Terrain Traversability Analysis with Image DeHazing

Ritarka Samanta
Electrical and Computer Engineering
rsamanta@andrew.cmu.edu

Woojin Kim
Electrical and Computer Engineering
woojink@andrew.cmu.edu

Nikhil Sobanbabu
Electrical and Computer Engineering
nsobanba@andrew.cmu.edu

1 Introduction

Autonomous navigation in off-road terrains is advancing rapidly, with applications across environmental exploration (e.g., space and mining), disaster response, agriculture, and defense. Robots in these unstructured environments encounter unpredictable terrain—from loose soil and dense vegetation to rocky surfaces and water bodies. Identifying traversable regions is essential for path planning and safety, allowing robots to avoid hazardous areas and adapt to dynamic conditions. Effective traversability analysis thus plays a vital role in mission success, but is complicated by inconsistent terrain and lighting conditions, which can obscure features crucial for navigation. Prior works have leveraged computer vision and deep learning to analyze traversability, utilizing color, texture, and depth data from RGB cameras and LiDAR sensors[1][16][10].

Existing segmentation models, however, require high-quality images to perform reliably. In real-world off-road settings, environmental conditions like dust, fog, or glare often degrade image quality, creating blurry or noisy visuals. For instance, a robot in a forest might encounter fog that reduces clarity, or a rocky terrain in direct sunlight might produce glare, complicating the identification of safe paths. Traditional CNN-based models, which rely heavily on local image features, are particularly vulnerable to degraded inputs, leading to misclassifications that compromise navigational safety. Although multi-sensor fusion techniques have been introduced to improve robustness, they remain complex and struggle in extreme conditions[6].

Diffusion models are known for detailed dehazing but are computationally intensive, requiring multiple processing steps. This limits their effectiveness for real-time applications, especially on mobile robots with restricted processing power.

Our work analyzes Dark Channel Prior (DCP), Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN) methods as the preprocessing/dehazing step improving the clarity of inputs for the GA-Nav framework [5], a robust segmentation model optimized for unstructured terrains. By combining dehazing method, we aim to enhance traversability analysis even under challenging visual conditions.

This work presents a novel pipeline integrating dehazing with terrain segmentation, contributing to safer, more reliable off-road navigation in visually degraded environments.

2 Related Work

Much work has been done in the realm of image-to-image processing. MSCNN [11] and DehazeNet [2] are two CNN based models that tackled the de-hazing problem. Notable, there have been a few models that have used Generative Adversarial Networks (GANs) to denoise images [14] or remove rain from incoming images [19]. De-raining, the earlier paper, focused primarily on removing rain streaks but as a consequence of its compound loss function, was able to help remove mist and haze as well. We also found out that diffusion models tend to be quite slow and subsequently infeasible for potential real-time processing. Using these as references, we believe CNN and GAN's capability of dehazing incoming images.

Previous research on terrain classification has explored various techniques using camera images and LiDAR data to support navigation in both on- and off-road environments. Many of these appearance-based classification methods leverage features like color, texture, and depth to differentiate terrain types, providing valuable input for traversability analysis. For instance, a study in [12] developed a classification method by normalizing RGB images to reduce gamma correction effects, thus improving contrast and color space clarity. This process, followed by an MLP classifier, achieved a notable 93% accuracy, with particularly strong performance in off-road environments, where dense visual information enhanced terrain identification.

Another approach in [9] focused on monocular camera-based off-road detection, where traversability was assessed using a road-type inference algorithm. By combining road model estimation with region predictions, this method enabled more reliable predictions in varying outdoor conditions. A similar effort to address off-road challenges was proposed by [3] with the Virtual Autonomous Navigation Environment (VANE). This study utilized decision tree classifiers for material classification and a segmentation labeling toolbox, demonstrating the importance of structured datasets for effective terrain segmentation in simulated and real environments.

A multi-modal approach to semantic segmentation was introduced in [15], utilizing the AdapNet++ architecture. This study integrated multiple datasets, including Cityscapes and Freiburg Forest, and implemented data augmentation to enhance segmentation across varied terrains. Their findings significantly improved classification accuracy and processing speed, validating the model on off-road terrains with diverse environmental conditions.

These appearance-based approaches underscore the advancements in terrain segmentation but also highlight the dependency on high-quality images and complex processing architectures, which can be challenged by degraded visual inputs in real-time settings. Unlike these methods, our work incorporates dehazing to pre-process images, enhancing feature clarity and enabling more accurate terrain classification even under adverse conditions. This approach offers a streamlined, single-pass solution that strengthens GA-Nav segmentation by providing robust, preprocessed images, thus improving traversability analysis in visually challenging outdoor environments.

3 Methods

3.1 Acquiring Training Data

It is difficult to find images of foggy roads and offroad terrain. Not only do people generally not take images of offroad terrain, foggy images (where the subject matter is not seen easily) is even more rare, leading to a rarity of those types of images on the internet. Pervious works have attempted to bypass this limitation by manually adding rain and fog through photoshop [19], but we attempted a more algorithmic approach to make this more scalable.

We have used a haze synthesis model, color jitter, and a Gaussian filter to create hazy images with which we will train the GAN. The intensity of the haziness and blurriness were randomly set for each image. Since rain and noise rarely tends to obfuscate features, we were worried about potential hallucinations when attempting to de-haze images. In other words, our GAN should try its best to infer the incoming terrain, but should also have the ability to gracefully fail and say it does not know.

There are many ways to solve this problem, but one approach we investigated is by separating the image into patches, calculating their individual entropies (as a proxy of the complexity of the image) and gracefully failing if the entropies are too low (i.e. too hazy/dark). This particular threshold can be calculated analytically by finding the distribution of entropies in the blurry dataset.

Regardless, our model is trained on the normal and blurry images of the RUGD dataset and follows a similar model architecture as the de-raining model. Similar to ID-CGAN [19], we will be using the three loss functions (pixel wise entropy, adversarial, and perceptual loss). Unlike ID-CGAN though, we will place less emphasis on adversarial loss and more so on perceptual loss). Since the generated image will be fed to a segmentation model, there is a greater need to preserve high-level features (such as important road marking), and less need to make it seem realistic.

We followed the generator and discriminator architecture of ID-CGAN to begin with. It is relatively small, a few densely connected blocks with skip connections for the generator, and a few convolutions as the discriminator. This lightweight architecture took the authors only 0.3 seconds to process an image on an NVIDIA Titan-X. This time will only be faster for modern GPUs. Judging by the results of the paper, a lightweight model was more than sufficient.

3.2 Dehazing

3.2.1 Haze Removal using dark channel prior

Hazy images can significantly affect the performance of computer vision systems, particularly in terrain traversability analysis for off-road environments. To address this, we incorporated the Dark Channel Prior (DCP)-based dehazing algorithm proposed by He, Sun, and Tang [7] into our preprocessing pipeline. This method exploits the statistical properties of haze-free natural images to estimate the scene radiance and remove haze. The dehazed images are subsequently utilized in terrain segmentation and traversability analysis using the GANAv framework. The presence of haze is modeled using the following image formation equation:

$$I(x) = J(x)t(x) + A(1 - t(x)), \tag{1}$$

where:

- $I(x)$: Observed intensity of the hazy image at pixel x,
- $J(x)$: Scene radiance or true intensity of the object in the absence of haze,
- $t(x)$: Medium transmission map, indicating the fraction of light reaching the camera without scattering,
- A: Global atmospheric light, representing the ambient light scattered in the environment,
- $(1 - t(x))$: Proportion of light scattered by haze particles.

The goal of image dehazing is to recover $J(x)$ from $I(x)$ using an accurate estimation of $t(x)$ and A. The Dark Channel Prior is based on the statistical observation that in most haze-free natural images, at least one color channel has very low intensity in non-sky regions. For a local patch $\Omega(x)$ centered at pixel x, the dark channel is computed as:

$$J_{dark}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} J^c(y) \right), \tag{2}$$

where $J^c(y)$ denotes the intensity of the color channel c at pixel y. In haze-free images, $J_{dark}(x)$ is close to zero for most non-sky regions due to shadows, color diversity, and strong textures. In hazy images, the presence of haze increases the intensity of the dark channel. This discrepancy is used to estimate the medium transmission map. The global atmospheric light A is estimated by identifying the brightest pixels in the dark channel map, as these regions are likely to be haze-dominated (e.g., sky or distant objects). We follow a heuristic approach where a small percentage (e.g., top 0.1%) of pixels with the highest intensity in the dark channel are considered. The pixel with the highest intensity among these is selected as A. The medium transmission map $t(x)$ is approximated using the Dark Channel Prior:

$$t(x) = 1 - \omega \cdot \min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c} \right), \tag{3}$$

where $\omega \in [0, 1]$ is a weight factor to retain a small amount of haze, preventing oversaturation in the final dehazed image. A typical value for ω is 0.95. The raw transmission map obtained from Equation 3 may be noisy and contain artifacts. To address this, a soft matting technique is employed to refine $t(x)$ by incorporating spatial smoothness and preserving edge details. For computational efficiency, we alternatively use a guided filter.The haze-free image $J(x)$ is recovered by inverting the haze model (Equation 1):

$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A, \tag{4}$$

where t_0 is a lower bound for the transmission map to avoid division by zero. This ensures numerical stability and prevents artifacts in the dehazed image The dehazed images obtained using DCP are passed to the GANAv framework for terrain segmentation. By removing haze, the DCP technique enhances the visibility of terrain features, improving the accuracy of traversability analysis in challenging environments.

3.2.2 Haze Removal using CNN based architecture

Dehazing objective can be represented with the following equation: $J(x) = \frac{1}{t(x)} I(x) - A\frac{1}{t(x)} + A$ [8] where $J(x)$ is the clean image that we want to obtain, $I(x)$ is our hazy image, A is the global atmospheric light constant and transmission $t(x)$, the amount of light reaching without absorption. $t(x)$ can be further expanded to $t(x) = e^{-\beta d(x)}$ where β is the scattering coefficient and $d(x)$ is the object and camera distance. By combining $t(x)$ and A together, $J(x) = \frac{1}{t(x)} I(x) - A\frac{1}{t(x)} + A$ can be rewritten as $J(x) = K(x)I(x) - K(x) + b$ where $K(x) = \frac{\frac{1}{t(x)}(I(x)-A)+(A-b)}{I(x)-1}$, showing the relative depth and haziness in the image.

Previously, traditional methods such as dark channel prior (DCP), estimated t(x) using the information gained from the image, and constant A was also often estimated through a statistical approach, which were time-consuming and often made errors from wrong assumptions. However, convolutional neural network (CNN) based architecture, AOD-Net (All-in-One Dehazing Network) [8] introduces a deep learning method that uses CNN architecture to directly estimate K(x).

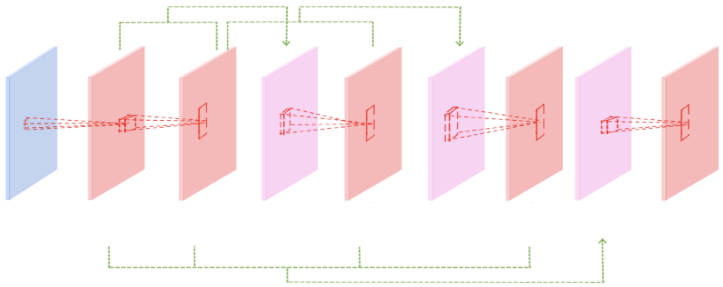


Figure 1: AOD-Net architecture

The architecture consists of five 2D convolution layers, which extract low- and high-level features, concatenations to combine and preserve the intermediate outputs, and ReLU layers that help to estimate the complexities in estimating the term $K(x)$.

However, the complex nature of the haze in our dataset could not be fully resolved using the baseline architecture. In the training with the baseline architecture, we observed that loss has stuck and stopped going down at certain point, thus we tried to increase the learning rate and implement scheduler (step and CosineAnnealing) to further decrease the loss. We also increased the batch size to reduce training time and solve the limitation of our resources. In the next step of the experiment, we incorporated the Laplacian loss into the original mean square loss. This adjustment was made as the segmentation objective in our terrain traversability work focuses on segmenting out large sections, rather than finer segmentation objectives such as identifying rocks, trees or birds. We believed that adding Laplacian loss into the total loss would help to better identify the boundaries between these large sections. Lastly, we added a residual layer that adds the input to the last output. This was to prevent any potential vanishing gradient problem and preserve the data better. The results of these studies will be explained further in the result section below.

3.2.3 Haze Removal using GAN based architecture

Motivated by the promising results of GANs in removing rain [19] and noise [14], we experimented with training a GAN (with the same architecture as ID-CGAN) to try and dehaze the images.

While the authors of ID-CGAN used photoshop to create rainy images, as mentioned before we used blurs, color jitters, and other ML models to create the hazy images. Additionally, it is to be noted that Zhang et. al primarily used colorful pictures of urban landscapes, while we used images from the RUGD dataset which primarily contains off-road and natural scenery. It was expected that due to the stronger haze, blurrier edges, and less contrastive colors of our hazy dataset, that our trained GAN would not possess results as remarkable as the ones in ID-CGAN.

For the training, we first resized every image to a 256x256, added some color jitter, and normalized it to 0.5 mean and 0.5 standard deviation across all dimensions. We down-scaled all of these images primarily because we did not care immensely about the fidelity of the low-grained details. The higher level features such as the trail locations, trail bumpiness, trees, etc. are captured more easily as a consequence. The generator and discriminator were both trained with the Adam optimizer. The generator in particular had a learning rate of 0.002, 10x larger than the learning rate of the discriminator (0.0002). This disparity was determined through empirical evaluation upon realizing the discriminator was typically able to quickly identify fakes and learn fakes far faster than the generator. Everything was trained with a batch size of 16.

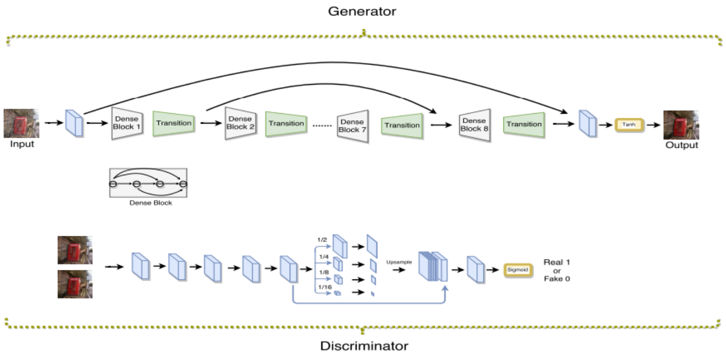


Figure 2: ID-CGAN architecture

Some light experimentation was done with loss functions and was eventually settled on a 1 to 6 ratio of adversarial loss and reconstruction loss (based on l2 norm). Had we had more time, this would have been the most critical aspect of further experimentation. From related works, it has been realized that the performance of the Generator is tied closely to some linear combination of 2 or 3 different types of losses. The largest barrier to this sort of experimentation is the number of training loops that would be required.

3.3 Segmentation

After obtaining de-hazed image, next part will be a coarse-grained segmentation task to classify image regions into where the vehicle should avoid or prioritize if there are multiple valid paths for navigation.

There are two important things to be considered for segmentation part in the navigation task, the segmentation (including class classification) accuracy and computation efficiency. During traversing, wrong segmentation could bring detrimental damage or unexpected situation to the vehicle which could possibly lead to vehicle malfunctioning. On top of that, fast but not much degraded (in terms of accuracy) computation is required to match up the speed of path decision with the vehicle's speed so that the vehicle can continuously move without stop and hesitate.

GA-Nav [5] suggests a outdoor environment segmentation model that improves on these two important details.

In regards to segmentation accuracy, it would be harder for the segmentation model to be precise if there are many classes to predict. In terms of navigation, it's not necessary to classify whether the obstacle in front is a fence or tree since they are both obstacles limiting the vehicle's path. Thus, it would be beneficial to the model's accuracy if it simply predicts 'yes' or 'no' regions instead. Additionally, separating 'yes' regions into a little more detailed categories, such as 'smooth', 'bumpy', 'rough' regions would be also beneficial for the vehicle as it can put more weights in certain path if there are multiple path options to decide from and if it has or requires different settings such as suspension, lowering/raising its body for different types of terrain. GA-Nav, compared to traditional segmentation models (which each object has finer classification labels such as 'tree', 'water', 'rock', 'mud', etc.), it implements coarse-grained segmentation method and segments images into 'smooth', 'bumpy', 'rough', 'forbidden', 'obstacle', 'background' labeled regions. GA-Nav simply merges the RUGD dataset class labels into these 6 new categories by relabeling the notations and re-run the training.

GA-Nav's Mixed Transformer (Segformer [18]) based backbone with additional Multi-Head Self Attention (MHSA) network mechanism provides an approach for segmentation that's not only fast-efficiently computed but still maintaining decent level of accuracy.

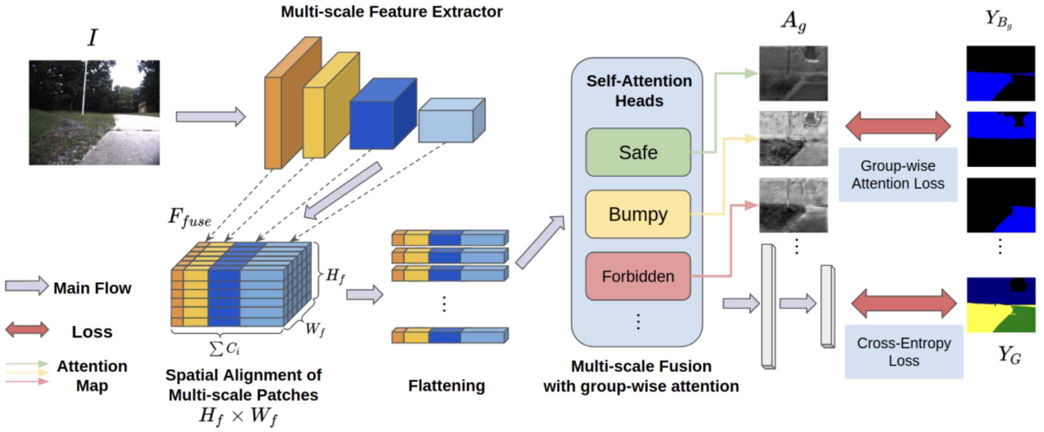


Figure 3: GA-Nav architecture

In the beginning of the network, the transformer encoders generate multi-scaled feature maps with the input of overlapping patches (size 7 by 7 with a stride of 4). These outputs have different dimensions, the smaller height and width are chosen and bilinear interpolation has been used to align the dimensionality (Spatial Alignment). The computational complexity has reduced from this dimensionality reductions but the decrease in performance would also be expected at the same time. But, the network recovers this degradation by introducing group-wise self-attention. MHSA will output new attention map A_{out} for each group, which is calculated by the following equation: $A_{out} = softmax\left(k(A_{in})^T \cdot q(A_{in})^T\right) \cdot v(A_{in})$. Next, using A_{out} and going through multiple 1x1 convolution blocks and up-sampling, the final probability map for each group, P_g, can be obtained which will be translated to the final segmentation feature map.

During training, group-wise attention loss is applied using the following binary cross-entropy loss, $\mathcal{L}^g_{GA} = -\sum_{h,w} y_G \log(B_g)$ (where B_g is self-attention score acquired from the attention map for each group) for the purpose to adjust each head so that it can better focus on region that it needs to be attended.

Lastly, traditional cross-entropy loss is used to update and optimize the model $\mathcal{L}_{CE} = -\sum_{h,w} \sum_{g \in G} y_{GT} \log(P_g)$ (where y_{GT} is ground-truth label)

The following evaluation metrics will be used to calculate the performance of the segmentation. Intersection over Union (IoU) for class i: $IoU_i = \frac{\sum_I \sum_{x,y} \mathbf{1}(P(x,y)=i \, and \, G(x,y)=i)}{\sum_I \sum_{x,y} \mathbf{1}(P(x,y)=i \, or \, G(x,y)=i)}$

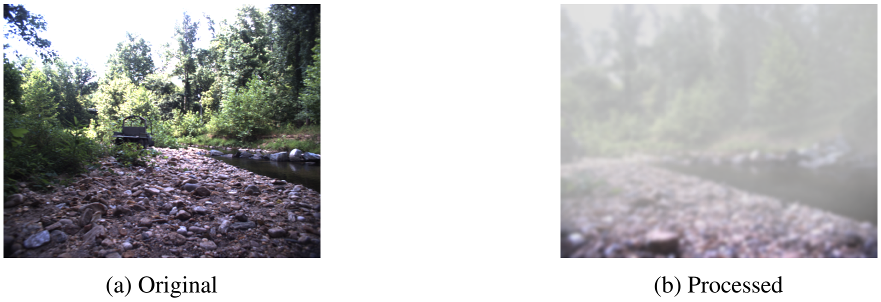
Mean IoU (mIoU): $mIoU = \frac{\sum_i mIoU_i}{\sum_B 1}$

Mean Pixel Accuracy (mAcc): $mAcc = \frac{\sum_{i \in B}\left(\sum_I \sum_{x,y,G(x,y)=i} 1(P(x,y)=G(x,y))\right)}{\sum_B 1}$

4 Results

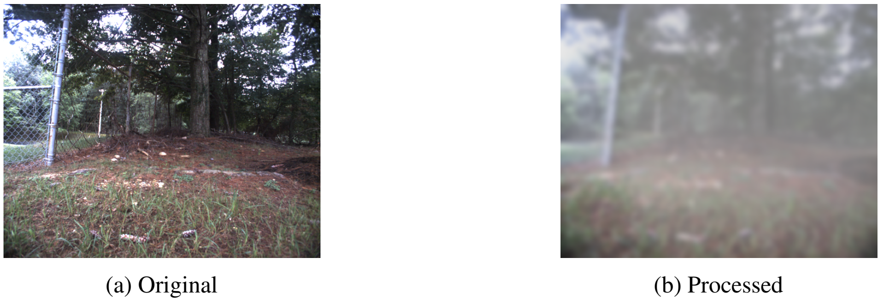
4.1 Acquiring Hazy Data

By using a haze synthesis model [13] on the RUGD dataset [17] and applying a gaussian filter, we were able to produce a dataset with hazy and blurry images.


(a) Original

(b) Processed

Figure 4: Comparison of Original and Processed Images


(a) Original

(b) Processed

Figure 5: Comparison of Original and Processed Images

4.2 Dehazing Results

4.2.1 Results from dark channel prior method

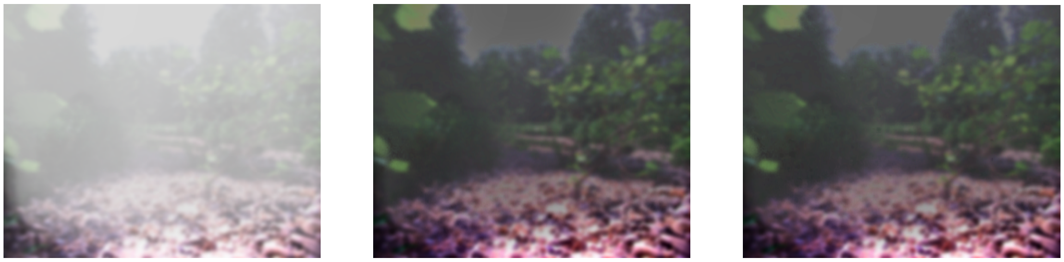
To evaluate the impact of key parameters in the DCP-based dehazing process, we conducted experiments on the following:

- **Patch Size ($\Omega(x)$):** Different patch sizes were tested to analyze their effect on the quality of the dark channel and the transmission map. Larger patch sizes smooth out noise but blur fine details, whereas smaller patch sizes preserve details but amplify noise. This artifact can be clearly seen in Fig 6 where the quality of dehazing is significantly reduced with a high patch size.
- **Refinement Techniques:** We compared the performance of transmission map refinement using soft matting and guided filtering. Although soft matting is shown to provide more accurate results than methods like guided filtering, we noticed that for the terrain segmentation task, the output produced is not significantly different. However, the inference time of soft matting is **9.96x** times than that of guided filtering,Fig7. This is owed to the computationally efficieny of guided filtering and is better suited for real-time applications.

(a) Patch Size: $15x15$

(b) Patch Size: $75x75$

Figure 6: Comparison of dehazing with patch size



(a) Original Hazed Image

(b) Guided-Filter(Inference time 0.51 sec)

(c) Soft-Matting (Inference time 5.09 sec)

Figure 7: Comparison of t-map refinement

4.2.2 Results from CNN based method

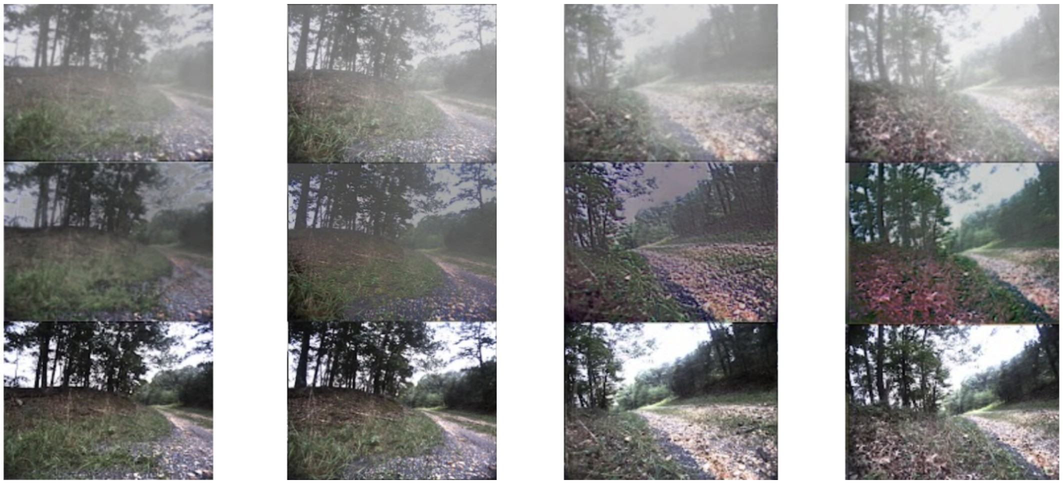


Figure 8: De-hazing results

This is an example results from each modifications that the test images have similar environment and haziness. The top row is the input test hazy image, the bottom row is the original clean image and the middle row is the de-hazed image.

The left most column is the result from the baseline, and the haziness still remains and the brightness went down, also there are some artifacts observed in the sky. The second column is where learning rate was increased with the implementation of the step scheduler and batch size was increased. The artifacts in the sky is gone but the image is still quite dim, but it has higher definition to the baseline result. In the third column, Laplacian loss is merged into the mean square loss. The results shows emphasized definition and color with sharper edges which would help for the segmentation task. In the fourth column, the residual layer is added and shows the best result out of all baseline and modifications.

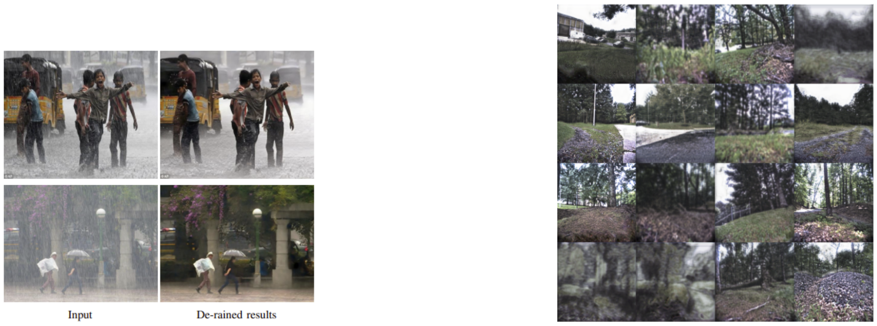
4.2.3 Results from GAN based method

It is clear that ID-CGAN performs excellently in removing the rain and even mist from images, and as such, using the same architecture and their novel loss functions seemed to be a natural step. Although the architecture was preserved, it will be shown that there was some experimentation with the loss function.

As can been seen from the image below, the GAN does particularly well for images with many contrastive shapes. If it does fail (see the two sub-images to the bottom right), does not generate any new images but instead fails elegantly. This was a concern while training the GAN. Generating new and fake landscapes that a rover would perform terrain traversal on would be quite worrying.

However in this case it seems like for the particularly blurry and hazy images, no new information is created.

The GAN also performed inference very quickly on RTX 4090s, taking only about 0.06 seconds per image. We do not believe terrain traversability and dehazing should be run multiple times a second unless the landscape changes dramatically. This suggests that the dehazing GAN could be made more deeper and more complex as potential next steps.



(a) ID-CGAN example output by Zhang et. al

(b) Fine-tuned ID-CGAN on hazy RUGD

Figure 9: ID-CGAN Results

4.3 Segmentation Results

We have experimented with the GANav framework, adopting the code base from [4] and have trained the model using a 6 group label, The RUGD data was loaded and relabelled with 6 group course segmentation label and trained using the GANav frame work. Further, we have tested the framework using the original RUGD dataset, our hazy dataset and the outputs of the three dehazing techniques keeping the same test split as the original GANav framework. We report the mean-IoU(**m-Iou**) for the various methods in Fig 10 comparison for the same methods.

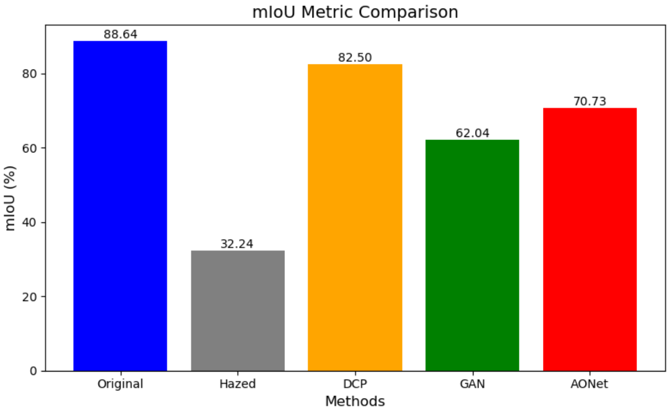


Figure 10: mIoU Comparison

As it can be clearly seen that, even though the trained model has higher performance metrics on the original test dataset, when the input image is blurry or hazy, the performance drastically drops, which can be seen from the **mIoU** metric dropping from 88.64 to 32.24. However, with the different dehazing methods that we tried we can see that the performance improves during with mIoU reaching 82.50 for DCP method, which is almost comparable to the original dataset. We see that even for GAN-based and CNN-based methods there is a performance improvement. Although the AONet(CNN) based method provides lesser mIoU compared to the DCP method, we observe that the inference time

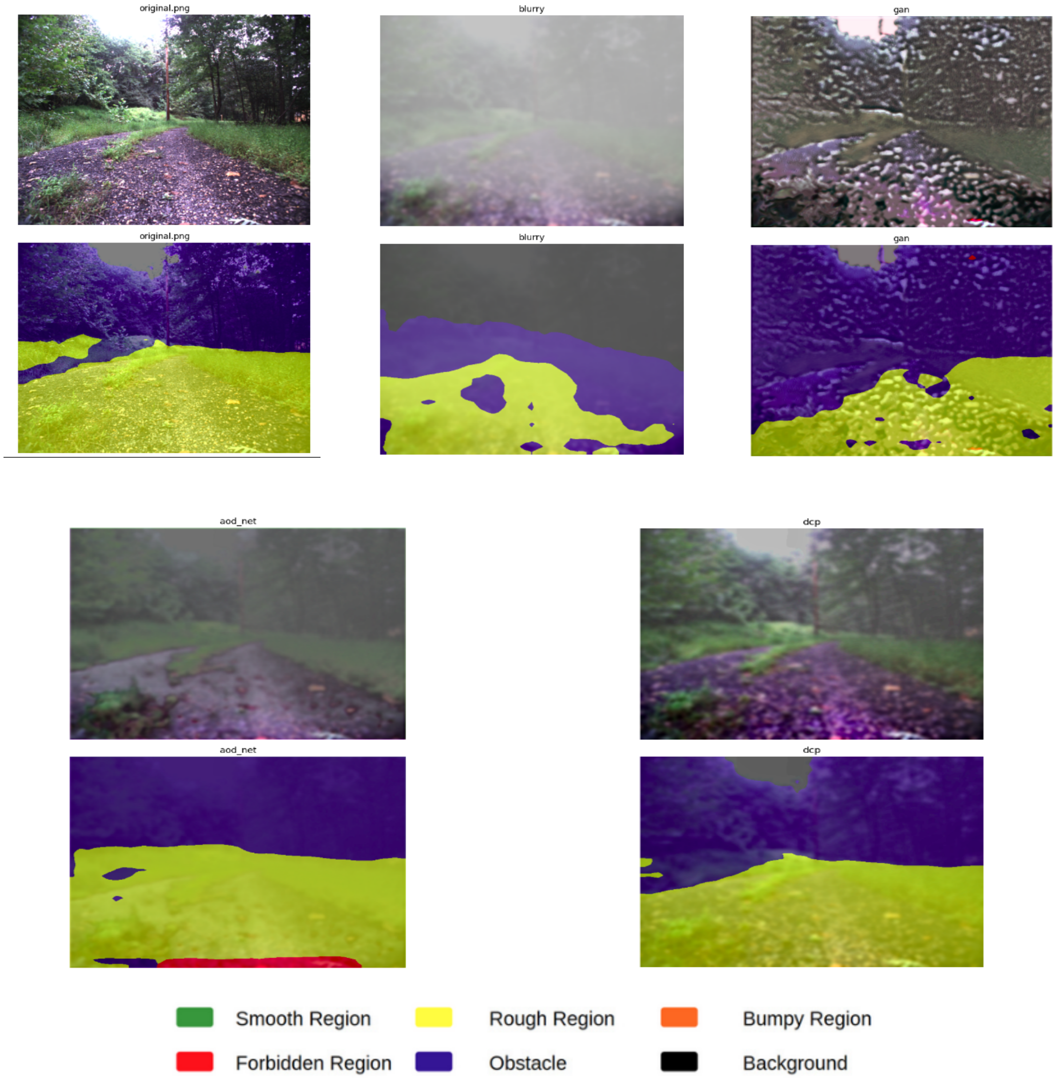


Figure 11: Terrain segmentation of a single sample through multiple dehazing methods

drastically increases for DCP with the image size. Hence, we observe that AONet provides the best of both worlds and can be used in real-time applications.

We also visualize the segmentation masks produced by the model on a single sample image in Fig 11

5 Discussion and Conclusion

In this work, we demonstrated the importance of image dehazing as a preprocessing step for improving terrain traversability analysis in visually degraded environments using the GA-Nav framework. Among the methods evaluated, DCP achieved the highest mIoU on hazy datasets, nearly matching the original dataset performance. However, its high computational cost limits real-time applicability, where the AONet method offers a practical balance between accuracy and efficiency. These results highlight the significant impact of degraded inputs on segmentation performance and the effectiveness of preprocessing in mitigating these effects. While our approach enhances robustness in challenging conditions, future work could explore adaptive or hybrid preprocessing methods and hardware-optimized implementations to improve scalability for diverse terrains and environmental conditions.

References

[1] Jhonghyun An. Traversable region detection and tracking for a sparse 3d laser scanner for off-road environments using range images. *Sensors*, 23(13), 2023.

[2] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal, 2016.

[3] David R. Chambers, Jason Gassaway, Christopher Goodin, and Phillip J. Durst. Simulation of a multispectral, multicamera, off-road autonomous vehicle perception system with Virtual Autonomous Navigation Environment (VANE). In David A. Huckridge, Reinhard Ebert, Mark T. Gruneisen, Miloslav Dusek, and John G. Rarity, editors, *Electro-Optical and Infrared Systems: Technology and Applications XII; and Quantum Information Science and Technology*, volume 9648, page 964802. International Society for Optics and Photonics, SPIE, 2015.

[4] Ray Guan et al. GANav-offroad: Gps-denied all-terrain navigation with visual-inertial odometry. `https://github.com/rayguan97/GANav-offroad`, 2023.

[5] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. Ganav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments, 2022.

[6] Garen Haddeler, Meng Yee (Michael) Chuah, Yangwei You, Jianle Chan, Albertus H. Adiwahono, Wei Yun Yau, and Chee-Meng Chew. Traversability analysis with vision and terrain probing for safe legged robot navigation. *Frontiers in Robotics and AI*, 9, August 2022.

[7] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2010.

[8] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network, 2017.

[9] Jilin Mei, Yufeng Yu, Huijing Zhao, and Hongbin Zha. Scene-adaptive off-road detection using a monocular camera. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):242–253, 2018.

[10] Giulio Polato, Sebastiano Chiodini, Andrea Valmorbida, Marco Pertile, Giada Giorgi, and Enrico C. Lorenzini. Semantic terrain traversability analysis based on deep learning aimed at planetary rover navigation. *Aerotecnica Missili & Spazio*, Oct 2024.

[11] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks, 2016.

[12] T Selvathai, Jayashree Varadhan, and Swarna Ramesh. Road and off road terrain classification for autonomous ground vehicle. In *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–3. IEEE, 2017.

[13] Le-Anh Tran, Chung Nguyen Tran, Dong-Chul Park, Jordi Carrabina, and David Castells-Rufas. Toward improving robustness of object detectors against domain shift. In *2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pages 01–05. IEEE, 2024.

[14] Subarna Tripathi, Zachary C. Lipton, and Truong Q. Nguyen. Correction by projection: Denoising images with generative adversarial networks, 2018.

[15] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *CoRR*, abs/1808.03833, 2018.

[16] Nan Wang, Xiang Li, Zhe Suo, Jiuchen Fan, Jixin Wang, and Dongxuan Xie. Traversability analysis and path planning for autonomous wheeled vehicles on rigid terrains. *Drones*, 8(9), 2024.

[17] 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.

[18] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021.

[19] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network, 2019.