ForestSim: A Simulated Dataset for Forest Scene Understanding

Pragat Wagle, Zheng Chen, Lantao Liu

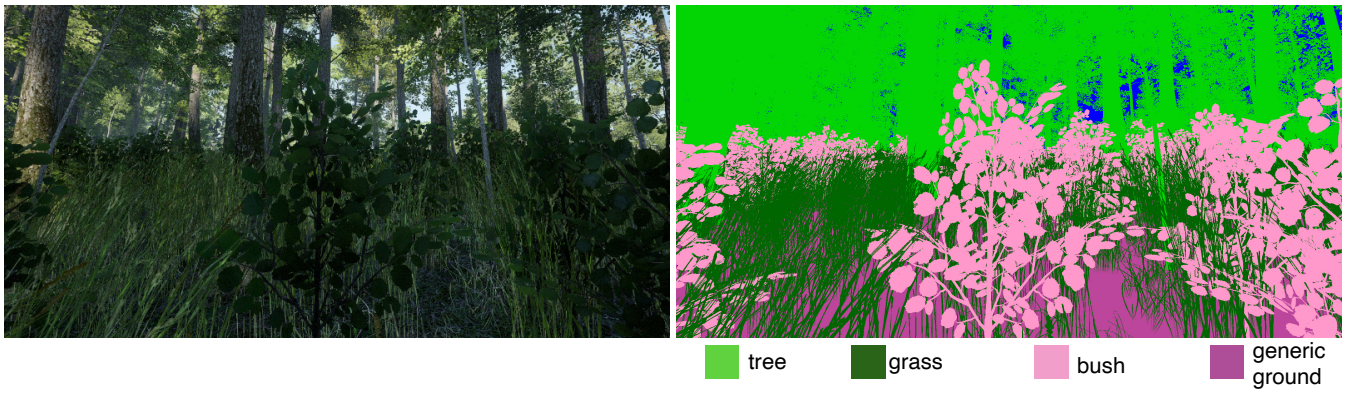

Fig. 1. An example image in **ForestSim** dataset. The size, shapes, and ratios are similar to those in the real world and demonstrate characteristics of an unstructured environment where the edges of objects are challenging to discern and almost blend into one another.

***Abstract*— Semantic segmentation, a critical process in computer vision, involves the intricate task of classifying each pixel in an image into specific categories. This process is essential for machines to comprehend the visual world with precision. However, the creation of large-scale datasets with pixel-level labels, necessary for training semantic segmentation models, is a challenging and labor-intensive endeavor. It requires 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. The lack of data for such scenarios 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. The resulting dataset, named ForestSim, comprises 2094 finely annotated images obtained from 25 diverse environments using a ground vehicle. We evaluate ForestSim using state-of-the-art benchmark techniques and make our dataset and associated code publicly available for further research and development.**
Dataset: `https://vailforestsim.github.io`
Code: `https://github.com/pragatwagle/ForestSim`

I. 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 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 taxonomy of large datasets varies depending on their intrinsic characteristics. One popular taxonomy scheme categorizes image datasets into structured and unstructured datasets. Structured datasets, reminiscent of urban environments with clear boundaries, vehicular traffic, and regular-shaped buildings [6]–[10], have been crucial in constructing high-performing semantic segmentation models. Furthermore, datasets are employed to augment existing models, enhancing the accuracy 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. For instance, the CamVid dataset required 60 minutes per image for high-quality labeling [16], while the Cityscapes dataset demanded 90 minutes per image [7]. 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 built using Unreal Engine to generate pixel-accurate ground truth data for training semantic segmentation models.

All authors are with Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408, USA. Email: {pwagle, zc11, lantao}@iu.edu.

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. Due to the simulated nature, data collection was expedited, with annotation taking at most a few hours for the most diverse environment, while some environments required only 30 minutes for 100 images. The pixel RGB assignment propagated through environments, decreasing labeling time substantially compared to CamVid and Cityscapes. 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.

We introduce this dataset with the goal of enhancing the capabilities of existing machine learning models. By leveraging this dataset, 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.

II. 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 samples through convolutional layers, converting low dimensional features to original resolutions, which are the feature maps that are eventually used for pixel wise prediction.

A. 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].

B. 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.

III. 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 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 near-sightedness 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

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

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.

IV. 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. Benchmark evaluations conducted on synthetic datasets have demonstrated comparable accuracy to real-world data in image segmentation tasks. Moreover, with the application of domain adaptation techniques, synthetic data can not only mimic but also outperform real-world datasets, thereby broadening the scope of dataset applications [34].

A. ForestSim Environments

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

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.

diversity. Notably, the environments utilized in our ForestSim are meticulously crafted for 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. 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 [35].

B. Hardware and Software

Data collection involves a combination of manual intervention and automation, presenting unique challenges. Similar to the TartanAir dataset [36], 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 [37] 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.

C. Data Acquisition

Data collection predominantly relies on automated procedures driven by Python scripts, with occasional manual interventions. At five-second intervals, RGB and segmentation images are captured by a 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

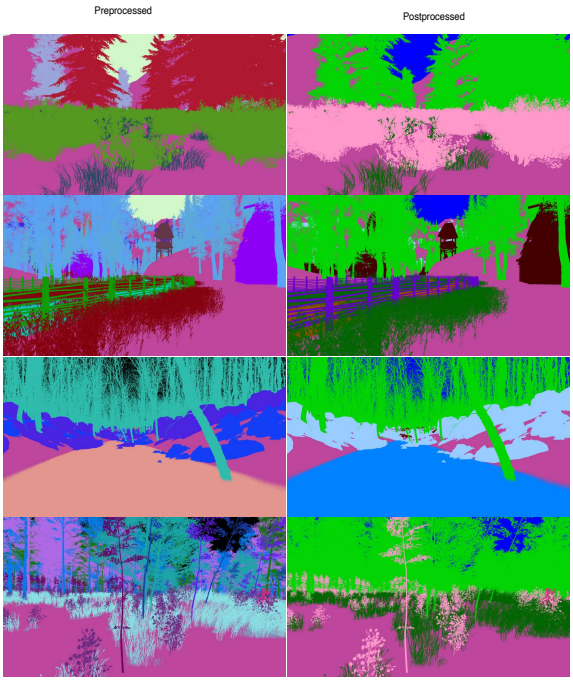

Fig. 4. Examples of segmentation images captured directly from AirSim are on the left. These images were processed by manually determining the object each RGB value corresponds with and using this mapping to generate the ground truth pixelwise labels on the right.

collision risks, occasionally necessitating manual control to resolve navigation issues. Fig. 3 illustrates a scenario of such challenges encountered during data acquisition.

D. 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 assigns a unique ID to each static mesh, mapping it to an RGB value from a predefined palette of 255 RGB values. However, inconsistencies arose in object labeling and color assignments across different environments. To address this, each environment underwent manual curation, establishing mappings to reconcile variations in object labeling and RGB assignments. For instance, disparate RGB values assigned to the same object class are consolidated, ensuring uniformity across environments. Fig. 6 presents the finalized mappings of object classes to RGB values, with trees serving as an illustrative example.

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 required 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.

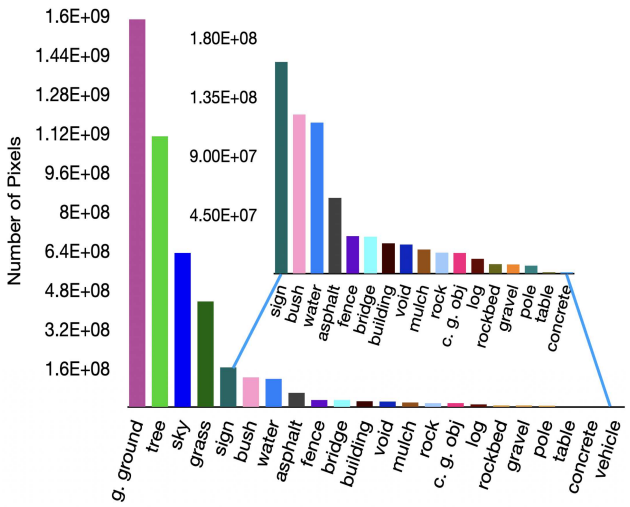

Fig. 5. Numbers of total pixels per class in the dataset in descending order.

V. Annotation Statistics and Ontology

Our ForestSim dataset consists of a diverse array of classes essential for semantic segmentation, 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.

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.

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 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.

VI. Benchmarks for Domain Adaptive Segmentation

A. Baselines and Experimental Setups

The benchmarking process for ForestSim leverages a unified framework implemented using mmsegmentation [38]. Models are structured following an encoder-decoder pattern,

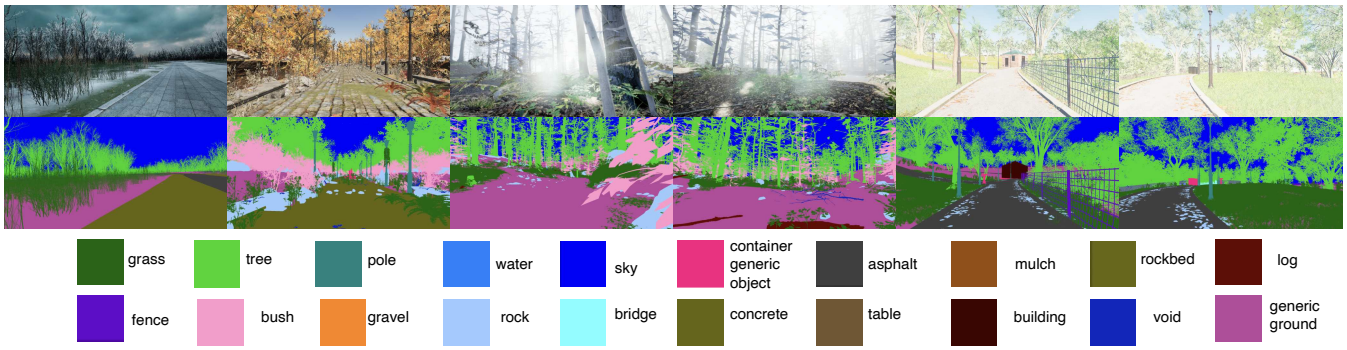

Fig. 6. Examples of ground truth annotations from the ForestSim dataset. The first row is the photorealistic RGB image collected from the environment, and the second row is the corresponding semantic segmentation. Please note that these are the pixel-wise, true semantic segmentation images after consolidation and labeling.

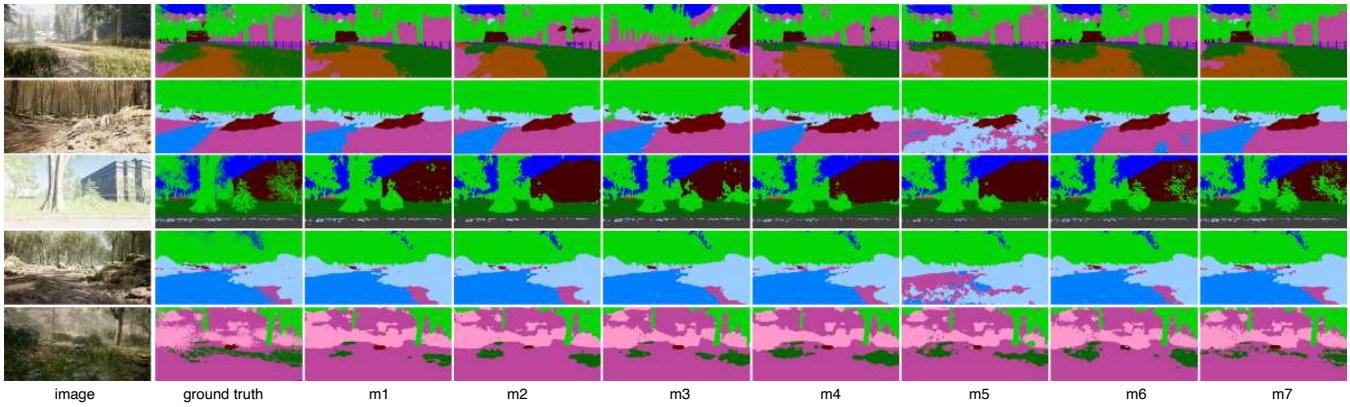

Fig. 7. The original image, the ground truth, and the predicted image annotation for models 1 to 7.

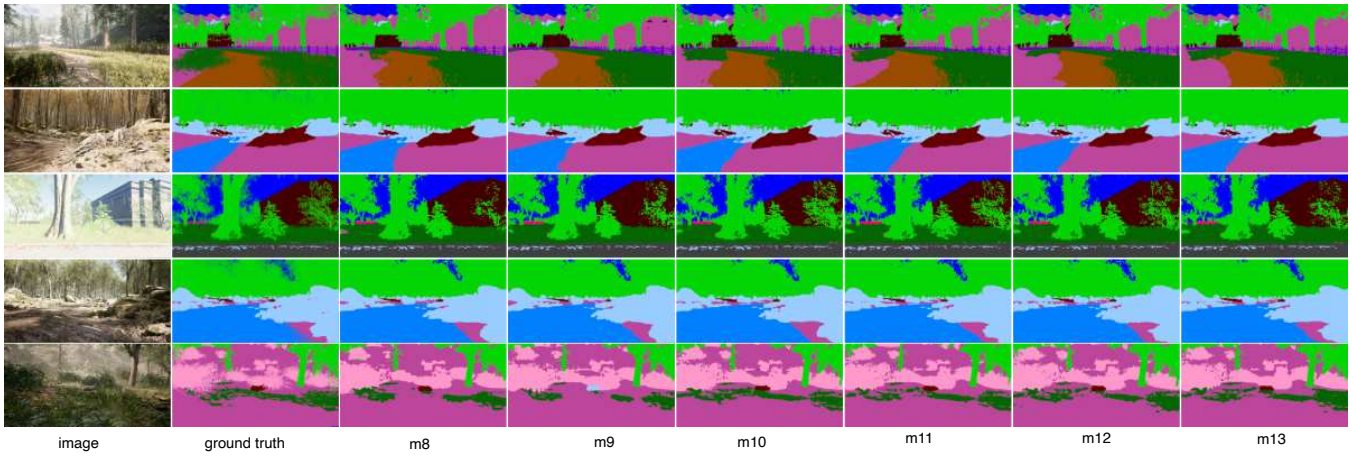

Fig. 8. The original image, the ground truth, and the predicted image annotation for models 8 to 13.

with various configurations explored to optimize segmentation performance.

One approach utilizes a pretrained ResNet50v1c model as the encoder, coupled with a PSPNet decoder, 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 DepthwiseSeparable, each utilizing Cross Entropy Loss with specific weight configurations. Furthermore, two models integrate MixVisionTransformer and Segformer decoders with Cross Entropy Loss, based on pretrained mit-b0 and mit-b5 models, respectively. Another set of models employ ResNet encoders paired with Mask2Former decoders, augmented with MSDeformAttnPixel and trained with various loss functions and optimizer settings. 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.

TABLE I

PREDICTION RESULTS FROM ARCHITECTURES USED BEGINNING WITH THE MODEL NUMBER, THE PRETRAINED MODEL, ENCODER, AND DECODER.

Model	Method	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

B. Data Split, Training, and Evaluation Metrics

The data was split randomly using a train/test split so that 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 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 [39]. 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.

C. Analysis and Experimental Evaluation

The trained models were employed to make predictions on the randomized test set, and their performances were evaluated and summarized in Table I. Notably, the ForestSim dataset stands out for its exceptional quality, with significant effort dedicated to preparing accurate ground truth labels. 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 I 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 and boundaries of objects are also negatively impacting this result, which is one of the existing challenges of unstructured environments. 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 of all of the classes, was negatively impacted due to a 0 percent accuracy for vehicle. 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.

D. Discussion and Future Work

The findings 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.

VII. 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] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.

[2] Hui Li, Jianfei Cai, Thi Nhat Anh Nguyen, and Jianmin Zheng. A benchmark for semantic image segmentation. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2013.

[3] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video, 2022.

[4] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2403–2412, 2019.

[5] Di Feng, Christian Haase-Schuetz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, W. Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges, 02 2019.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016.

[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Zisserman. The pascal visual object classes (voc) challenge, 2010.

[9] Bolei Zhou, Hang Zhao, Xavier Puig abd Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset, 2017.

[10] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan 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] Mikkel Kragh and James Underwood. Multimodal obstacle detection in unstructured environments with conditional random fields. *Journal of Field Robotics*, 37(1):53–72, March 2019.

[12] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung 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] Kai A. Metzger, Peter Mortimer, and Hans-Joachim Wuensche. A fine-grained dataset and its efficient semantic segmentation for unstructured driving scenarios, 2021.

[14] Bhakti Baheti, Shubham Innani, Suhas Gajre, and Sanjay 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)*, pages 1473–1481, 2020.

[15] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis, 2022.

[16] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. Video-based Object and Event Analysis.

[17] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, March 2021.

[18] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset, 2020.

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[20] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale, 2020.

[21] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, October 2020.

[22] Henry Roncancio, Marcelo Becker, Alberto Broggi, and Stefano Cattani. Traversability analysis using terrain mapping and online-trained terrain type classifier. pages 1239–1244, 06 2014.

[23] Michael Shneier, Tommy Chang, Tsai Hong, William Shackleford, Roger Bostelman, and James Albus. Learning traversability models for autonomous mobile vehicles. *Auton. Robots*, 24:69–86, 11 2008.

[24] Dongshin Kim, Jie Sun, Sang Oh, James Rehg, and Aaron Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. pages 518 – 525, 02 2006.

[25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[26] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin 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] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation, 2018.

[28] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation, 2017.

[29] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation, 2018.

[30] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[31] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation, 2019.

[32] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation, 2020.

[33] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. Unsupervised domain adaptation for semantic segmentation by content transfer, 2020.

[34] Alireza Shafaei, James J. Little, and Mark Schmidt. Play and learn: Using video games to train computer vision models, 2016.

[35] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine, 2016.

[36] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam, 2020.

[37] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[38] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.