

# ForestSim: A Simulated Dataset for Forest Scene Understanding

Pragat Wagle, Zheng Chen, Lantao Liu

**Abstract**— Autonomous driving presents many challenges including object detection and recognition, which are essential functions in providing autonomous agents the environmental context to make informed decisions. To provide autonomous agents with higher intelligence, many datasets have been collected to provide the necessary context for many of these algorithms. The majority of existing data sets focus on structured data from urban environments where objects in the images have clear boundaries and very regular shapes. There is a gap in data available for unstructured environments, where boundaries may be less clear and contain irregular objects. This unstructured data would be representative of non-urban outdoor forested mountainous off-road environments or disaster relief scenarios. Here we present ForestSim, a unstructured dataset collected in a simulated setting. The ForestSim dataset consists of 2094 annotated images collected using a ground vehicle from 25 different environments. The aim is to help lessen that gap and provide a means to more easily collect this data without utilizing as many resources. Here we aim to build a automated pipeline to collect data with which we are able to obtain multi modal data including segmentation, depth perspective, and rgb images of the environment. Images are processed, grouped, and then labeled to provide ground truth labels following the groundwork of the RUGD dataset and Tartan Air dataset. Benchmarks evaluations are performed with highly reputable techniques currently considered state-of-art. Dataset link: <https://vailforestsim.github.io>. Repository link: <https://github.iu.edu/pwagle/ForestSim>

## I. INTRODUCTION

Existing structured datasets exhibit characteristics seen in an urban environment where boundaries are clear, vehicles are driving, pedestrians are walking on built pavements, clear boundaries outline the road, and buildings are shaped regularly. [1], [2], [3], [4], and [5]. There is an abundance of datasets of structured environments but there exists a gap in available data for unstructured outdoor environments. [6], [7], [8], and [9]

Existing datasets, of both types, have proven and have laid a foundation for improving object detection, classification, and semantic segmentation leading to the development of benchmark data sets. [10], [11], [12], and [13]. This is especially the case when the environments are represented well in the data. [14]

There have been efforts to close the gap, in quantity, that exists between structured and unstructured environments. [6], [7], [8], and [9] There are many challenges that are presented in collecting data from unstructured off-road environments including a lack of resources and challenging traversability which has led to the progression of utilizing a simulated

Thanks to Naval Surface Warfare Center, Crane Division, (NSWC Crane) Crane, IN and Vehicle Autonomy and Intelligence Lab, Bloomington, IN

approach to gather data to push and benchmark existing slam algorithms. [15]



Fig. 1. An image collected from the Alder Tree Environment, a simulated forested environment during the Autumn season. The size, shapes, and ratios are similar to the real world and demonstrates characteristics of an unstructured environment where edges of objects are challenging to discern and almost blend into one another. Certain objects are taller such as the grass(dark green) and bushes(pink).

Unstructured Environments have large variations in geometry, appearance, and shapes where tall grass and rough terrain may be interpreted as non-navigable terrain. [16] These environments can include agricultural fields and orchards. Challenges in these environments include illumination occlusion and unconventional shape, size, and color. As we continue to improve upon the existing gap, the aim is to utilize the data to improve in activities such as sorting timber, harvesting operations, operations in agricultural fields and surveillance, and robots working along side humans in unstructured environments such as forklifts to safely work along side humans. [17] and [18]

The ForestSim data set will help to improve accuracy in mountainous regions and forested areas. Environments during various seasons were selected. The data consists of 2094 rgb images with 2094 corresponding pixel wise ground truth annotated labeled images. These were collected from 25 different high quality, realistic environments environments with varying seasons.

Benchmarks are performed using up to date, state of the art methodologies such Mean Intersection-Over-Union (Mean IoU) and Pixel Accuracy metrics. [6] and [9]

## II. SEGMENTATION DATASETS

Semantic Segmentation Datasets aim to partition an image into meaningful parts using pixel wise annotation. Semantic segmentation can be classified into one or two stage pipelines. [19].

### A. Structured Datasets

There are many more structured datasets than unstructured datasets, leading to a large gap. [1], [3], [4], [5], [20], [2], [21], [22], and [23]. Mapillary [22] provides a benchmark data set to classify traffic signs. KITTI [21] is a dataset that

consists of common objects found in an urban environment such as building, tree, sky, car, and road annotated manually by researchers. The ApolloScape [23] provides data from various cities and day times by integrating camera videos, consumer-grade motion sensors (GPS/IMU), and a 3D semantic map. These datasets generally involve a small ground vehicle mounted with multiple sensors that capture data. [2], [23], and [21].

### B. Unstructured Datasets

RUGD [6] is a benchmark unstructured dataset consisting of environments near creeks, vegetation, body of water, trails, and villages. Video was collected using a robot equipped with a Velodyne HDL-32 LiDAR and a Prosilica GT2750C camera and every 5th frame was used for training.

The TAS500 dataset [7] aims to distinguish between traversable and non-traversable regions. 44 different objects are categorized into nine groups: animal, construction, human, object, sky, terrain, vegetation, vehicle, and void. Infrequent classes are mapped to the closest category and specific class distinctions are consolidated leading to a final 20 classes. The data was collected using a MuCar-3 [24] with a vision system with a camera sensor that provides color images at 2.0 MP resolutions, [25] where pixel-wise masks were select from every hundredth recorded image.

The Rellis dataset [9] consists of synchronized sensor data collected using a Clearpath Robotics Warthog platform and the data includes different runways, aprons, lakes and ect..

## III. RELEVANT USES

Using prior knowledge of characteristics of an environment can support path estimation. The usage of both 3D terrain information and visual characteristics provide better results when used in combination rather than as a singular resource [26]. Models can be created to create colors image and assign traversability costs to regions based on the geometry and appearance. [27] Features based on image texture can help in binary classification of traversability based on on-board sensors such as IMU, motor current, and bumper switch [28]. Learning approaches to improve nearsightedness uses by using models trained on data seen at different points in time to be referenced at a later trajectory [28].

Synthetically attained data has been seen to improve the performance of deep neural networks on image segmentation and results show similar accuracy and results to that of real-world data in image classification. Furthermore if domain adaption is applied, it can provide not only similar but better results compared to the real-world datasets. [29]. The VAIL Dataset consists of realistic unstructured environments. Utilizing 25 different unstructured environments we hope to provide data that is more adaptable to make the binary decision of traversability and improve object classification accuracy. This work in the future can also be combined with existing approaches of synthetic image production and GAN networks for domain transfer between synthetic and real world data sets [30].

## IV. DOMAIN ADAPTATION

Domain adaption of semantic segmentation datasets allow using Unsupervised Domain Adaption(UDA) to label environments, which is both labor and time intensive.[31] UDA models are trained with labeled source data and unlabeled target data with the goal to achieve state of the art performance by reducing the domain gap between the source and target domains.

Methodologies for UDA include latent representation alignment of the two domains in feature space [32] and [33]. Reducing the visual difference between the two domains for input level adaptation [34] and [35]. Domain transferring images to train the segmentation model [36]. Adding a layer of discriminators to adapt predictions from two domains. [37] and [38]

Domain adaption can be utilized to make the process of data preparation for training easier. [39]

## V. DATA COLLECTION

The unreal engine provides photo realistic environments with presence of moving objects, various illuminations, and changing light conditions [15]. Data collection required both manual intervention when automation provided challenges. The TartanAir dataset [15] utilized similar methods to collect multi modal data including stereo rgb image and segmentation data.

### A. Hardware and Software

Data was collected on an Intel NUC NUC11PHKi7 11th Gen Core i7-1165G7 Quad-Core up to 4.70 GHz Processor, 32GB DDR4 RAM, 1TB PCIe NVMe SSD, GeForce RTX 2060 6GB GDDR6 Graphics running the Windows 11 OS. The Windows OS and Mac OS have strong support from both Unreal and AirSim [40] with hardware being a potential limitation. Epic Games Launcher was used to install Unreal Engine and download environments. AirSim is a plugin which provides a simulation platform for AI research to interact with a ground or air vehicle programmatically in Unreal Engine. AirSim can be used to retrieve images, get state, control the vehicle along with many other functionalities. Python 3.7 was used to interact with the AirSim API.

### B. Environments

Diverse simulated seasonal outdoor environments containing forested or natural environments in light conditions were focused on for this dataset. Some of the environments used here are semi-structured but these were very large environments with regions that fit our criteria. These environments were selected based on an predetermined criteria of large variations in geometry, appearance, shapes, objects containing irregular boundaries, and forested or mountainous.

Winter Environments that were used are Bleak Winter, Land, Foilage Winter, Nordic Auto Biome Winter, Northern Winter, Procedural Oak Forest w+Winter, and Viking Winter. These environments resemble a mountain or forested region during winter.

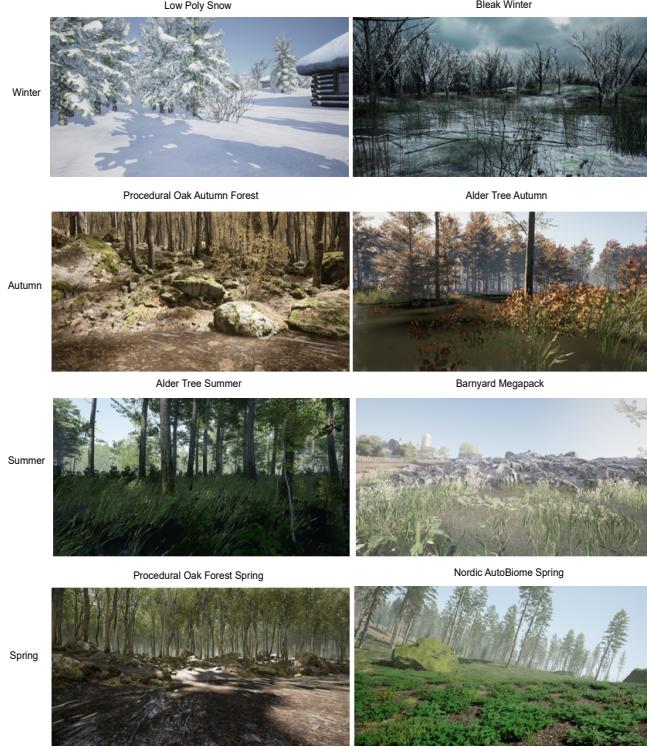


Fig. 2. Example rgb images of seasonal environments. These pictures demonstrate the unstructured, off-road, and forested characteristics of the environments.

Autumn Environments that were used are Alder Tree Autumn, Autumn Alley, Nordic Auto Biome Autumn, and Procedural Oak Forest Autumn. Majority of these environments were made up of forested areas with characteristics of fall such as a lower density of grass.

The Summer environments used Alder Tree Summer, Foilage Summer, Nordic AutoBiome Summer, Northern Summer, and Viking Summer. These were majority forested environments with trees full of leaf and dense green grass.

Spring environments used were Nordic Auto Biome Spring and Procedural Oak Forest Spring and were similar in characteristics to the summer environments.

Other environments include Barnyard Megapack, Medieval Village, Park Full, and Open Worlds Mediterranean all of that which had objects of various shapes and sizes and were forested.

### C. Data Acquisition

Majority of the pipeline for data collection was able to be done programmatically using Python with some manual intervention. AirSim provides a ground vehicle or aerial vehicle to capture data. ForestSim uses the UGV(unmanned ground vehicle) to collect data using three cameras attached the front left, front center, and front right. Rgb and segmentation images were collected using AirSim at 5 second intervals. The vehicle was directed on a specific path using time intervals and was very efficient in more open environments. Certain areas were challenging to navigate due to small untraversable objects leading to collisions causing

the ground vehicle to get stuck requiring keyboard control. Figure 3 demonstrates two different environments that shows an example of this issue.



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.

### D. Data Processing and Statistics

The majority of the data processing was performed on the segmentation images collected using AirSim. AirSim randomly assigns a id to each static mesh and then maps that id to an rgb value from an existing palate of 255 different rgb values. The segmentation images provided by AirSim presented a few challenges that were reconciled. Within certain environments the same object class was assigned the same rgb value while other times they were assigned different rgb values. Another challenge was that in different environments the same class was labeled with a different rgb value. For example the rgb value assigned to trees were different across environments. To consolidate this each environment was manually curated and a mapping was created to map a environment specific rgb value to a predetermined rgb value for that class. For example, for trees, all of the rgb values seen within an environment were recorded and mapped to an object id 1. An id of 1 was predetermined to be the id for tree in our dataset. This mapping was done for each and every environment used within our dataset. All of the same object classes within and across environments were converted from their AirSim rgb value into our dataset assigned rgb value. Through this process we were able to provide clear semantics on our dataset by providing the ground truth labels while avoiding redundancy in our data.

### VI. ANNOTATION STATISTICS AND ONTOLOGY

The number of times a specific object rgb pixel appeared in each segmentation image was counted and then averaged over total to determine the percent of pixels that contained that object class.

The classes that are included in the ForestSim dataset are grass, tree, pole, water, sky, vehicle, container, generic object, asphalt, gravel, mulch, rockbed, log, bicycle, person, fence, bush, sign, rock, bridge, concrete, table, building, void, and generic ground. The 20 different objects classes were each assigned a specific rgb. Generic ground includes all traversable ground. In AirSim any flat ground was labeled a specific rgb value, most likely as no static mesh was used for it during development. These are traversable flat regions. Generic container objects includes all generic objects that would involve collision. Generic container objects include

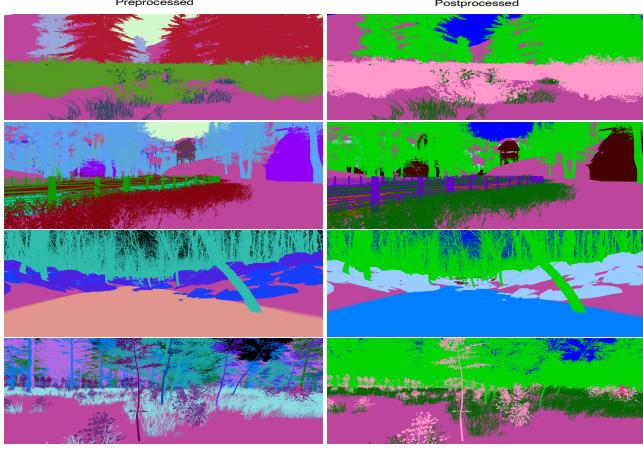


Fig. 4. Examples of segmentation images captured directly from AirSim are on the left. These images were processed by manually determining the object each rgb value corresponded with and using this approach were processed into the images on the right.

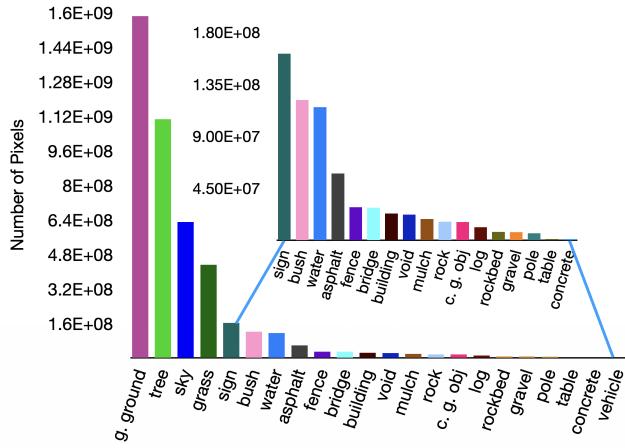


Fig. 5. Number of total pixels per class in the dataset, ordered in descending order

benches, trash cans, playgrounds, water containers, water fountains, log containers, and other similar objects.

This data can be applied to help improve an autonomous agent's maneuvering decisions in an unstructured environment by aiding in the decision of traversability. The approach used is in its infancy stage as it focuses on a simulated unstructured environments but it has shown promise with evidence. The data could help an autonomous agent make a binary decision of traversability in a static unstructured environment. Data sparsity exists in the data as vehicle, concrete, pole, gravel, and rock-bed make up a very small percent of data. This leads to a problem of identification of these objects which can lead to inaccurate decisions. Dynamic situations not readily available within AirSim and this data set does not provide that data. The approach used to collect data replicated using other simulation environments in combination with what AirSim and Unreal Engine provide could alleviate some of the existing sparsity issues related to dynamic behavior and data sparsity.

## VII. BENCHMARKS FOR DOMAIN ADAPTIVE SEGMENTATION

### A. Baselines and Experimental Setups

We have utilized state of the art techniques for training and testing on the ForestSim Dataset. These techniques are built off a unified framework for both implementation and evaluation using mmsegmentation. All of the models are built using a encoder and decoder pattern.

One approach to building a model was to use a pre-trained resnet50v1c model with a encoder of ResNetV1c with a depth of 50 with a decoder of PSPNet using Cross Entropy Loss with a loss weight of 1.0. Similarly four other models were built using a combination of a pre-trained resnet50v1c and resnet101v1c model with an Atrous Spatial Pyramid Pooling (ASPP) decoder using Cross Entropy Loss with a loss weight of 0.4 and the other a decoder of DepthwiseSeparable with Cross Entropy Loss with a loss weight of 1.0. The 4 models built from that combination were a pre-trained resnet50v1c with a ASPP decoder, pre-trained resnet50v1c with a DepthwiseSeparable, a pre-trained resnet101v1c with a ASPP decoder, and pre-trained resnet101v1c with a DepthwiseSeparable decoder

Two models were trained using an decoder type of MixVisionTransformer and encoder of Segformer utilizing Cross Entropy Loss with a loss weight of 1.0 with one model based off a pretrained mit-b0 model with a optimizer of AdamW with lr of .00006 and weight decay of 0.01 and the other model with just a pretrained mitb5 model.

Models were trained using a encoder of ResNet using a pretrained resnet50 with a decoder of Mask2Former that was more finely tuned and configured to also use a pixel level decoder MSDeformAttnPixel using various losses such as CrossEntropyLoss and DiceLoss with a optimizer of AdamW of lr of 0.0001 and weight decay of 0.05 and a scheduler of PolyLR. This same configurations were used to but with a pretrained resnet101.

A decoder of SwinTransformer with a combination of a pretrained swin tiny model and pretrained with swin small were also built. SwinTransformer as the decoder and Mask2Former as an encoder using an AdamW optimizer and PolyLR scheduler with a pretrained swin base model and a swin large pretrained model were also used to build a model.

In summary the models were trained with PSPNet with a backbone or decoder of ResNetV1c, DeepLabV3 with a decoder of ResNet50, ResNet101, DeepLabV3+ with a backbone of ResNet50, ResNet101, SegFormer with a backbone of MixVisionTransformer, Mask2Former with a backbone of ResNet and SwinTransformer.

### 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% was used for testing.

Training of models occurred on 4 nodes with each containing SUSE Enterprise Linux Server (SLES) version 15 with 256 GB of memory and two 64-core, 2.25 GHz, 225-watt



Fig. 6. Examples of ground truth annotations from the ForestSim Dataset. These include 3 of 25 different environment used for data collection. This dataset contains 23 different classes. The first row is the photo realistic rgb image collected from the environment and the second row is the corresponding semantic segmentation. Please note that these are the semantic segmentation images after post processing and consolidation.

TABLE I

RESULTING PERCENTAGES FROM VARIOUS ARCHITECTURES USED BEGINNING WITH THE PRETRAINED MODEL, ENCODER, AND DECODER.

| Method   | IOU $\downarrow$ | Pix. Acc. $\downarrow$ | M. Pix. Acc. $\downarrow$ |
|--|------------------|------------------------|---------------------------|
| resnet50v1c + ResNetV1c + PSPHead                      | 61.64            | 89.85                  | 72.14                     |
| resnet50v1c + ResNetV1c + ASPPHead                     | 61.87            | 89.91                  | 72.81                     |
| resnet101v1c + ResNetV1c + ASPPHead                    | 62.81            | 89.86                  | 73.13                     |
| resnet50v1c + ResNetV1c + DepthwiseSeparableASPPHead   | 59.16            | 89.31                  | 72.93                     |
| resnet101-v1c + ResNetV1c + DepthwiseSeparableASPPHead | 59.22            | 88.32                  | 69.56                     |
| mit-b0 + MixVisionTransformer + SegformerHead          | 61.82            | 90.52                  | 71.12                     |
| mit-b5 + MixVisionTransformer + SegformerHead          | 67.93            | 92.05                  | 76.42                     |
| resnet50 + ResNet + Mask2FormerHead                    | 67.48            | 91.34                  | 75.77                     |
| resnet101 + ResNet + Mask2FormerHead                   | 65.80            | 91.29                  | 74.61                     |
| swin-base + SwinTransformer + Mask2FormerHead          | 74.50            | 92.57                  | 82.30                     |
| swin-large + SwinTransformer + Mask2FormerHead         | 75.31            | 92.65                  | 82.68                     |
| swin-tiny + SwinTransformer + Mask2FormerHead          | 70.46            | 92.14                  | 79.79                     |
| swin-small + SwinTransformer + Mask2FormerHead         | 74.02            | 92.39                  | 81.39                     |

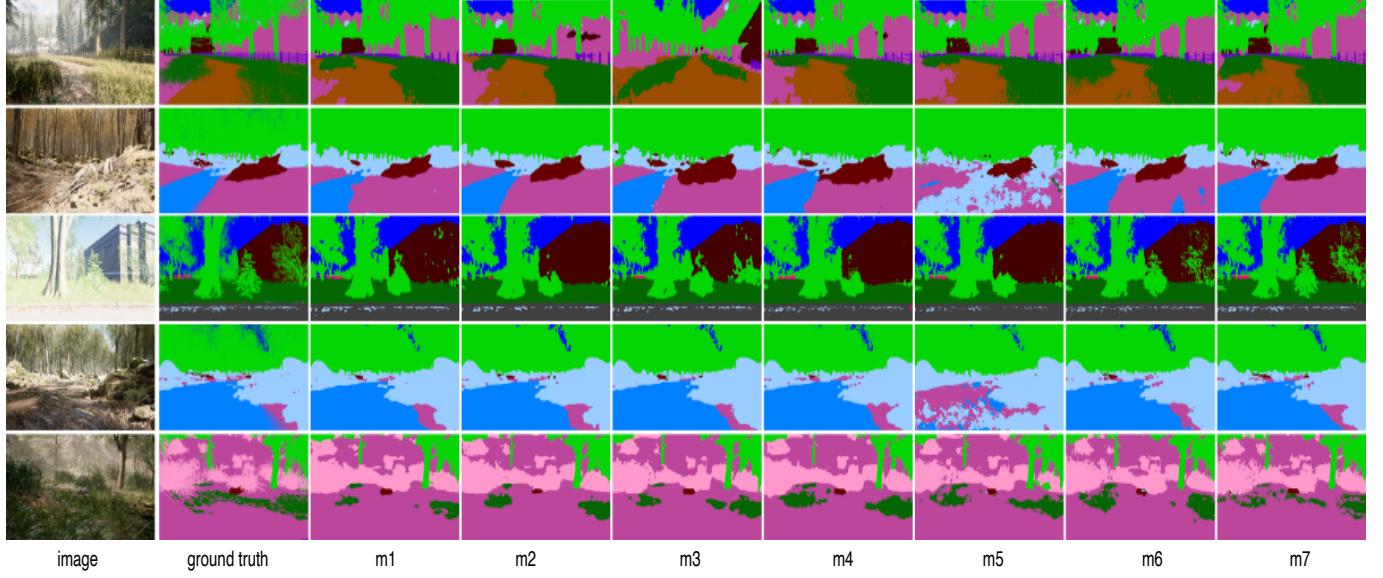


Fig. 7. The original image, the ground truth, and predicted image annotation for models 1 to 7.

AMD EPYC 7742 processors running 4 tasks per node and 4 NVIDIA A100 GPUs per node. The number of iterations for training varies based of 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

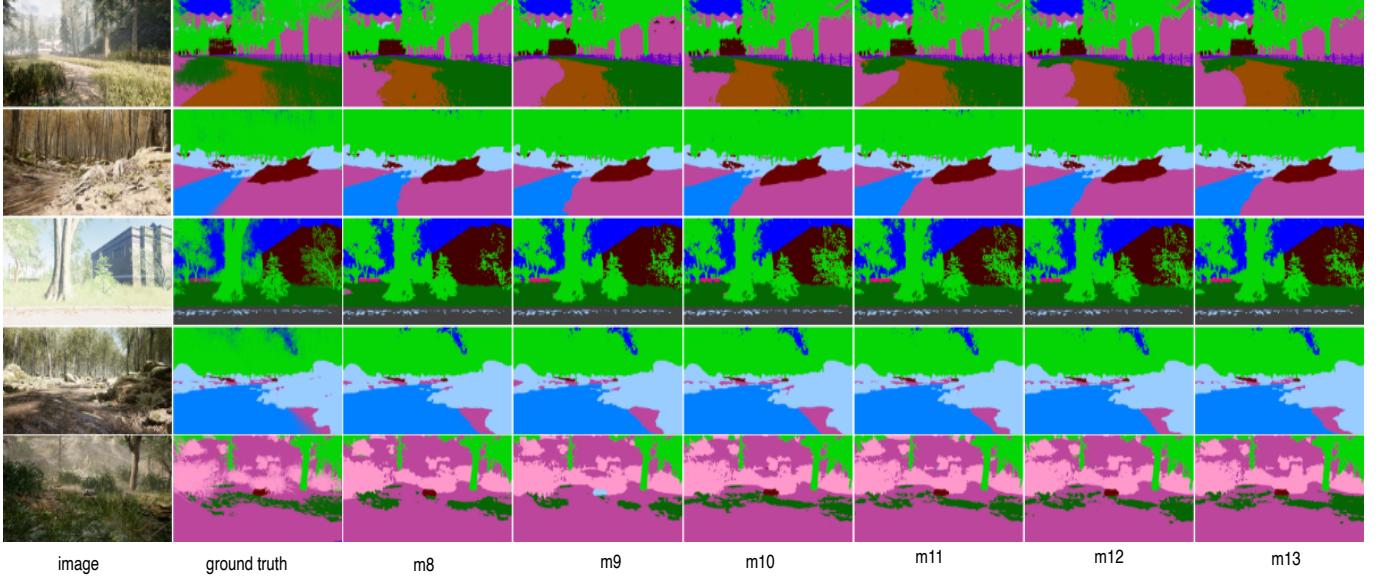


Fig. 8. The original image, the ground truth, and predicted image annotation for models 8 to 13.

segmentation standards metrics such as Mean IOU and pixel wise classification. Mean IOU is the average IOU between all classes [41]. The IoU for each class is computed as  $\text{TP}/(\text{TP}+\text{FP}+\text{FN})$ . Mean pixel wise classification accuracy is also used which is the average classification accuracy per model and that evenly weights each class.

### C. Analysis and Experimental Evaluation

Using the models to make predictions on the randomized test, the performances are reported in Table 1. This dataset provides a very good baseline for training on a unstructured simulated dataset. The high scores seen in the Pixel Acc column shows that the objects are learned well and do predict with high accuracy the highly represented objects such as tree, sky, and generic ground(traversable land).

The sparsity of the dataset presents a challenge but approaches of synthetic image production and GAN networks for domain transfer between synthetic and real world data sets can provide beneficial results in this domain.[30]

Table 1 summarizes all of the model results on the test set. Based on Table 1 it can be seen that the models using the SwinTransformer perform the best. Object classes predictions performed well with table IoU and Accuracy being the lowest out of all classes. This can most likely be attributed to similarly shaped objects that are over represented leading to predict table as that object. Data balancing is something to strongly consider once methodologies are applied to counter data sparsity.

## VIII. CONCLUSION

ForestSim is a unique dataset that presents a simulated dataset geared towards unstructured environments. Here we collected images from highly realistic offroad, forested, and mountainous environments of varying seasons that met the criteria of an unstructured environment. Though the dataset

contains 20 classes in total, bicycle was unable to be predicted due to the sparsity of data. All of the images in the ForestSim dataset are captured from a simulated environment from a UGV vehicle in illuminating bright conditions. The data sparsity presents a challenge but there are tested ways to alleviate the issue of data sparsity such as synthetic image production and GAN networks. Our future plan is to locate more environments available within Unreal Engine and to add more natural off road environments to the ForestSim dataset. Other simulated environments also exist, which may provide more diversity in their environments, and the pipeline used to process and train images here can be adapted to that simulated environment. After collecting more data and improving our models testing on existing datasets would be the next course of action.

## REFERENCES

- [1] 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.
- [2] 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.
- [3] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Zisserman. The pascal visual object classes (voc) challenge, 2010.
- [4] Bolei Zhou, Hang Zhao, Xavier Puig abd Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset, 2017.
- [5] 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.
- [6] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A rugged dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

- [7] Kai A. Metzger, Peter Mortimer, and Hans-Joachim Wuensche. A fine-grained dataset and its efficient semantic segmentation for unstructured driving scenarios, 2021.
- [8] Bhakti Baheti, Shubham Innani, Suhas Gajre, and Sanjay Talbar. Effunet: 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.
- [9] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis, 2022.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] Di Feng, Christian Haase-Schuetz, Lars Rosenbaum, Heinz Hertlein, Claudio 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.
- [15] 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.
- [16] 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.
- [17] Matthew R. Walter, Matthew Antone, Ekapol Chuangsawanich, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Yuli Friedman, James Glass, Jonathan P. How, Jeong hwan Jeon, Sertac Karaman, Brandon Luders, Nicholas Roy, Stefanie Tellex, and Seth Teller. A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments. *Journal of Field Robotics*, 32(4):590–628, 2015.
- [18] Ester Martínez and Angel P. del Pobil. Robust object recognition in unstructured environments. *Advances in Intelligent Systems and Computing*, 193:705–714, 01 2013.
- [19] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudio Gläser, 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.
- [20] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühllegg, 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.
- [21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [22] 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.
- [23] 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.
- [24] Michael Himmelsbach, Thorsten Luettel, Falk Hecker, Felix Hundelshausen, and Hans-Joachim Wuensche. Autonomous off-road navigation for mucus-3 - improving the tentacles approach: Integral structures for sensing and motion. *KI*, 25:145–149, 05 2011.
- [25] Alois Unterholzner and Hans-Joachim Wuensche. *Active Multifocal Vision System, Adaptive Control of*, pages 46–64. Springer New York, New York, NY, 2012.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] Alireza Shafaei, James J. Little, and Mark Schmidt. Play and learn: Using video games to train computer vision models, 2016.
- [30] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine, 2016.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation, 2018.
- [36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.
- [37] 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.
- [38] 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.
- [39] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. Unsupervised domain adaptation for semantic segmentation by content transfer, 2020.
- [40] Shital Shah, Debadatta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
- [41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.