weight of 1.0. Additionally, other models combine pretrained ResNet50v1c and ResNet101v1c

models with different decoders, including Atrous Spatial Pyramid Pooling (ASPP) and Depthwise Separable Convolutions, each utilizing Cross Entropy Loss with specific weight configurations. ASPP captures data on multiple scales of time and space with dilated convolutions with v1c representing a specific dilation of 32-32-64 at the beginning channel [40].

Furthermore, two models integrate MixVisionTransformer (mit) which utilizes scalable transformed based architectures and Segformer decoders with Cross Entropy Loss, based on pretrained mit-b0 and mit-b5 models, respectively [39]. Another set of models employ ResNet encoders paired with Mask2Former decoders which uses a transformer based approach, augmented with MSDeformAttnPixel and trained with various loss functions and optimizer settings [39].

Additionally, models incorporating SwinTransformer decoders, built from combinations of pretrained Swin Tiny, Swin Small, Swin Base, and Swin Large models, are utilized. These models are trained using AdamW optimizer and PolyLR scheduler, providing learning rates and regularization [39].

6.2 Data Split, Training, and Evaluation Metrics

The data was split using a stratified train/test split. 90% of the 2094 labeled images were used for training and 10% were used for testing. Training of models occurred on 4 nodes, each containing SUSE Enterprise Linux Server (SLES) version 15 with 256 GB of memory and two 64-core, 2.25 GHz, 225-watt AMD EPYC 7742 processors running 4 tasks per node and 4 NVIDIA A100 GPUs per node. The MMSegmentation uses an IterBasedRunner and updates parameters per batch [39]. The number of iterations for training varies based on the scheduler that was used when configuring the models, but it ranged from 40,000 to 160,000 iterations.

Metrics to measure performance include standard semantic segmentation metrics such as Mean IoU and pixel wise segmentation. Mean IoU is the average IoU between all classes [41]. The IoU for each class is computed as $\frac{TP}{TP+FP+FN}$. Mean pixel wise segmentation accuracy is also used, which is the average segmentation accuracy per model and is a preferred metric as it evenly weights each class.

6.3 Analysis and Experimental Evaluation

The trained models were taken from the last checkpoint and were employed to make predictions on the randomized test set, and their performances were evaluated and summarized in Table 1. Overfitting was not considered in this context as the primary goal was to benchmark our dataset. This provides benchmarks for future comparison due to the highly relevance to our data. The trained models were able to predict within the dataset, classes with high accuracy and a high mean IoU. Our mean IoU is high due to the over representation of our most common classes within our test set. The results are positive, in that the ground truth images, used to train the models, have demonstrated reasonably high mean IoU and pixel accuracy. The significant effort dedicated to preparing

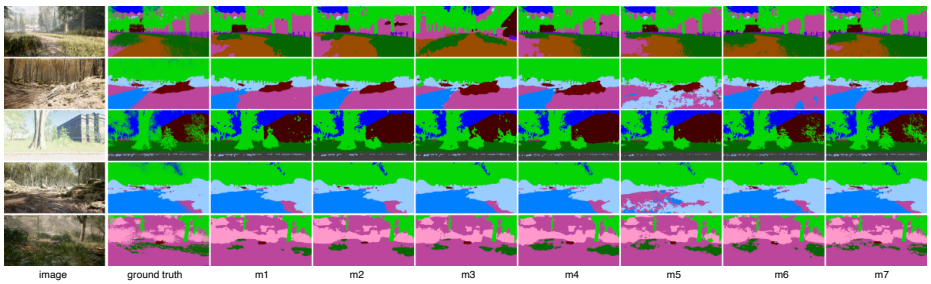

Fig. 7: The original image, the ground truth, and the predicted image annotation for models 1 to 7. Comparing the boundaries of the objects between the ground truth and predictions demonstrates the difficulty in predicting object boundaries.

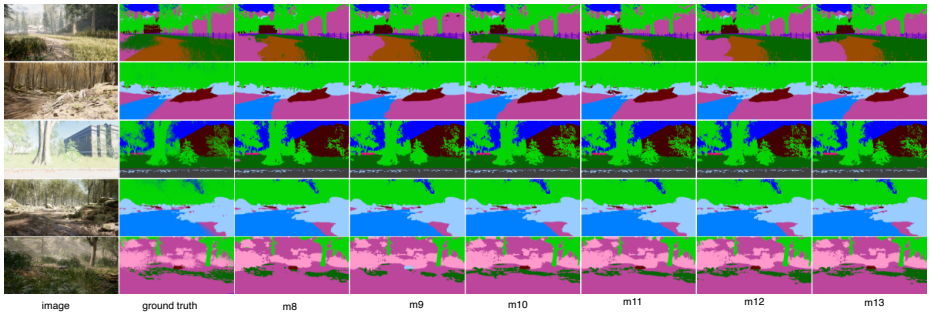

Fig. 8: The original image, the ground truth, and the predicted image annotation for models 8 to 13.

accurate ground truth labels, with which the models were trained, is returning consistent results, demonstrating the validity of the process. Rigorous review and refinement processes were implemented, ensuring the removal of low-quality data and enhancing the dataset's integrity as a high-quality baseline for training on an unstructured simulated environment.

High scores observed in the Pixel Accuracy column underscore the models' proficiency in learning and accurately predicting highly represented objects such as trees, sky, and generic ground (traversable land). Visual representations of prediction results from the trained models are illustrated in Fig. 7 and 8.

Table 1 summarizes all of the model results on the test set which was a random 10% of our data. All of these models were trained and tested on the same data. The table breaks down the methods that are built using a pretrained model, an encoder, and a decoder. Our results show the IoU ranging from a low of 59.16% to 74.02%. The IoU value ranges demonstrate that the model is performing relatively well and is predicting the majority of the class correctly. The unclear edges leading to difficulty predicting at boundaries of objects are

Model	Method	M. IoU↓	Pix. Acc.↓	M. Pix. Acc. ↓
m1	resnet50v1c + ResNetV1c + PSPHead	61.64	89.85	72.14
m2	resnet50v1c + ResNetV1c + ASPPHead	61.87	89.91	72.81
m3	resnet101v1c + ResNetV1c + ASPPHead	62.81	89.86	73.13
m4	resnet50v1c + ResNetV1c + DepthwiseSeparableASPPHead	59.16	89.31	72.93
m5	resnet101-v1c + ResNetV1c + DepthwiseSeparableASPPHead	59.22	88.32	69.56
m6	mit-b0 + MixVisionTransformer + SegformerHead	61.82	90.52	71.12
m7	mit-b5 + MixVisionTransformer + SegformerHead	67.93	92.05	76.42
m8	resnet50 + ResNet + Mask2FormerHead	67.48	91.34	75.77
m9	resnet101 + ResNet + Mask2FormerHead	65.80	91.29	74.61
m10	swin-base + SwinTransformer + Mask2FormerHead	74.50	92.57	82.30
m11	swin-large + SwinTransformer + Mask2FormerHead	75.31	92.65	82.68
m12	swin-tiny + SwinTransformer + Mask2FormerHead	70.46	92.14	79.79
m13	swin-small + SwinTransformer + Mask2FormerHead	74.02	92.39	81.39

Table 1: Prediction results from architectures used beginning with the model number, the pretrained model, encoder, and decoder.

also negatively impacting this result. This is one of the existing challenges of unstructured environments and can be seen in Fig. 7 and Fig. 8. Pixel accuracy, which is the total correct predicted divided by the total number of pixels, ranges from a low of 88.32% to 92.65%. We conclude that objects that are represented more in the data are being predicted with high accuracy. The mean pixel accuracy, which is the average prediction accuracy between all of the classes, was negatively impacted by objects that are not represented well within the dataset, such as vehicle, which is an uncommon class in an unstructured environment. The mean pixel accuracy for concrete and table was the next two lowest after vehicle. That correlates with our conclusion that this is due to data sparsity, as these were the three least represented classes. Pole was able to be predicted well, but most likely due to how uniquely it is shaped.

6.4 Discussion and Future Work

Despite the comprehensive coverage of classes, data sparsity is observed in certain categories, such as vehicles, concrete structures, poles, gravel, and rock beds, constituting a minimal percentage of the dataset. This sparsity presents challenges in accurately classifying these objects, potentially leading to erroneous decisions during segmentation tasks. Additionally, dynamic scenarios such as, trees falling or strong wind, are not readily simulated within AirSim, limiting the availability of data capturing such situations. However, existing methodologies can address these limitations, offering avenues for enhancing dataset diversity and mitigating segmentation challenges in ForestSim.

We believe the findings presented are promising. Our future work will be further improvement through data balancing and enrichment of sparse classes. For instance, augmenting ForestSim with complementary datasets shows promise in enhancing the adaptability of semantic segmentation models. Integration with diverse simulation environments or existing datasets can address challenges such as dynamic behavior and data sparsity. Moreover, leveraging synthetic image

production and GAN networks for domain transfer between synthetic and real-world datasets holds considerable potential for gaining valuable insights and making significant improvements.

7 Conclusion

To enhance the adaptability of semantic segmentation models, especially in off-road environments, we introduce ForestSim. This new dataset is designed for unstructured environments, featuring realistic off-road, forested, and mountainous terrains across various seasons. ForestSim includes 20 classes, offering comprehensive coverage. This dataset comprises 2094 ground truth pixel-wise annotations, providing a valuable resource with high accuracy for semantic segmentation tasks.

References

1. H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision (IJCV)*, 2018.
2. H. Li, J. Cai, T. N. A. Nguyen, and J. Zheng, "A benchmark for semantic image segmentation," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.
3. A. Athar, J. Luiten, P. Voigtlaender, T. Khurana, A. Dave, B. Leibe, and D. Ramanan, "Burst: A benchmark for unifying object recognition, segmentation and tracking in video," 2022.
4. H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2403–2412.
5. D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," 02 2019.
6. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.
7. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," 2016.
8. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and Zisserman, "The pascal visual object classes (voc) challenge," 2010.
9. B. Zhou, H. Zhao, X. P. abd Sanja Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," 2017.
10. R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

11. M. Kragh and J. Underwood, "Multimodal obstacle detection in unstructured environments with conditional random fields," *Journal of Field Robotics*, vol. 37, no. 1, p. 53–72, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1002/rob.21866
12. M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.
13. K. A. Metzger, P. Mortimer, and H.-J. Wuensche, "A fine-grained dataset and its efficient semantic segmentation for unstructured driving scenarios," 2021.
14. B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Eff-unet: A novel architecture for semantic segmentation in unstructured environment," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1473–1481.
15. P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," 2022.
16. Epic Games, "Unreal engine 4." [Online]. Available: https://www.unrealengine.com
17. D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, p. 1341–1360, Mar. 2021. [Online]. Available: http://dx.doi.org/10.1109/TITS.2020.2972974
18. J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2d2: Audi autonomous driving dataset," 2020.
19. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.
20. C. Ertler, J. Mislej, T. Ollmann, L. Porzi, G. Neuhold, and Y. Kuang, "The mapillary traffic sign dataset for detection and classification on a global scale," 2020.
21. X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, p. 2702–2719, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2926463
22. H. Roncancio, M. Becker, A. Broggi, and S. Cattani, "Traversability analysis using terrain mapping and online-trained terrain type classifier," 06 2014, pp. 1239–1244.
23. M. Shneier, T. Chang, T. Hong, W. Shackleford, R. Bostelman, and J. Albus, "Learning traversability models for autonomous mobile vehicles," *Auton. Robots*, vol. 24, pp. 69–86, 11 2008.
24. D. Kim, J. Sun, S. Oh, J. Rehg, and A. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," 02 2006, pp. 518 – 525.
25. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," 2017.
26. L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

27. Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," 2018.
28. J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," 2017.
29. Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," 2018.
30. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2020.
31. Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," 2019.
32. Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," 2020.
33. S. Lee, J. Hyun, H. Seong, and E. Kim, "Unsupervised domain adaptation for semantic segmentation by content transfer," 2020.
34. V. Prabhu, D. Acuna, A. Liao, R. Mahmood, M. T. Law, J. Hoffman, S. Fidler, and J. Lucas, "Bridging the sim2real gap with care: Supervised detection adaptation with conditional alignment and reweighting," 2023. [Online]. Available: https://arxiv.org/abs/2302.04832
35. A. Shafaei, J. J. Little, and M. Schmidt, "Play and learn: Using video games to train computer vision models," 2016.
36. W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," 2016.
37. W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," 2020.
38. S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065
39. M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.
40. T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1812.01187
41. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.