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 decisions. To provide these agents with higher intelligence, many datasets have been collected to provide the context needed 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 the ForestSim: A Simulated Dataset collected in a simulated setting, which consists of 2084 annotated images collected using a ground vehicle from 25 different environments which aims 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 prospective, and RGB images of the environment. Images are processed and 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. Data set link: `www.dataset.com`**

I. Introduction

Data sets collected have shown to improve decision making. Existing structured datasets where boundaries are clear and are of urban environment scenarios, vehicles driving, pedestrians walking, and built pavements with structured buildings include [1], [2], [3], [4], and [5]. In current datasets there exists a gap in available data for unstructured outdoor environments. This can include disaster scenarios, off road maps, and outdoor forested environments with unclear boundaries and irregular objects. [6], [7], [8], and [9]

These datasets have laid a foundation for improving object detection, classification, and semantic segmentation and have led to the development of benchmark data sets. [10], [11], [12], and [13]. The data featured in these datasets reflect real world interactions. These datasets have demonstrated that they have improved object detection and that these datasets provide an agent with a higher level of intelligence in understanding the environment and context, especially when the environments are represented well in the data. [14]

There have been efforts to close this gap that existing in unstructured environments. [6], [7], [8], and [9] but there are many challenges that are presented due to lack of resources to be able to collect in these types of environments and the are difficult to traverse and are location in only certain regions.

Thanks info...

There is also a simulated approach to collecting data that has been used to push and benchmark existing slam algorithms in structured environments. [15] This process can be reused against to collected data in a simulated setting of unstructured environments.

Fig. 1. An image collected from the Alder Tree Environment, an example of a forested environment during the Autumn season. This demonstrates characteristics of a unstructured environment where edges of objects are challenging to discern and certain objects are taller such as the grass(dark green) and bushes(pink). You can see that the trees blend into one another making it more challenging to determine the object class.

Unstructured Environments are classified as rough unstructured terrain. There are large variations in geometry, appearance, and shapes where tall grass and rough terrain may be interpreted as non-navigable terrain. In unstructured areas there is more ambiguity in visual perception. [16] These environments can include agricultural fields, orchards. Challenges in these environments include illumination occlusion and unconventional shape, size, and color. Use can include 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]

Our data set attempts to provide data that consists of images that contain changing light conditions, low illumination, unique angles,objects of varying shapes and sizes that could provide ambiguity in if it is a region that can be traversed or not.. to close gap to improve accuracy on less structured areas such as mountainous regions and forested areas, collection on differing seasons including summer, winter, fall, and spring. Provide an newer, more innovative, less resource consuming approach to collecting data, and work to create a reusable data collection pipe and post processing pipeline to provide an accessible approach to data collection and processing.

The data consists of 2400 RGB images with pixel wise ground truth annotated labeled images for learning and evaluation. These were collected from 25 different environments with varying seasons when available. The environments were already created and available through Unreal. ForestSim: A Simulated Dataset provides high quality, highly realistic images collected using Unreal Engine.

Benchmarks are performed using up to date, state of the art

methodologies using mmsegmenation with various models provided through mmsegmenation. Our results are positive and when the objects are well represented well in the images they are predicted with a high accuracy. Limitations are based on the quality of environments or what is available but with high quality environments on unreal our results show that a simulated approach can be a viable alternative. [6] and [9]

II. Segmentation Datasets

Semantic Segmentation Datasets aim to partition an image in meaningful parts using pixel wise annotation. This has begun to become particularly important and more valuable in autonomous driving to provide data for vehicles to make safer decisions. Semantic segmentation can be classified into one stage or two stage pipelines. The one stage pipeline can utilize a encoder to create hierarchical representations of an image with a backbone such as ResNet where as the decoder up samples through convolutional layers low dimensional features to original resolutions and these feature maps are used for pixel wise prediction. The pixel value provides context as to the class of that objects ranging from grass, tree, sky, pole, ect [19].

A. Structured Datasets

Mapillary [20] provides a benchmark data set to classify traffic signs. KITTI [21] is a dataset that consists of classes such as building, tree, sky, car, sign, road, pedestrian, fence, pole, sidewalk, and bicyclists that were annotated manually by researchers. The ApolloScape [22] provides data from various sites, cities and day times focusing on providing semantic segmentation data by integrating camera videos, consumer-grade motion sensors (GPS/IMU), and a 3D semantic map. Many of this data is collected from a small ground vehicle mounted with multiple sensors that capture data. [2], [22], and [21] There are other existing segmentation dataset that exist for urban environments [1], [2], [3], [4], [5], [23] and all of these heavily focus on structured areas and thus are extremely useful for driving in urban areas.

B. Unstructured Datasets

There is extensive research and datasets for urban structured environments but the lack of structure in off road environments makes it difficult to use these structured datasets for navigation decisions in unstructured environments. Some datasets that are existing of unstructured environments. RUGD [6] is a unstructured dataset where data was collected from environment such as a creek near vegetation with a body of water, a park with buildings and paved roads, a trail with non-paved areas and gravel, and a village with buildings and limited pavement. Here the data was collected as video and every 5 frame was used in the training. The data was collected using a robot equipped with a sensor payload consisting of a Velodyne HDL-32 LiDAR, a Garmin GPS receiver, a Microstrain GX3-25 IMU, and a Prosilica GT2750C camera. The camera is capable of 6.1 megapixels at 19.8 frames per second, but most sequences are collected at half resolution and approximately 15 frames per second.

The TAS500 dataset [7] aims to distinguish between drivable and non drivable surfaces and they collect data consisting of 44 class labels that 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 23 classes. The data was collected using MuCar-3 [24] with a vision system mounted on MarVEye, a camera platform, with a camera sensor that provides color images at 2.0 MP resolutions, [25] with data collected with a frame rate of 10 Hz with pixel-wise masks provided for every hundredth recorded image.

Rellis dataset [9] was collected using a Clearpath Robotics Warthog platform and the data includes different runways, aprons, terrain, forested areas, bushes, pastures, and lakes. Rellis consists of five sequences of synchronized sensor data including RGB camera images, LiDAR point clouds, a pair of stereo images, high-precision GPS measurement, and IMU data. All captured in off road environments.

These unstructured data sets aim to close the gap and provide data and some multi sensor data to improve algorithm and model performance in unstructured off road environments. We chose our environments based on the characteristics of unstructured environments described above.

III. Relevant Uses

Datasets are used to provide context about existing environments to help determine if areas of the environment are traversable or non-traversable. Using prior knowledge of expected characteristics of an environment can support path estimation. The usage of both 3D terrain information and visual characteristics provide be results when used in combination rather a as a singular resource [26]. Models can be created to create color image and assign traversibilty costs to regions based on the geometry and appearance. [27] Features based on image texture can help in binary classification of if a region is traversable or non-traversable 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].

Here the use of this synthetically created dataset is to provide a dataset to run existing state-of-art models to determine if existing models adapt and perform well on synthetic unstructured environments. Synthetically attained data has been seen to improve the performance of deep neural networks on image segmentation and results show that a convolutional network trained with this type of data can achieve similar accuracy and results to that of real-world data in image classification. Furthermore if domain adaption is applied, 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 environments we hope to provide data that is more adaptable to provide improved environment context to make the binary decision of traversability. 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. Labeling is both labor intensive and time intensive. This requires training of a segmentation model. [31] In this case of UDA for semantic segmentation the model is 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. Many different methodologies have been suggested and attempted, Latent representations 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 which are uses as domain-transferred 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. This work does not include the usage of domain adaption to create and add labels to images. For the dataset present in this paper objects were labeled manually per environment with a specific class id and then converted into a pre-specified rgb value per class and thus performed manually for the ForestSim: A Simulated Dataset but work is available to make the process of labeling easier with state of art performance. [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 data quality depends on the quality of the environments that were already created and available in Unreal. Time and effort is required to search for and determine what environments meet the criteria of environment needed for the current research goal. Our criteria for environments were forested, off road, irregularity in the environment structured with irregularly shaped objects of different shapes and sizes, seasonal, mountainous, and unclear object boundaries. The TartanAir dataset [15] utilized similar methods to collect multi modal data including stereo RGB image, depth image, segmentation, optical flow, camera poses, and LiDAR point cloud with focus on structured environments.

A. Resources

Data was collected on an Intel NUC running an Windows OS. The Windows OS is supported by both Unreal and Airsim which are essential to set up the environment for data collection. There is also support for the MacOS but the built in graphics hardware was not powerful enough to run more CPU and GPU intensive environments and a more power machine the Intel NUC was utilized. Epic Games Launcher was used to install Unreal Engine and download environments that met the criteria. Airsim is a plugin which provides a simulation platform for AI research which exposes API's to interact with a ground vehicle car or air vehicle drone programmatically in Unreal Engine to retrieve images, get state, control the vehicle along with many other functionalities. Airsim was utilized to collect rgb, segmentation, and depth perspective images along with camera poses including the camera intrinsic matrix from environments running on Unreal Engine. Python 3.7 was used to interact with Airsim API's to automate data collection.

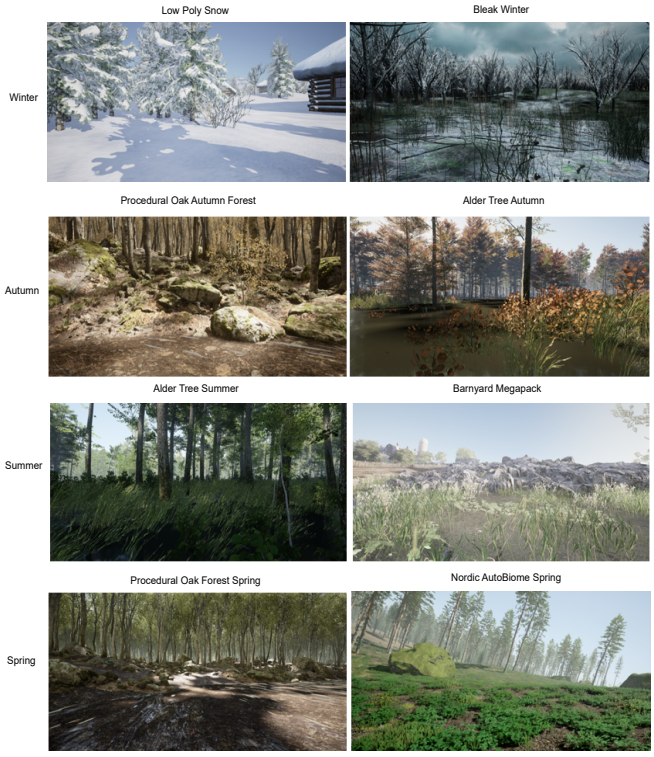

Fig. 2. Rgb example images of various seasonal environments data was collected using. These pictures demonstrate the unstructured, off-road, and forested characteristics of the environments.

B. Environments

Diverse outdoor environments simulated using the unreal engine, consisting of seasons Spring, Autumn, Summer, and Winter in light conditions were the environments focused on for this data. Environments that were illuminated well were used to ensure that the image can be made out clearly. Some examples can be seen in figure 2.

The primary focus for this dataset was to collect data from outdoor environments that contained primarily forested or natural environments which represented characteristics of unstructured environments. Some of the environments used here are semi-structured but these were very large environments where there were areas that fit our criteria. These environments were selected based up an predetermined criteria where within the environment large variations in geometry, appearance, and shapes where tall grass and rough terrain. Environments that were majority natural were

selected even if human structures made of parts of the environment, such as city surrounded by trees and a nature like composition. These

Winter Environments that were used are bleak winter, land, foilage winter, nordic auto biome winter, northern winter, procedural oak forest winter, and viking winter. The winter environments consisted of environments with snow and ice on the ground. Land was on the environments that replicated a cold mountain region, while the other's resembled a forested environment in winter.

Autumn Environments that were used are alder tree autumn, autumn alley, nordic auto biome autumn, and procedural oak forest autumn. Autumn environments included a semi structured environment in an alley way with a forest background and tall grass and bushes. Others were forested and environments where leafs had fallen off trees and grounds had color and 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. Many, if not all of these environments are forested environments with trees full of leaf and dense green grass. Spring environments used were nordic auto biome sprint and procedural oak forest spring. These environments were very similar to summer environments and are both forested environments.

Other environments are barnyard megapack is very characteristic of a large farm land containing the irregularities and regularized structures normally seen in a farm. City is an environment that was included as there were areas within this environment that are forested, different shaped grass, bush, and trees, also includes buildings in surrounded by forested green areas. Medieval Village was also included which contains structure in the homes and buildings but are surround by a forest and mountainous region that lack structure. Park full was also included as it includes forested areas with trees, grass, bush, and hill regions. The last environment was open worlds mediterranean that which had various shaped bushes, grass, trees, and rock in a forested mountainous area.

Our goal with the environments was to be as inclusive as we could and for that reason some semi-structured environments were also included.

C. Data Acquisition

Majority of the pipeline for data collection and preparation was able to be automated. The process included using and installing Unreal Engine and then installing and adding the Airsim plugin to environments that data is to be collected from. Airsim allows the usage of a ground vehicle and aerial vehicle to use to traverse an environment for data collection. This data set utilizes the ground vehicle to collect data using three cameras attached the front left, front center, and front right. Three types of images were collected at interval of 5 seconds, these included RGB, Depth Perspective, and Segmentation images all of using are available within Airsim. The Airsim plugin provides an interface for the python Airsim client to communicated directions to the UGV(unmanned ground vehicle) within Unreal Engine. Along with the three images types Airsim also provides other data such as features of the camera including the camera intrinsic matrix. Running the python scripts to help automate collection improved overall efficiency in time spent to collect data. The directed path was a pattern of movement that was iteratively repeated using time interavls and was very efficient in more open environments. Environments such landscape-mountains had a high number of objects that led to collisions and thus needed manual intervention at times to help the ground vehicle get unstuck and certain areas were challenging to navigate with the script as they were small areas filled with objects that could lead to collisions causing the ground vehicle to get stuck so data collection in those situations were done manually. 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 of the ground vehicle and data collection could not be automated where on the right, an example of a less dense environment, a python script was able to be used to collect data.

D. Data Processing and Statistics

The majority of the data processing was performed on the segmentation images provided by Airsim. Airsim's behavior is to randomly assign a id to each static mesh or object of specific type then map 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 through a data processing pipeline. Within certain environments the same object classes were assigned the same rgb value while other times this was not the case. In certain environments, the provided segmentation images within a certain environment assigned the same class of object differing segmentation ids leading to differing segmentation rgb values. This was on a case by case basis based on how the environment was created. Another challenge was that in different environments the same class was labeled with a different id and different segmentation rgb values. For example the segmentation rgb values 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 values to a predetermined class id. For example all of the rgb values seen within an environment for trees were recorded and mapped to an object id 1. An id of 1 was predetermined to be the id for tree in our entire dataset. This mapping was done for each and every environment used within our dataset. Finally

once a mapping was done for all of the environments, all of the same object classes within and across environments were then 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 providing the true labels for the images collected while avoiding redundancy in our dataset by creating and converting same object classes to a predetermined segmentation id and rgb value.

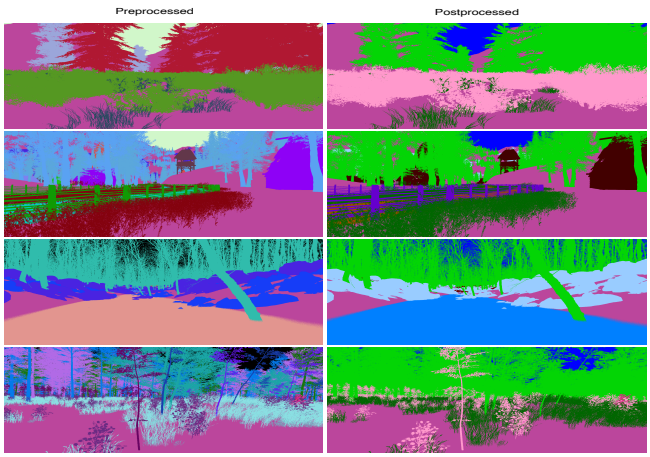

Fig. 4. Examples of segmentation images captured directly from Airsim are on the left. These images were processed by manually labeling each rgb value, for each environment, to a predefined id representing that class. Each of the predefined ids correspond to a specific rgb and using that mapping the pre processed images were processed onto the images on the right.

VI. Annotation Statistics and Ontology

The number of times a specific object rgb pixel appeared in each segmentation image was counted and using that we were able to determine the percent of pixels that contained that object class. Our focus in data set was to provide images of terrain and objects that would be helpful in providing an autonomous agent to determine if an certain area is maneuverable or not maneuverable and more specifically in a unstructured environment.

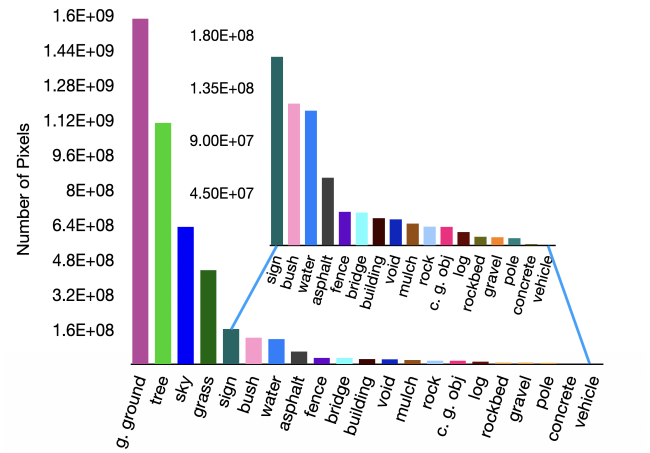

Fig. 5. Number of total pixels per class in the dataset, ordered in descending order

The classes that are included in the Vail 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 23 different objects classes were assigned a specific rgb. For each environment a mapping from the Airsim assigned rgb value to the class object was manually determined by manually labeling the data in each environment. 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 but were traversable flat regions. Generic container objects includes all generic objects that would involve collision, the include benches, trash cans, playgrounds, water containers, water fountains, log containers, and other objects outside of objects that are more commonly associated with other unstructured datasets. Figure 4 shows the classes represent in this data along with the percent ratio of class object pixel to total pixels.

This data can be applied to making maneuvering decisions in an unstructured environment and this approach used is in an infancy stage as it focuses on a synthetic unstructured environments. This data could help an autonomous agent make a binary decision of traversability in a more static unstructured environment but type of terrain is not differentiable as their is sparsity in that data and also dynamic situations were not readily available within what Airsim supports. Data sparsity is seen in vehicle, table, pole, gravel, rockbed which leads to a problem of differentiating type of terrain as differing ground vehicles can traverse different terrains for example if the vehicle is tracked or wheeled.This approach used here replicated using other simulation environments in combination with Airsim could alleviate some of the existing sparsity issues related to dynamic behavior and terrain types.

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 Vail Dataset. For all approaches a model of type encoder decoder is used for training a model on the Vail Dataset.

One approach 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 another two approaches used the same process where one of the two rather used 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. Both ASPP decoder and DepthwiseSeparable decoder approaches were also used to train another model with a pretrained resnet101v1c model. These resulted in 5 different models of various combinations.

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

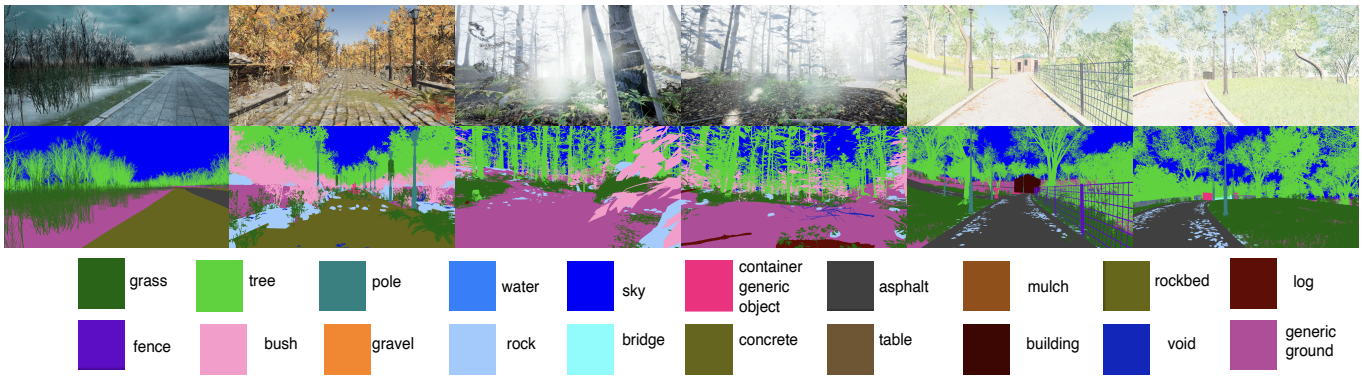

Fig. 6. Examples of ground truth annotations from the Vail Dataset. These include 3 of 25 different environment used to collected data from using. This dataset contains 20 classed of pixel wise labeled images. The first row is the photo realistic rgb image collected from the environment that corresponds to the semantic segmented image below in row 2.

TABLE I

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

Method	IOU↓	Pix. Acc.↓	M. Pix. Acc. ↓
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

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. Two different models were trained using these configurations.

More 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 approach was used also on a pretrained resnet101 and then also another model with a different decoder of SwinTransformer using a a pretrained swin tiny model, and another with the same decoder but pretrained with swin small. Another model used a SwinTransformer as the decoder and Mask2Former as an encoder with a pretrained swin base model using an AdamW optimizer and PolyLR scheduler. Lastly a model was also trained using these configurations but with a swin large pretrained model.

Overall 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 a 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 AMD EPYC 7742 processors running 4 tasks per node and 4 NVIDIA A100 GPUs per node. The iteration number varies between the models from 40,000 to 160,000 iterations.

Metrics to measure performance include standard segmentation standards metrics such as Mean IOU and pixel wise classification. Mean IOU is the average IOU between all classes [40]. The IoU for each class is computed as TP/(TP+FP+FN). Mean pixel wise classification accuracy is also used which the average classification accuracy per model and that evenly weights each class. A mean pixel classification per class is also presented.

C. Analysis and Experimental Evaluation

Using the models to make predictions on the randomized test, the performances are reported in Table 1. This dataset

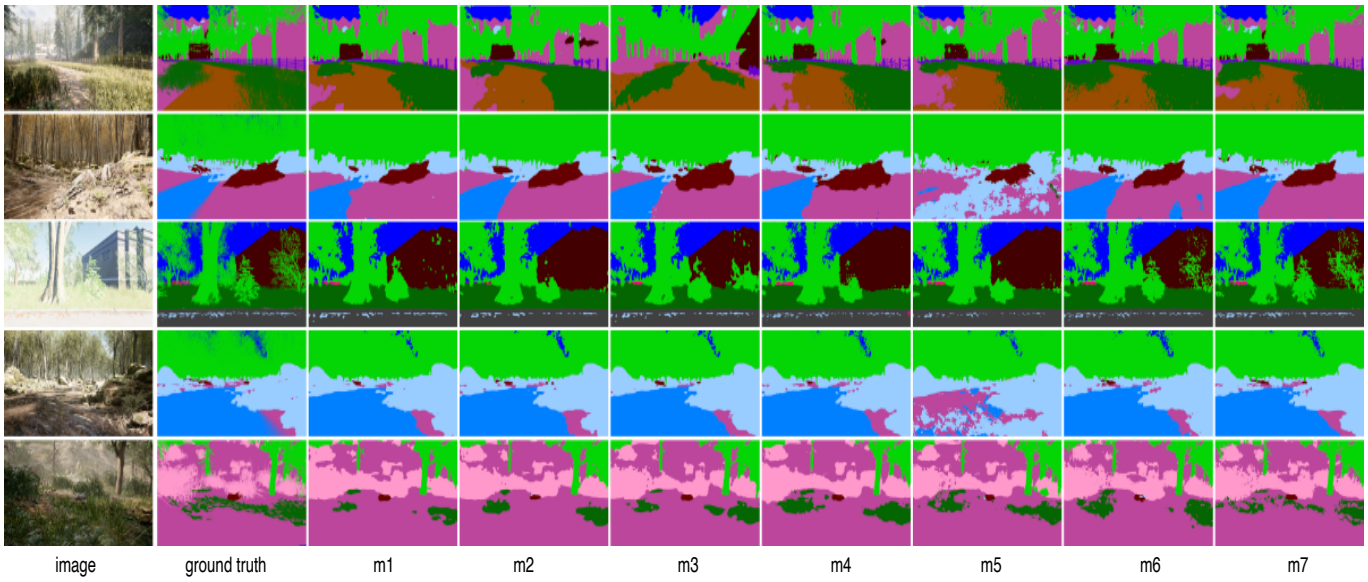

Fig. 7. The original image, the ground truth, and predicted image annotation for models 1 to 7.

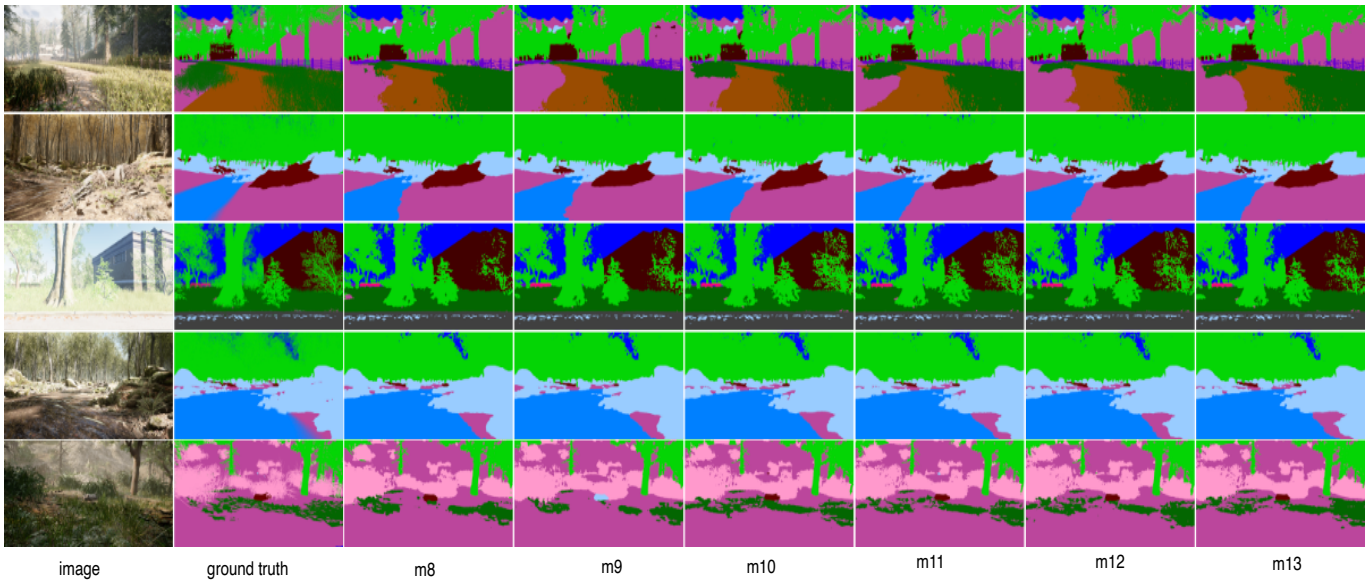

Fig. 8. The original image, the ground truth, and predicted image annotation for models 8 to 13.

provides a very good baseline for training on a simulated dataset of the unstructured type. The high scores seen in the Pixel Acc column shows that the objects are learned well and do predict highly represented objects such as tree, sky, and generic ground(traversable land).

The sparsity of the dataset for lower represented models and that does present 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]

Note that the pixel accuracy is high as these the test set is not a held out test set bit. Vehicle, bicycle, person, and sign were predicted as a nan or 0 due to sparsity of the data. Bridge, table, container generic objects also had very low IOU's. The container generic objects consisted of any object that we did not define in the list to classify so it had various shapes and the most likely contributed to the low IOUs. The reason was most likely also similar for bridge and tables and this was due to sparsity and different shapes and sizes. IOU's are important as they contribute to steering and the low IOU's may translate to errors. Looking at figures 7 and 8 the segmentation output for all of the 13 base line models. The models perform well on the test set. Model 5 does not perform very well in areas of water that have an object beneath it but the other preformed well. Also looking at the last row in both figure the model did not perform well

in predict grass when the lengths are different and regions of shorter grass it predicted as generic ground. This is seen in other segmentation outputs.

VIII. Conclusion

ForestSim: A Simulated Dataset is a unique dataset that presents a simulated dataset geared towards unstructured environments. This dataset contains 23 classes 4 of which are unpredictable due to the sparsity and this includes vehicle, bicycle, person, and sign. Here the dataset contains images where the data does not contain discernible edges and these images are captured in a simulated environment from a moving vehicle contain illuminating conditions in the rgb images that could negatively effect the testing. The data sparsity present a challenge but there are tested way to alleviate the issue of data sparsity. Our future plan is to locate more environments available within a simulated environment to use other natural off road environments and add to our data set. Furthermore to use already understood approaches to generate images using data sparsity training to tackle data sparsity and then train the model on this new data. Using those models we would then test this against held out test sets.

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 rugd 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. 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.

[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, 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.

[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 Chuangsuwanich, 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, 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.

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

[22] 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.

[23] 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.

[24] Michael Himmelsbach, Thorsten Luettel, Falk Hecker, Felix Hundelshausen, and Hans-Joachim Wuensche. Autonomous off-road navigation for mucar-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 seg-

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