Robust Depth-Aided Segmentation for Drivable Region Detection in Challenging Environments

Vipul Ramtekkar[1], Liza Dahiya[1], Nikunj Shah[1], Kenji Nishimiya[1], Takahiro Kuroki[1],
Chaehyeon Song[2], Ayoung Kim[2], Myung-Hwan Jeon[2*]

***Abstract*— This paper proposes a method for detecting drivable regions in challenging terrains using RGB-D data. By integrating depth information with semantic segmentation, our approach significantly improves detection accuracy across diverse landscapes. Leveraging the SegFormer architecture, we effectively distinguish drivable from non-drivable areas. Additionally, we introduce a depth-based refinement mechanism to ensure reliable performance in real-world scenarios. Extensive evaluation in both off-road and on-road environments confirms the effectiveness of our approach. Using the SA-1B dataset with grounded SAM, our method achieves precise delineation of road classes during training. Overall, this work advances autonomous navigation systems by providing a comprehensive solution for drivable region detection in complex terrains in real time, even on edge computing devices.**

I. Introduction

Understanding drivable region in uneven terrain is a challenging problem which is essential for all terrain mobile robots which are deployed in rough terrain environments. During this phase, critical road features such as road boundary lines play a vital role. However, road signs may not always be present on such environments. To deal with off-road environments, we should consider extra factors. One such factor is traversability, which refers to the ability of the vehicle to access an area physically. This factor takes into account the maximum gradient and height that the mobile robot can traverse. By estimating the variation of surface elevation, it is possible to identify and exclude regions that are physically inaccessible to the Mobile robot.

Determining a drivable area based solely on traversability is not a comprehensive criterion. While a region may be physically traversable, there may only be a limited number of directions suitable for driving to optimize traffic flow and ensure safety. This attribute is referred to as ”drivability.” In off-road environments, explicit road boundaries are not available. However, areas frequently used for transportation display distinctive textures compared to their surroundings. By analyzing the texture variation of the ground surface, vehicles can effectively identify drivable regions of off-road environments.

Our primary objective is to ascertain the drivable area in front of the vehicle by employing two onboard cameras. Given the absence of sufficient traffic information in off-road environments, estimating ground geometry and texture information through image analysis is critical. To achieve this goal, we intend to robustly combine the depth information obtained from the stereo camera setup as a post-processing to the segmentation results generated by deep learning model that will be trained on public datasets.

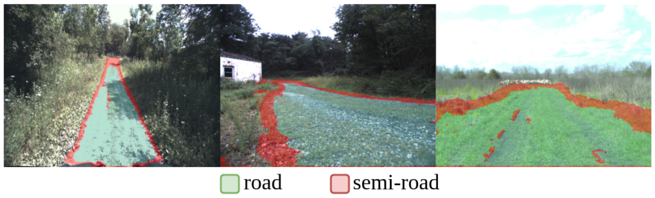


Fig. 1: Performance in unstructured outdoor terrains

II. Related Work

Drivable region recognition in off-road environment depends on the road traversability as well as on the capability of the mobile robot. Many traditional approaches, as summarized in [1], have been extensively explored over time. LiDAR based methods that either only use only PointCloud data [2] or fuse it with RGB image data [3, 4] are also garnering a lot of interest from the field. Despite the depth advantage that LiDAR offers, these approaches are less robust with changes in sensor capabilities.

Semantic segmentation methods to obtain traversable region in off-road environment has been explored in various works such as [5–8]. However, these methods typically rely on monocular image data (RGB), which limits their ability to capture 3D spatial intuition necessary for real-world applications.

A. *Using stereo depth data as a model input*

Previous works such as [9–12] focus on directly using multi-camera RGB or RGB-D data as input to an encoder-decoder model architectures in order to exploit the depth information for better drivable region detection. Based on the results published they do have a notable performance improvements in the respective data domain. These methods especially rely on the assumption that deep neural networks being used will be able to capture the relevant features given enough labeled examples rather than exploring the physical importance of depth itself. Conventional image processing methods have also been surveyed in [1], but they also use domain specific

*This work was supported by Honda R&D Co., Ltd.

[1]V. Ramtekkar, L. Dahiya, N. Shah, K. Nishimiya and T. Kuroki are with Solution System Development Center in Honda R&D Co,. Ltd., Wako-shi, Japan [ramtekkar_vipul,liza_dahiya, shah_nikunj, kenji_nishimiya, takahiro_kuroki]@jp.honda

[2]C. Song, A. Kim, and M. Jeon are with the Department of Mechanical Engineering, SNU, Seoul, S. Korea [chaehyeon, ayoungk, myunghwan.jeon]@snu.ac.kr

knowledge to perform performance improvements. Due to this limiting requirements of large domain specific data we look into more first principles approach described in the next subsection.

B. *Using stereo depth data for post-processing*

The focus of our research is also to develop a model for embedded devices with real time inference. Traditional RGB-D data-based models as explained above are inherently slower due to their multi-encoder based designs. Thus, incorporating depth based rejection of the segmentation area to improve the segmentation output can help maintain the real time inference. Using depth data as a means of correction has had been explored in the past [13]. But in contrast to existing approaches our method is invariant to roughness and can also handle vehicles according to their traversing capability.

III. Dataset Generation

Several open-source datasets, including Yamaha-CMU Off Road [14], CAVS-CaSSed [15], CAVS-CAT [16] and OFFSED [7] are available for off-road environments. Additionally, RELLIS-3D [17] and RUGD [18] are other two popular datasets for off-road robotics. However, in our study, we exclusively utilize the first four datasets: Yamaha, CAVS-CaSSed, CAVS-CAT, and OFFSED. The rationale for this selection is elaborated in Appendix I-A. Table I provides an overview of these public off-road datasets used in our study.

Dataset	Images	Classes	Modality
Yamaha-CMU Off Road [14]	1076	8	RGB
CAVS-CaSSed [15]	1679	6	RGB
CAVS-CAT [16]	12,300	-	RGB
OFFSED [7]	1018	19	RGB

TABLE I: Open-Source datasets for off-road environments

These datasets are not diverse enough to be used directly in our research. Hence, to augment both size and diversity, we employed a vision language foundation model, Grounded-SAM [19] for creating segmentation masks.

Grounded-SAM combines two key foundation models, SAM (Segment Anything Model) [20], general-purpose segmentation model trained on large-scale dataset and Grounding DINO [21], text-prompt based object detection model. This integration enables Grounded-SAM to detect and segment any regions based on arbitrary text-prompt inputs.

Our dataset generation pipeline (see Fig.2) comprises a pre-trained Grounded-SAM model and a Road-Classifier. Grounded-SAM model takes images and text-prompts (ex: “dirt road”,“puddle”, etc.) and outputs a segmented mask, which we refer as ***region candidates***. These region candidates need further tuning as Grounded-SAM is likely to generate false positives on such loosely worded text prompts like “off-road”. Hence, we train a Road Classifier from the public off-road dataset (I) and use this as a validator for this language-based segmentation model. This finally classifies the pixels within the segmented masks into three different classes, namely ‘road’, ‘semi-road’ and ’background’. A semi-road includes an additional safety area adjacent to the drivable region. It serves as a fallback when the drivable area is unavailable or when extra space is needed for maneuvering.

It is worth noting that the input data for Grounded-SAM is sourced from the publicly available SA-1B dataset [20]. SA-1B consists of 11M images and 1.1B mask annotations. Leveraging this extensive dataset as input for our data generation pipeline, we successfully curated a new dataset consisting of 2.7 million images tailored specifically for off-road environments.

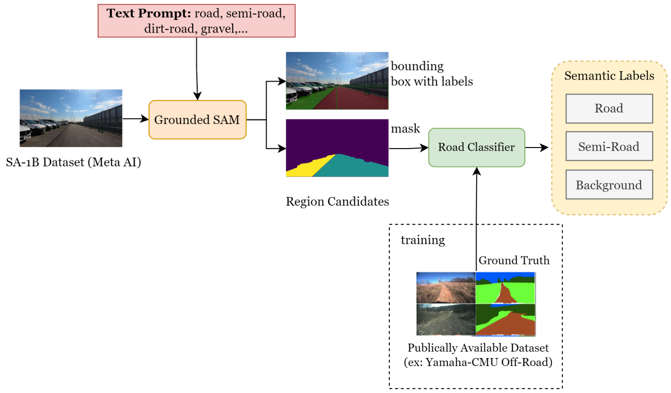


Fig. 2: Dataset Generation Pipeline

IV. Our Pipeline

A. *Image-based Drivable Area Segmentation*

1) Methodology: Vision Transformers (ViT) have emerged as a leading approach for semantic segmentation tasks, achieving state-of-the-art results. Several notable ViT models, including SETR [22], Swin [23], Segmenter [24], SegFormer [25], and Mask2Former [26], have gained significant attention from the research community. However, ViT models often come with a downside of large numbers of parameters, making them bulky and resource-intensive.

SegFormer employs hierarchical encoder layers with a ”light-weight” MLP decoder and introduces a ”positional-encoding-free”. This reduces the overall parameters compared to other models [25] and more robust making it appropriate for safety-critical applications [27].

I. Implementation We used the MiT-B3 pre-trained model as the backbone for retraining the SegFormer. MiT-B3 is trained on ImageNet-1K dataset. The SegFormer model was trained on the entire 2.7M image dataset with a total of 7 epochs and 30 batch size. The total parameters retrained were 64 million.

II. Training We retrained the backbone SegFormer model on the generated dataset (2.7M images) with the following computational resources:

- GPU: NVIDIA RTX 4090 (24GB) x 4
- CPU: Intel i9-10900X
- RAM: 192 GB

2) Class imbalance: The generated dataset has a class imbalance problem with 80% background, 15% road, and 5% semi-road classes. This class imbalance makes predicting the specific region as semi-road class difficult. To address

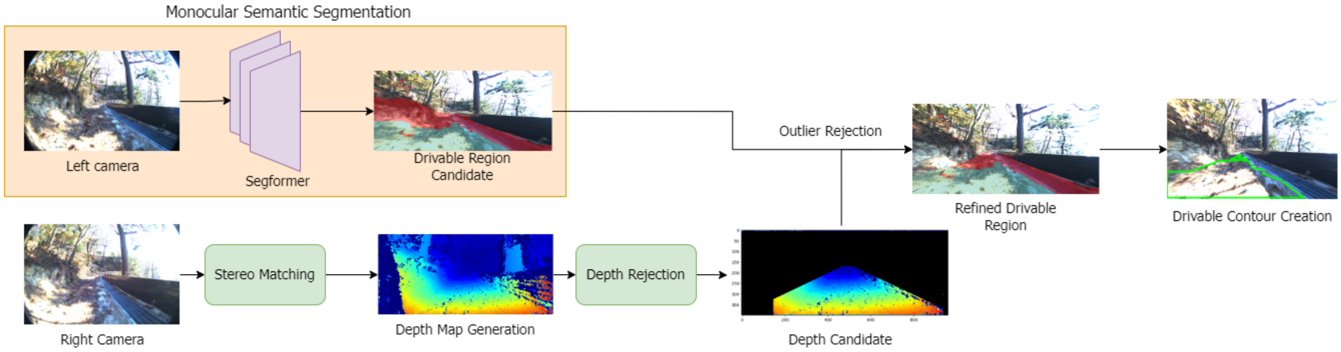


Fig. 3: System Architecture

this issue, the loss function is modified from original cross-entropy to a weighted cross-entropy. The loss function is then defined as follows:

$$loss = \Sigma w_i y_i \log(x_i) \tag{1}$$

$$w_i = \frac{1}{log(1.02 + p_i)}, p_i = \frac{n_i}{\Sigma n_i} \tag{2}$$

Here, n_i is the number of pixels belonging to class i.

B. *Depth Map-based Non-Drivable Area Rejection*

To assess the traversability of a given terrain, it is necessary to gather geometric data such as elevation and slope. This can be achieved by estimating pixel depth, which facilitates 3D reconstruction. Depth estimation methods employing multi-camera stereo setup or an integrated depth sensor can be used to obtain depth information. We reject the non-drivable area based on geometric understanding obtained from the depth information to ensure safe driving for the mobile robot.

The model discerns drivable regions from a semantic perspective so that it may include semantically drivable but physically non-drivable areas such as high slopes or bumps. To consider geometric constraints related to the specification of the UGV, we decided to fuse the depth information from stereo camera systems.

1) Methodology: As a pre-processing step, the image is undistored and cropped before using for depth prediction. Using the final depth map image, we build the corresponding point cloud. Each point in the point cloud has the segmentation result obtained from the SegFormer. The regions of interest are points segmented as a road. The strategy of depth map-based non-drivable area rejection (see Fig.4) we remove non-drivable points in the point cloud for reducing the computational cost.

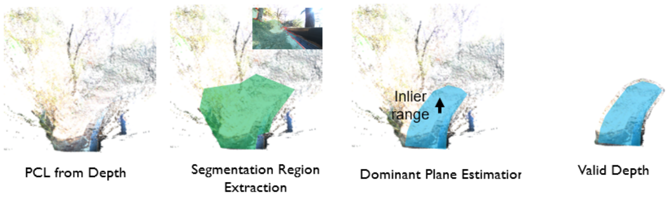


Fig. 4: Depth based rejection pipeline

Using the remaining points, we extract a dominant plane using RANSAC [28]. After estimating the dominant plane, we reject the points that are far more than the height thresholds from the plane. See Algorithm 1 for the pseudo-code of the above algorithm.

Algorithm 1 Non-Drivable Area Rejection

```
Require: Point cloud $P$, distance threshold $d_{\text{threshold}}$, minimum
    points for plane $n_{\text{points}}$, maximum iterations $i_{\max}$, height threshold
    $h_{\text{threshold}}$
Ensure: Filtered point cloud $P_{\text{filtered}}$
1: $P_{\text{drivable}} \leftarrow \text{ExtractDrivablePoint}(P)$
    {Remove non-drivable points}
2: $inliers \leftarrow \text{ExtractDominantPlane}(P_{\text{drivable}}, d_{\text{threshold}}, n_{\text{points}}, i_{\max})$
    {Using RANSAC to extract the dominant plane}
3: $P_{\text{filtered}} \leftarrow \text{RejectPoints}(P_{\text{drivable}}, \text{dominant plane}, h_{\text{threshold}})$
    {Reject points far from the dominant plane}
```

C. *Drivable Contour Extraction*

The motivation to extract drivable contour arises from the fact that they are a simpler and more intuitive representation of the road boundary and can assist the driver with a better visual drivable area in ADAS system. This also facilitates smooth trajectory planning.

Firstly all possible contours are extracted inside the road area in the segmentation map output. A polygon fitting using the Douglas-Peucker algorithm [29] is then performed over all these contours. Inside the polygon, obstacles are identified and using the Convex Hull algorithm [30] a convex polygon is identified. Finally, the boundary of the region and the obstacle is identified by finding exterior most points of the polygon and then converting into Cartesian coordinates. Finally, a spline is fitted through these final polygon points.

Given a set of polygon boundary points $\{b_1, b_2, ..., b_n\}$ obtained after the above processing, a curve $\mathcal{C}$ is fitted using spline interpolation. The spline curve is defined as:

$$\mathcal{C}(u) = \sum_{i=0}^{n} N_i(u) \cdot b_i$$

where $N_i(u)$ are the basis functions, typically cubic B-splines, and u is the parameter ranging from 0 to 1. The spline interpolation ensures a smooth and continuous trajectory that adheres to the drivable region's shape.

Dataset	mIoU (road)				mPrecision	mRecall	mF1
	cnns-fcn	**dark-fcn**	**dark-fcn-448**	**our**	**our**	**our**	**our**
YOCR	42.49	43.79	46.03	**71.25**	95.82	65.11	73.71
	ResNet34+PSP	**ResNet54+PSP**	**ResNet101+PSP**	**our**	**our**	**our**	**our**
CaT	80.12	79.36	**80.57**	76.28	88.92	85.97	85.42

TABLE II: Quantitative Results on seen Dataset

V. Results

We compare our model's mIoU performance on YOCR [14] and CaT [16] dataset reported in Table II. In case of YOCR dataset, we group smooth traversable region and rough traversable region to represent road class and report the performance comparison with the benchmark models as cited in [14]. In case of CaT dataset, three classes (pickup, sedan and off-road) are grouped together to represent the road class and the benchmark models as cited in [31] are used for comparison. Additionally, we report the mean precision, mean recall and mean F1 scores.

VI. Experiments

A. ROS2 Compatibility

We deployed the entire pipeline over ROS2 for testing in real-world. The system has two main nodes: Image Pre-Processing node for image acquisition, depth generation and rectifying image and Drivable Region Recognition node that generates both drivable segmented map and contour. The details of the implementation are explained in Appendix III

B. Testing on Edge Computing device

The NVIDIA Jetson AGX Orin is a new-gen edge computing platform. We deployed the segmentation pipeline on NVIDIA Jetson AGX Orin (Developer Kit). In the following subsections, we present the results of our experiments,

1) Time Analysis: The pipeline is divided into two main nodes: Image Pre-Processing node and Drivable Region Recognition node. Table III summarises the performance of the pipeline on Jetson Orin.

Image Pre-Processing		Drivable Region Recognition	
Sub-process	Time	Sub-process	Time
Image acquisition Rectification Depth estimation	8ms 10ms 32ms	Semantic Segmentation Depth rejection	61ms 28ms
Total Time	**50ms / 20 FPS**	**Total Time**	**89ms / 11 FPS**

TABLE III: FPS Analysis of the pipeline on Jetson Orin

2) Memory Usage: The memory capacity of Jetson orin is 64GB, and its memory architecture is shared memory (CPU and GPU). Hence, we analyzed memory consumption to check the stability of our methods. The image node consumes 825 MB, and the estimation node consumes 4GB; therefore, the total memory consumption is lower than 5GB.

C. Testing in the wild

To evaluate the performance of the end-to-end pipeline we tested it on on-road and off-road scenarios as presented in Table IV. The pipeline runs in real time at 11 FPS on Jetson Orin. The zero shot results on the this real world data show promising results. Thus, we believe this can be very useful in understanding the traversable terrain.

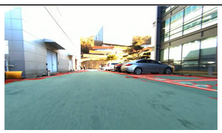
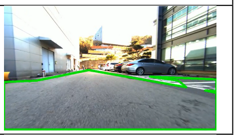
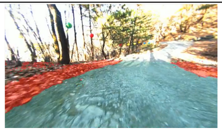
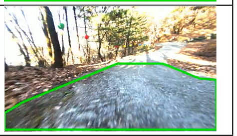
Scenario	Drivable Area	Drivable Contour
on-road		
off-road		

TABLE IV: On-road and Off-road results

The test setup employed to evaluate the model is depicted in Fig.5. A video demonstrating the off-road performance can be found at [Off-Road Video] and the on-road performance can be found at [On-Road Video]. Please refer to Appendix II for detailed results of the study.

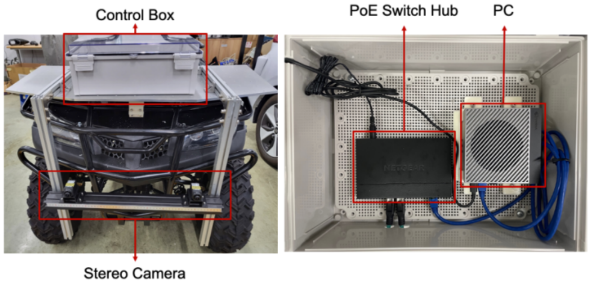


Fig. 5: Test setup

VII. Conclusion

Through our research, we have achieved state-of-the-art real-time performance in embedded systems for robust detection of drivable regions. Our results demonstrate that training the model with the SA-1B dataset has endowed it with the capability to operate effectively across diverse geographical contexts, as evidenced by its performance in real-world scenarios. However, we observed that shadows present challenges across different surface types, which represent an area for future investigation. Based on qualitative results we anticipate higher performance of the pipeline based on the vehicle's traversability. Moving forward, we plan to enhance the model's performance and establish an online learning framework to iteratively improve its functionality, ensuring adaptability to varying weather conditions.

References

[1] F. Islam, M. Nabi, and J. E. Ball, "Off-road detection analysis for autonomous ground vehicles: A review," *Sensors (Basel, Switzerland)*, vol. 22, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253354970

[2] K. Viswanath, P. Jiang, S. PB, and S. Saripalli, "Off-road lidar intensity based semantic segmentation," 2024.

[3] X. Zhou, Y. Feng, X. Li, Z. Zhu, and Y. Hu, "Off-road environment semantic segmentation for autonomous vehicles based on multi-scale feature fusion," *World Electric Vehicle Journal*, vol. 14, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2032-6653/14/10/291

[4] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Proceedings of 11th International Conference on Field and Service Robotics (FSR '17)*, September 2017, pp. 335 – 350.

[5] L. Dabbiru, S. Sharma, C. Goodin, S. Ozier, C. Hudson, D. Carruth, M. Doude, G. Mason, and J. Ball, "Traversability mapping in off-road environment using semantic segmentation," in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2021*, M. C. Dudzik, S. M. Jameson, and T. J. Axenson, Eds., vol. 11748, International Society for Optics and Photonics. SPIE, 2021, p. 117480C. [Online]. Available: https://doi.org/10.1117/12.2587661

[6] Y. Jin, D. K. Han, and H. Ko, "Memory-based semantic segmentation for off-road unstructured natural environments," 2021.

[7] P. Neigel, J. R. Rambach, and D. Stricker, "OFFSED: Off-road semantic segmentation dataset," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP-2021)*. SCITEPRESS, 2021. [Online]. Available: http://www.scitepress.org/PublicationsDetail.aspx?ID=bTLCXMAK4jM=&t=1

[8] I. Sgibnev, A. Sorokin, B. Vishnyakov, and Y. Vizilter, "Deep semantic segmentation for the off-road autonomous driving," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2020, pp. 617–622, 08 2020.

[9] R. Fan, H. Wang, P. Cai, and M. Liu, "SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 340–356.

[10] J. Li, Y. Zhang, P. Yun, G. Zhou, Q. Chen, and R. Fan, "Roadformer: Duplex transformer for rgb-normal semantic road scene parsing," *ArXiv*, vol. abs/2309.10356, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:262053782

[11] Z. Wu, Y. Feng, C. Liu, F. Yu, Q. Chen, and R. Fan, "S3m-net: Joint learning of semantic segmentation and stereo matching for autonomous driving," *ArXiv*, vol. abs/2401.11414, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267069258

[12] H. Wang, Y. Sun, and M. Liu, "Self-supervised drivable area and road anomaly segmentation using rgb-d data for robotic wheelchairs," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4386–4393, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201235453

[13] H. Wang, R. Fan, Y. Sun, and M. Liu, "Applying surface normal information in drivable area and road anomaly detection for ground mobile robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1109/IROS45743.2020.9341340

[14] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[15] S. Sharma, J. Ball, B. Tang, D. Carruth, M. Doude, and M. Islam, "Semantic segmentation with transfer learning for off-road autonomous driving," *Sensors*, vol. 19, no. 11, p. 2577, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2577

[16] S. Sharma, L. Dabbiru, T. Hannis, G. Mason, D. Carruth, M. Doude, C. Goodin, C. Hudson, S. Ozier, J. Ball, and B. Tang, "Cat: Cavs traversability dataset for off-road autonomous driving," *IEEE Access*, vol. 10, pp. 1–1, 01 2022.

[17] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," 2022.

[18] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[19] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[21] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," 2023.

[22] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," 2021.

[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[24] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," 2021.

[25] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021.

[26] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[27] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, "Semantic segmentation using vision transformers: A survey," 2023.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[29] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973. [Online]. Available: https://api.semanticscholar.org/CorpusID:60447873

[30] R. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, no. 1, pp. 18–21, 1973. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019073900203

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2017.

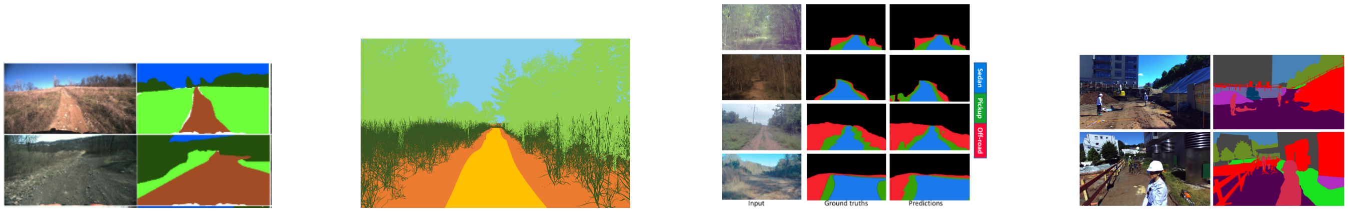
[32] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[33] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018.

Appendix I
Dataset Generation

A. Open-source Dataset Details

The publicly available datasets used in our study to generate the synthetic dataset include the Yamaha-CMU Off-Road dataset [14], CAVS-CaSSeD dataset [15], CAVS-CAT dataset [16], and OFFSED dataset [7]. The Yamaha-CMU Off-Road dataset comprises six class labels: sky, rough trail, smooth trail, traversable grass, high vegetation, non-traversable low vegetation, and obstacles. CAVS-CAT includes four *traversability* classes: sedan, pickup, off-road, and background. CAVS-CASSED consists of simulated data with four classes: sky, trees, vegetation, and ground. The OFFSED dataset contains construction site data and includes five environments: meadows, woods, construction sites, farmland, and paddocks. This enriched the training dataset with a broader range of traversability labels (i.e., off-road labels). On the other hand, RELLIS-3D [17] has 20 class labels, and RUGB [18] has 24 class labels, resulting in a lower percentage of traversability labels. Hence, we only including first 4 dataset in our study for training.



(i) Yamaha-CMU (ii) CaVS-CaSSeD (iii) CaVS-CAT (iv) Offsed

Fig. 6: Samples from Open-Source Off-Road Dataset

B. SA-1B Dataset

SA-1B consists of 11M diverse, high-resolution, privacy protecting images and 1.1B high-quality segmentation masks that were collected with our data engine. It is intended to be used for computer vision research for the purposes permitted under our Data License. The images are licensed from a large photo company. The 1.1B masks were produced using our data engine, all of which were generated fully automatically by the Segment Anything Model (SAM). Please refer to the paper for more details on the mask generation process.

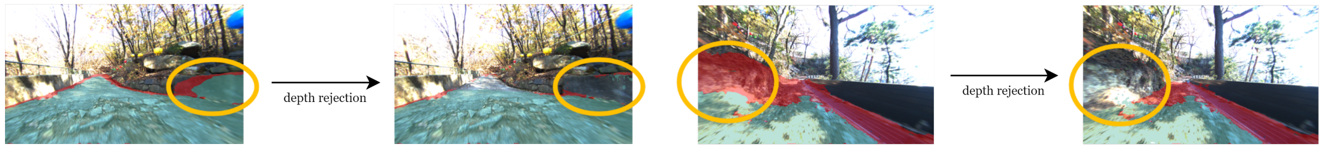
- Total number of images: 11M
- Total number of masks: 1.1B
- Average masks per image: 100
- Average image resolution: 1500 × 2250 pixels

NOTE: There are no class labels for the images or mask annotations.

Appendix II
Results

A. Improvement with depth module

For the current evaluation, we do not provide quantitative comparison of performance with and without depth module since the dataset (train and test) did not have depth information augmented in the ground truth. Figure 7 presents the visual improvement in depth as observed by the model in off-road real-world testing. As can be observed in the figure, our base SegFormer model predicts road and semi-road without considering their elevation. Such areas are not traversable given the robot constraints. Therefore, the depth rejection module removes such outlier areas from the final prediction.



(i) (ii)

Fig. 7: Outlier Rejection with depth module

B. Qualitative Results on Sample Off-Road images

Figure 10 includes various samples of off-road images sourced from different datasets with the performance of **our** model.

C. Performance in Unseen Environment

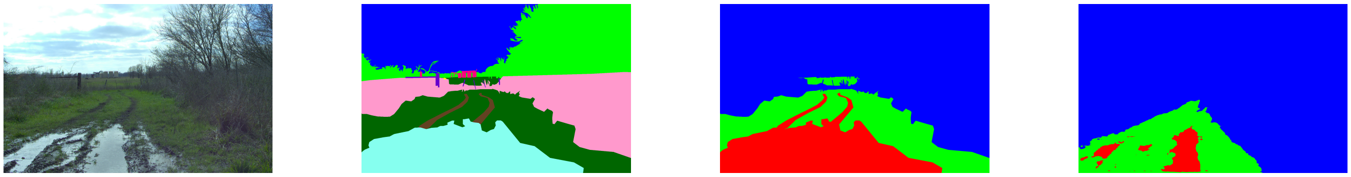
1) Motivation: In the exciting world of robotics, we often encounter situations where our machines need to navigate through new and confidential environments, ex: secret facilities or remote areas where data collection isn't feasible beforehand. In these cases, having a robot that can quickly adapt and perform well without prior training data is essential. We report our models performance on unseen dataset during training and claim that since our model has seen diverse off-road data during training (2.7M+ images), it performs reasonably well in unseen environment.

2) Setup: We test over two famous off-road datasets unseen during training in our pipeline - RELLIS-3D[17] and RUGD [18]. Since, both datasets have 20+ labels in its ground truth we had to transform them into road, semi-road and background for comparing the performance of our model against the benchmark results. For this, we consider [32] as a reference to design our experiment. The classes defined in [32] are presented in Table V along with the modified labels that we use in our experiments.

Hierarchy Level in [32]	**Classes in [32]**	**Re-defined labels**
Navigable Smooth Region	Concrete, Asphalt	road
Navigable Rough Region	Gravel, Grass, Dust, Sand	road
Navigable Bumpy Region	Rock, Rock bed	semi-road
Forbidden	Water, Bushes, Tall Vegetation	background
Obstacles	Trees, Poles, Logs, etc.	background
Background	Void, Sky, Sign	background

TABLE V: Texture based terrain classification as seen in [32]

Figure 8 presents a sample from from RELLIS-3D dataset with original labels, modified labels and output from **our** pipeline.



(i) Original Image

(ii) Original Ground Truth

(iii) Transformed GT

(iv) SegFormer Output

Fig. 8: Comparison from RELLIS-3D Dataset

3) Results: We pick GA-NAV-r8, the best performing model from [32], DeepLabv3+ [33], famous encoder-decoder based semantic segmentation model and Segmenter [24], another Transformer based semantic segmentation model. We report mean IoU scores, mean Precision, mean Recall and mean F1 scores of our model and the benchmarks on both the datasets in Table VI. Note: RELLIS-3D and RUGD were not used in training the model.

Dataset	**mIoU (road)**				**mPrecision**	**mRecall**	**mF1**
	DeepLabv3+	**Segmenter**	**GA-Nav-r8**	**our**	**our**	**our**	**our**
RUGD	45.425	90.78	93.21	48.24	59.19	55.94	55.71
RELLIS-3D	72.78	65.03	83.375	53.95	98.06	54.87	64.36

TABLE VI: Quantitative Results on Unseen Dataset

Appendix III
ROS2 Compatiblity

The Robot Operating System 2 (ROS2), is an open-source software framework primarily designed for building robotic systems, with applications like Autonomous Driving (AD) and Advanced Driver-Assistance Systems (ADAS). We leverage the compatibility of our pipeline with ROS2, enabling seamless integration into existing robotic systems. Specifically, we deployed the segmentation pipeline with in ROS2 node. The ROS2 node subscribes to ROS2 topics including left_camera_image, left_camera_info, disparity_image topics. The disparity_image topic is outputed by the Image Processing node which takes both left and right camera images as input to calculate the disparity. Utilizing these input topics, our model pipeline processes all these inputs and computes the segmented map using our proposed model, which is then published over another ROS topic. The functionality is explained in the Fig 9.

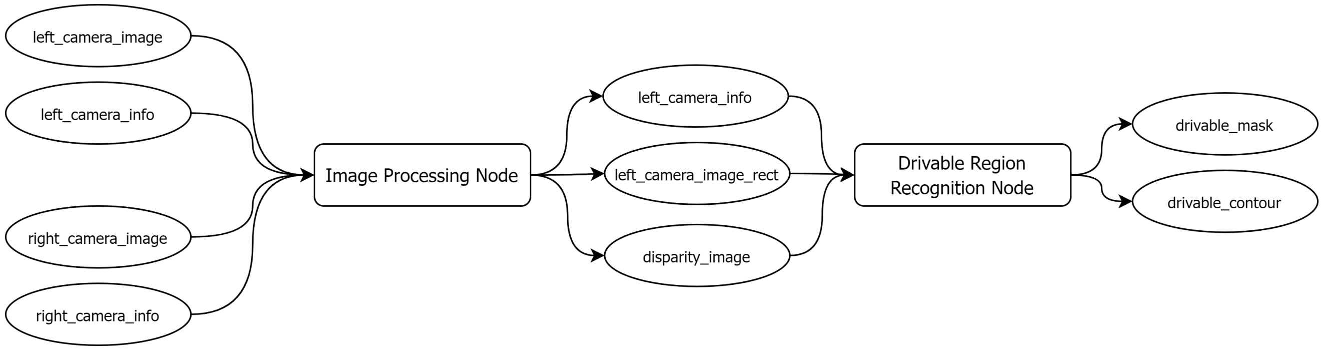
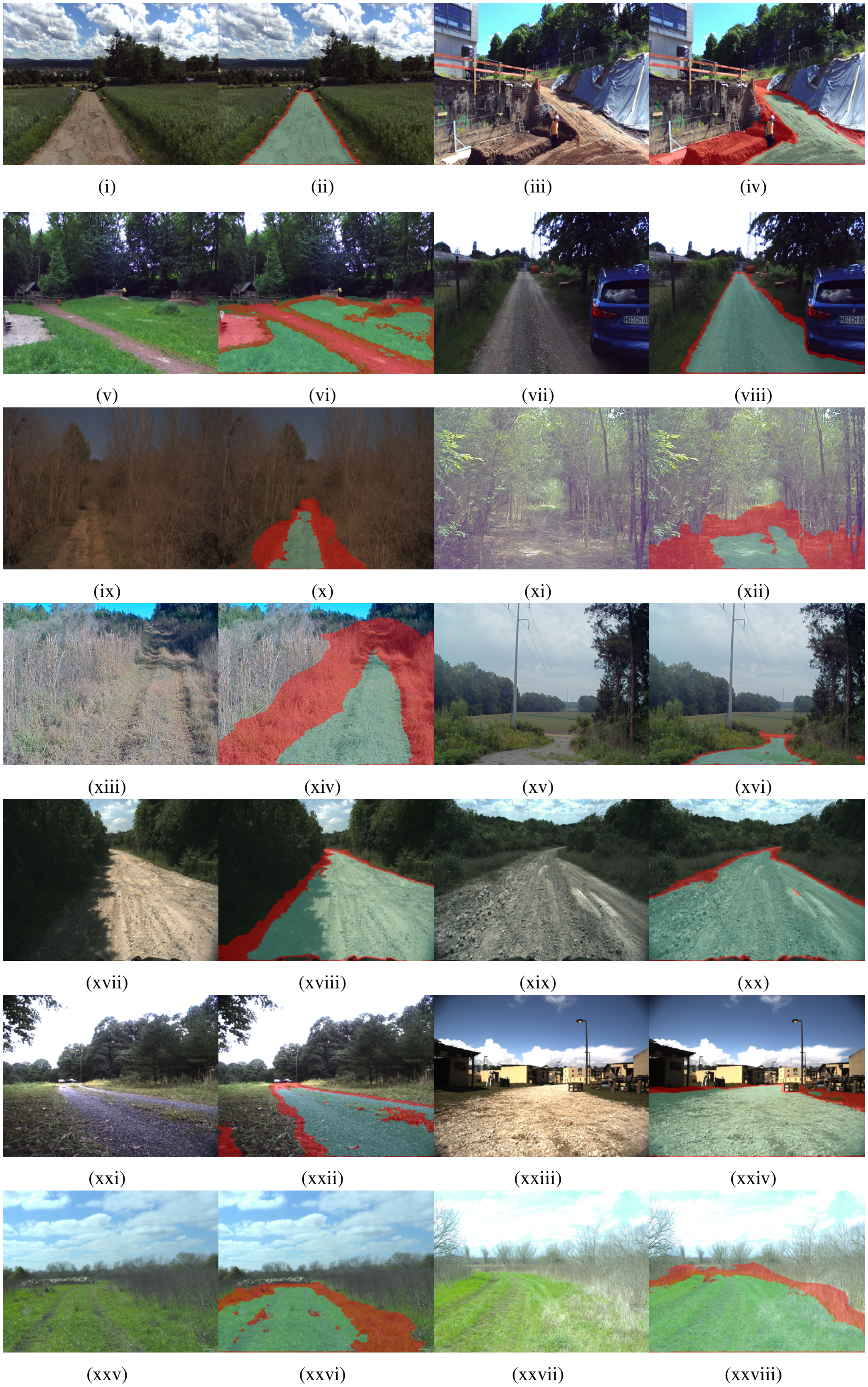


Fig. 9: ROS2 Framework



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

(ix) (x) (xi) (xii)

(xiii) (xiv) (xv) (xvi)

(xvii) (xviii) (xix) (xx)

(xxi) (xxii) (xxiii) (xxiv)

(xxv) (xxvi) (xxvii) (xxviii)

Fig. 10: Qualitative results on various off-road datasets. A mask obtained by **our** model is added to various samples from different datasets. (i) - (viii): OFFSED dataset [7]. (ix) - (xvi): CaVS-CaT dataset [16]. (xvii) - (xx): Yamaha-CMU dataset [14]. (xxi) - (xxiv): RUGD dataset [18] and (xxv) - (xxviii): RELLIS-3D dataset [17]

.