Semantic Terrain Segmentation

A PROJECT REPORT

submitted by

BASU HELA(2204101013)
HARSH VERMA(2204101021)
HIMANSHU CHHABRA(2204101023)
SOUMYA ASATI(2204101046)
VAISHALI CHAUDHARI(2204101055)

to
the Indian Institute of Technology Guwahati

in partial fulfillment of the requirements for the award of the Degree
of
Master of Technology
in
Computer Science and Engineering

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati,
Guwahati – 781039
May 2023

DECLARATION

We hereby declare that the project report entitled "**Semantic Terrain Segmentation**" submitted by us to the Indian Institute of Technology Guwahati during the academic year 2022-24 in partial fulfillment of the requirements for the award of Degree of Master of Technology in Computer Science and Engineering is a record of bonafide project work carried out by us under the guidance and supervision of Prof. Debanga Raj Neog. We further declare that the work reported in this project has been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other University.

Place: Guwahati
Date: July 29, 2023

BASU HELA (2204101013)
HARSH VERMA (2204101021)
HIMANSHU CHHABRA (2204101023)
SOUMYA ASATI (2204101046)
VAISHALI CHAUDHARI (2204101055)

DEPARTMENT OF Computer Science and Engineering
Indian Institute of Technology Guwahati,
Guwahati - 781039

CERTIFICATE

This is to certify that the report entitled "**Semantic Terrain Segmentation**" submitted by us to the Indian Institute of Technology Guwahati in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other Universities or institutes for any purpose.

GUIDE	HEAD OF THE DEPARTMENT
Prof. Debanga Raj Neog	Prof. Jatindra Kumar Deka

Contents

0.1 Problem Statement

We address the problem of assessing the mobility of three different types of robots in various terrains. The robots under consideration have distinct movement configurations: legs, belts, and wheels. The objective is to develop a Deep Neural Network (DNN) capable of performing semantic image segmentation, which enables the differentiation of four categories based on robot mobility: legged traversability, belted traversability, wheeled traversability, and non-traversable terrain. The primary focus of this study is to determine which type of robot is best suited for a given task or terrain, considering the advantages, disadvantages, and constraints associated with each robot's mobility capabilities. To achieve this, we propose developing a DNN model to analyze terrain images and classify them into four predefined categories based on robot mobility.

Robots Mobility Configuration

1. Legged Robot: The legged robot can traverse complex terrains by adapting its movements and leveraging its leg-based locomotion. It offers the advantage of enhanced agility and adaptability in uneven or challenging landscapes. However, legged robots may face speed, stability, and energy efficiency limitations compared to other configurations.

2. Belted Robot: The belted robot utilizes a system of belts or tracks for locomotion, allowing it to navigate through diverse terrains such as rough or slippery surfaces. These robots often exhibit superior stability, traction, and load-carrying capacity. However, they might encounter challenges when encountering obstacles or operating in tight spaces due to their larger footprint.

3. Wheeled Robot: The wheeled robot employs wheels as its primary mode of locomotion, making it well-suited for traversing smooth, flat, and well-maintained terrains. These robots typically excel in speed, energy efficiency, and maneuverability in open areas. However, they may face difficulties in uneven or rugged terrains due to adaptability and obstacle negotiation limitations.

Semantic image segmentation involves partitioning an image into meaningful regions and assigning each pixel to a specific class or category. In this context, the aim is to develop a DNN model that can analyze terrain images and accurately classify each pixel into one of the four categories based on robot mobility: legged traversability, belted traversability, wheeled traversability, and non-traversable terrain. This will enable a comprehensive understanding of the terrain's characteristics and suitability for different robots. This report outlines the problem statement of developing a DNN model for semantic image segmentation to differentiate between four categories of robot mobility: legged traversability, belted traversability, wheeled traversability, and non-traversable terrain. The study analyzes terrain images to determine the most suitable robot type for specific tasks, considering each mobility configuration's advantages, disadvantages, and constraints. The proposed solution aims to enhance the decision-making process for selecting the appropriate robot configuration in various environments, improving efficiency and performance in robotic applications.

0.2 Description of DataSet

Dataset Name and Inspiration

The dataset is named "Vale," which draws inspiration from its capture location, Campus Do Vale at The Federal University of Rio Grande Du Sul (UFRGS), Brazil. The dataset was specifically captured to address segmenting and classifying terrain based on the movability constraints of three different types of mobile robots.

Image and Segmentation Mask Information

The Vale dataset comprises 600 high-resolution RGB images of various terrains. Each image in the dataset is in full HD resolution, with dimensions of 480x270 pixels and three color channels (RGB). Alongside each image, there is an associated segmentation mask, resulting in a total of 600 segmentation masks.

Classes and Categories

The segmentation masks in the dataset consist of four distinct classes, representing different categories based on the mobility constraints of robots. The color coding used in the segmentation masks is as follows:

- Yellow: Belted Robots
- Green: Wheeled Robots
- Orange: Legged Robots
- Red: Non-Traversable Terrain

These colors are assigned to corresponding regions in the segmentation masks, indicating the class or category to which each pixel belongs. By leveraging these segmentation masks, the dataset enables the training and evaluation of models for semantic image segmentation and robot mobility classification.

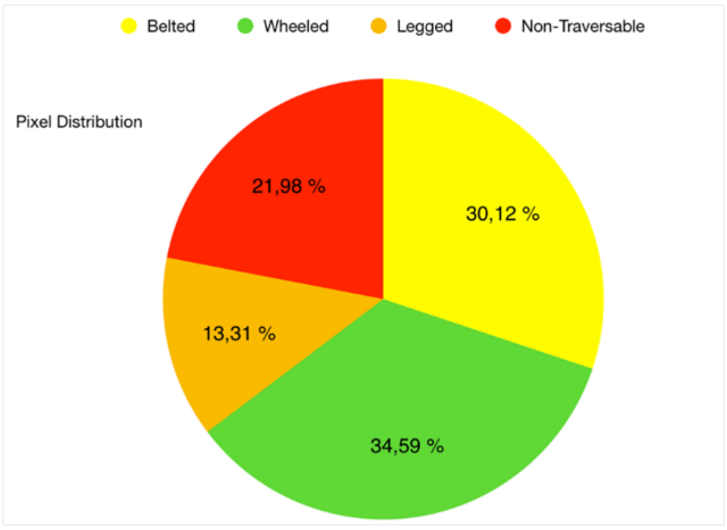

Figure 1: Pixel Distribution

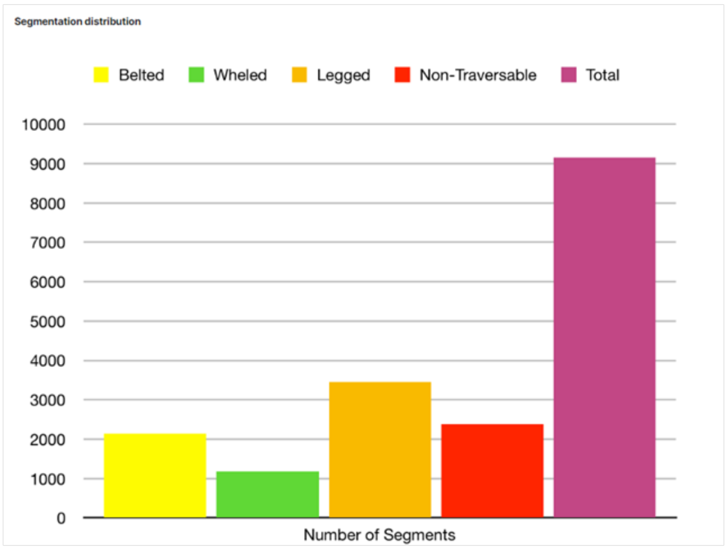

Figure 2: Segment Distribution

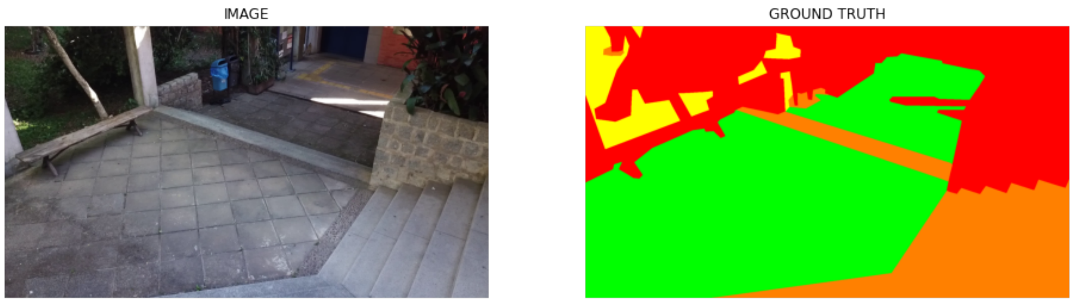

Figure 3: Image and its corresponding segmentation

0.2.1 Data Pre-processing

Bad Image Removal

During the initial data processing step, it was discovered that one image in the dataset could not be read using the OpenCV library. Consequently, this particular image was discarded from the dataset to ensure the integrity and consistency of the data. After this step, the dataset consists of 599 images for further analysis.

One-hot Encoding

A one-hot encoding technique is employed to represent the categories of classes in the dataset. For each pixel in the segmentation masks, a vector of size 4 is created. Each entry in the vector represents a specific class, and the value is either 0 or 1. If a pixel belongs to a particular class, the corresponding entry in the vector is set to 1, while the other entries are set to 0. This encoding scheme allows for a more efficient representation and processing of the class labels during training and evaluation.

Augmentation

Data augmentation techniques are applied to create new training data by applying various transformations to the existing images. These transformations help increase the training dataset's diversity and variability, improving model generalization and robustness. The augmentation techniques used in this project include:

- Horizontal Flip: The image is horizontally flipped with a probability of 0.5, resulting in a mirrored version of the original image.

- Vertical Flip: The image is vertically flipped with a probability of 0.5, producing an inverted version of the original image.

- Rotation: The image is rotated by an angle where the limit is set to 30, meaning that each image can be rotated by a maximum of 30 degrees in either the clockwise or counter-clockwise direction and with a probability of 0.5, allowing the model to learn from various orientations of the terrains.

- Elastic Transform: The elastic transform with parameters alpha is 1, sigma is 50, affine is 50, and p is 0.5 introduces local deformations or distortions to the images in the dataset. These distortions simulate variations in the terrain and enhance the model's ability to generalize. The alpha parameter controls the intensity of the deformation, with a value of 1 indicating a moderate level of distortion. The sigma parameter determines the spatial extent of the distortions, set to 50. The affine parameter controls the magnitude of the accompanying affine transformation, while the probability parameter ($p = 0.5$) specifies the likelihood of applying the elastic transform to each image.

- Brightness-Contrast: The image's brightness and contrast with parameters brightness limit is 0.2, contrast-limit is 0.2, p is 0.5 adjusts the brightness and contrast of images in the dataset. It allows for a variation of up to 20% in brightness and contrast levels and is applied with a probability of 0.5 to introduce diversity and improve the model's robustness.

- Gamma: A gamma correction is applied to the image with parameters gamma-limit is (80,120), p is 0.5, adjusting the gamma values of the images in the dataset. By altering the gamma values, this technique introduces variations in brightness and contrast. The gamma-limit parameter specifies the range within which the gamma values will be adjusted, allowing for a diverse range of brightness and contrast modifications. The augmentation is applied with a probability of 0.5, contributing to improved model robustness and adaptability.
- Coarse dropout: Random patches of pixels are masked or dropped out from the image with a probability of 0.3, introducing localized missing information.
- Color Jitter: The color jitter augmentation technique, with parameters saturation=0.3, hue=0.1, p=0.5, applies random color transformations to the images in the dataset. By adjusting the saturation and hue values, this technique introduces variability in the color distribution. The saturation parameter determines the range of adjustment for color saturation, while the hue parameter controls the range of adjustment for color hue. The augmentation is applied with a probability of 0.5, enhancing the model's ability to handle different color variations and improving its robustness.

Train-val-Test Splitting

The dataset is split into train, validation, and test data. The training data comprises 479 images, which accounts for 80% of the dataset. The validation data comprises 60 images, equivalent to 10% of the dataset. Similarly, the test data comprises 60 images, representing the remaining 10% of the dataset. This division allows for appropriate training, validation, and evaluation of the deep neural network model. For the final model, the train and validation dataset were combined to create the train dataset.

Data Loading

The train and validation data subsets are loaded using a data loader, which facilitates the batching and shuffling of the data during training. A batch size is set to 15. Additionally, the shuffling parameter is set to True, ensuring that the order of the images in each batch is randomized to avoid any bias during training. We are dividing each batch into 4×4 grid, and passing these grids randomly to the model one at a time.

0.3 Literature Survey

In our work, we drew inspiration from the "U-Net: Convolutional Networks for Biomedical Image Segmentation" research paper for its deep convolutional neural network architecture. However, we made modifications to adapt it to our specific task of segmenting full HD RGB images of terrains and producing multi-class segmentation masks.

We also incorporated new advancements into our work, such as Batch Normalization and Attention Mechanisms, Xavier Initialization, gradient clipping and so on.

Batch Normalization

We referred to the paper "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift" to implement Batch Normalization in our network architecture. Batch Normalization is a technique that helps accelerate the training process of deep neural networks by reducing internal covariate shift. It normalizes the activations of each mini-batch within the network, making the optimization process more stable and efficient.

Attention Mechanisms

To enhance the performance of our model, we explored the concept of Attention Mechanisms. We referred to a Medium article, "Self-Attention in Convolutional Neural Networks," to understand and implement attention mechanisms in convolutional neural networks. Attention mechanisms allow the network to learn to focus on the most informative regions of the image while disregarding irrelevant or noisy regions. This enables the model to improve its segmentation accuracy by selectively attending to relevant features.

By incorporating these approaches into our work, we aimed to improve the performance and accuracy of our deep neural network architecture for semantic image segmentation of terrain images.

Xavier Initialization

The Xavier initialization method sets the initial weights of each neuron in a layer by sampling values from a Gaussian distribution with zero mean and a variance that depends on the number of incoming and outgoing connections to that neuron. Precisely, the variance is calculated as:

$$\text{variance} = \frac{2}{n_{in} + n_{out}}$$

where n_{in} is the number of incoming connections to the neuron and n_{out} is the number of outgoing connections from the neuron. By using this initialization scheme, Xavier initialization ensures that the initial weights are not too small or too large, which can prevent issues such as vanishing or exploding gradients during training. It balances the initialization so that the signals can flow well through the network without being amplified or attenuated.

Gradient Clipping

By limiting the magnitude of the gradients, gradient clipping helps stabilize the training process and prevents the weights from changing too drastically in a single update. It allows for smoother and more controlled learning, facilitating convergence and improving the overall training performance of the model.

0.4 Methodology

0.4.1 UNet Architecture

The U-Net architecture is a widely used convolutional neural network (CNN) architecture designed for biomedical image segmentation tasks. In our work, we have adapted the U-Net architecture for semantic image segmentation of terrain images. The architecture consists of an encoder and a decoder, connected by skip connections that allow the decoder to access feature maps from the encoder at multiple scales.

BASE MODEL

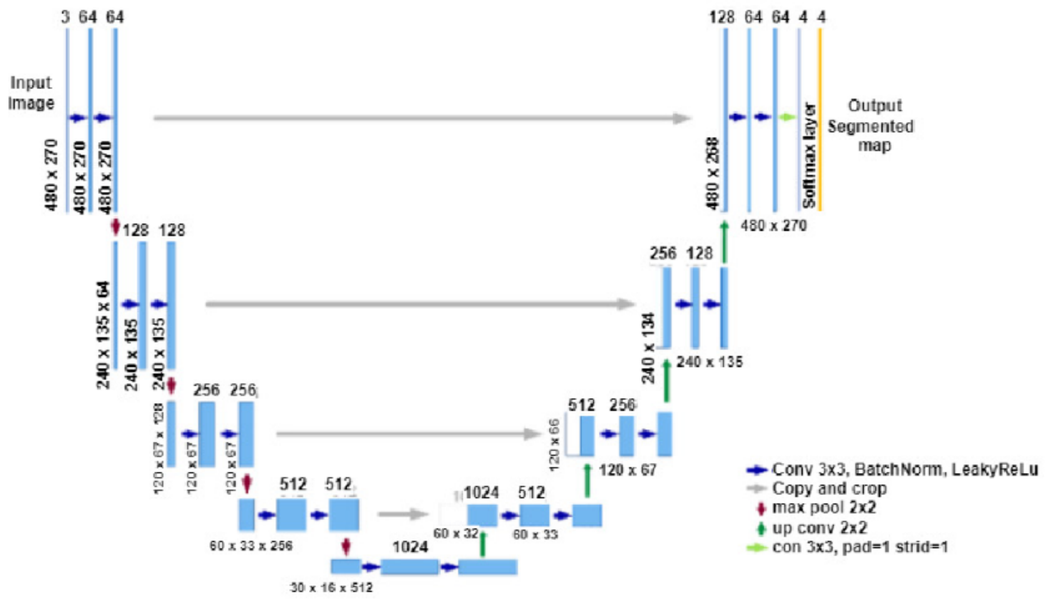

Figure 4: BASE MODEL

U-Net is an architecture for semantic segmentation.The input to this is 480 × 270 × 3.
It consists of a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of DoubleConv blocks followed by 2 × 2 max pooling operation with stride 2 for downsampling.The DoubleConv block consists of Same Convolution → BatchNorm → LeakyReLU → Same Convolution → BatchNorm → LeakyReLU. At each downsampling step, we double the number of feature maps.

At the end of this, we have a bottleneck layer that uses DoubleConv block to double the number of feature maps without changing the spatial dimension of each feature map.

Every step in the expansive path consists of a Transposed convolution followed by Skip connection followed by Double Convolution. Transposed convolution doubles the spatial dimensions of each feature map. Double Convolution halves the number of feature maps.

Further, the convolution and SoftMax layers are applied to the final output to get the final output size of $480 \times 270 \times 4$. Thus, we get the output of size $480 \times 270 \times 4$, which is the required segmented mask.

MODEL-1

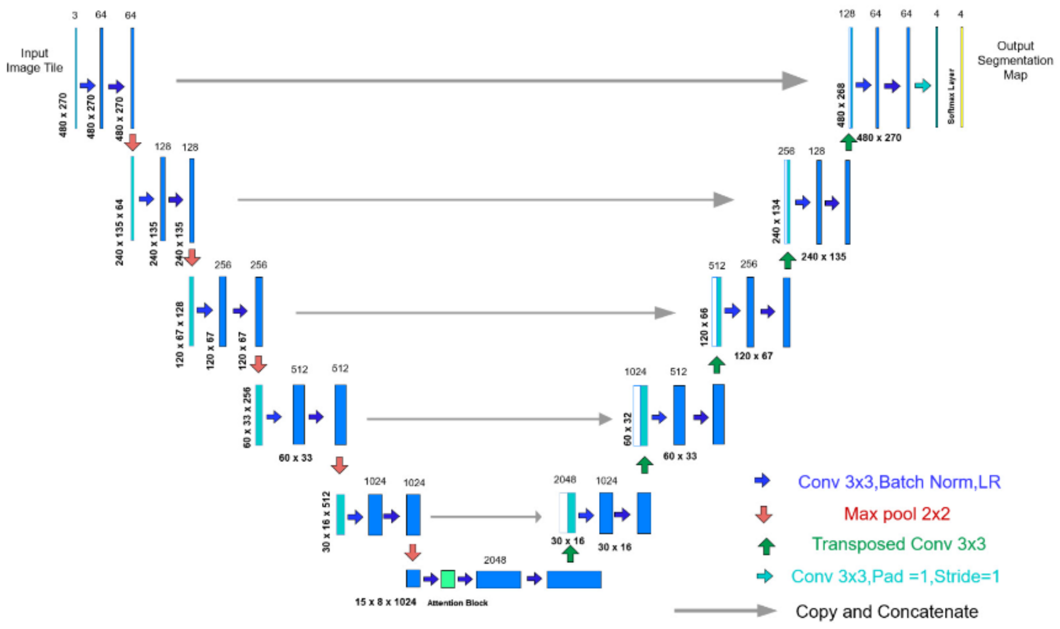

Figure 5: Model-1 5 levels with Attention block before bottleneck layer

This model consists of 5 encoder and decoder layers and a bottleneck layer. We have added an attention layer/block before the bottleneck layer. Rest are same as the base model.

MODEL-2

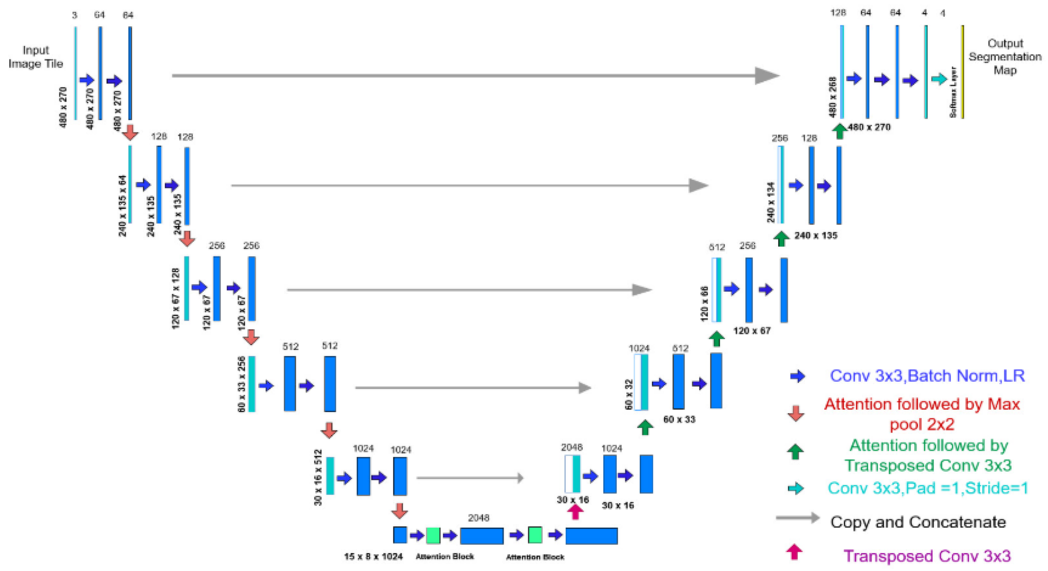

Figure 6: Model-2 5 levels with Attention block at each level

The model shown in Fig. 6 has 5 layers of encoder and decoder and a bottleneck layer. We have added attention layers after the DoubleConv layer in contracting path, before and after the bottleneck layer, and after the DoubleConv layer in expanding path. Rest are same as the base model.

MODEL-3

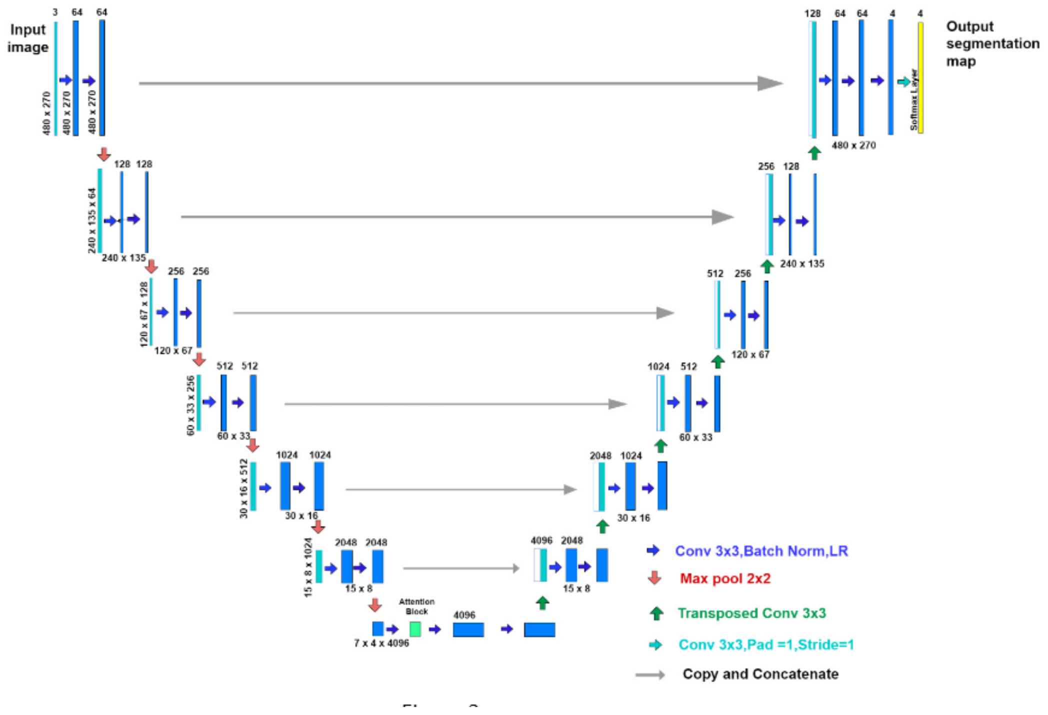

Figure 7: Model-3 6 levels with Attention block before bottleneck layer

The model shown in Fig. 7 is a modified U-Net-based model. It has 6 layers of encoder and decoder along with a bottleneck layer. We have added an attention layer/block before the bottleneck layer. Rest are same as the base model.

MODEL-4

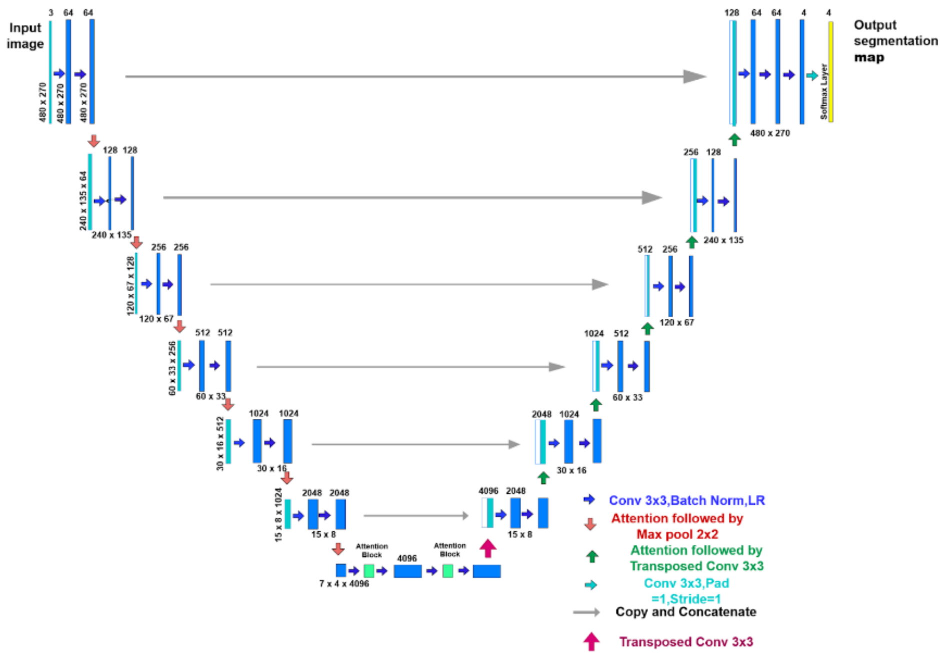

Figure 8: Model-4 6 levels with Attention block at each level

The model shown in Fig. 8 is a modified U-Net-based model. It has 6 layers of encoder and decoder along with a bottleneck layer. We have added attention layers after the DoubleConv layer in contracting path, before and after the bottleneck layer, and after the DoubleConv layer in expanding path. Rest are same as the base model.

FINAL MODEL

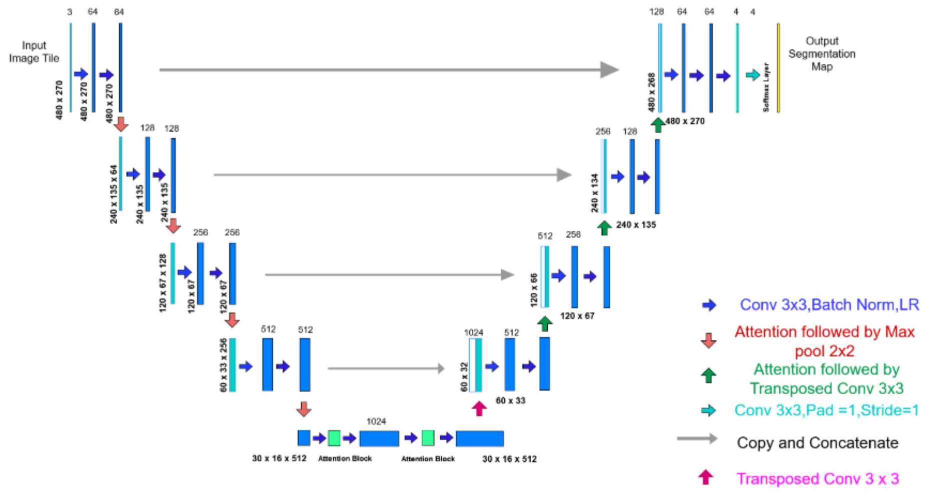

Figure 9: Final model, 4 levels with Attention block at each level

The model shown in Fig. 9 has 4 layers of encoder and decoder and a bottleneck layer.. We have added attention layers after the DoubleConv layer in contracting path, before and after the bottleneck layer, and after the DoubleConv layer in expanding path. Rest are same as the base model.

0.5 Results and Evaluations

Evaluation Metrics for Semantic Image Segmentation

In the validation phase of our project, we employed two evaluation metrics to assess the performance of our semantic image segmentation model: Accuracy and Dice Score.

Accuracy

Accuracy is a commonly used metric that measures the overall correctness of the model's predictions. It is calculated by dividing the number of correctly classified examples by the total number of classified examples. In semantic image segmentation, accuracy is computed by comparing each predicted pixel to its corresponding ground truth label. The formula for accuracy is:

$$\text{Accuracy} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}$$

Dice Score

The Dice Score is a metric commonly used in medical image segmentation and applies to other tasks. It measures the overlap or similarity between the predicted segmentation and ground truth masks. The Dice Score ranges from 0 to 1, 1 indicating a perfect overlap between the two masks. The formula for calculating the Dice Score is as follows:

$$\text{Dice Score} = \frac{2 \times \text{Intersection(Truth and Predicted)}}{\text{Ground Truth} + \text{Predicted}}$$

Here, the intersection refers to the number of pixels where both the ground truth and predicted masks have a positive value (indicating the presence of the class). The numerator accounts for the overlap, while the denominator represents the sum of pixels in both masks.

Both accuracy and Dice Score are valuable metrics for evaluating the performance of semantic image segmentation models. Accuracy provides an overall measure of classification correctness. At the same time, the Dice Score explicitly assesses the similarity between the predicted and ground truth masks, providing insights into the model's ability to capture the precise boundaries and details of the segmented objects.

Results of base model during training phase

The base model was trained for 100 epochs. The learning rate was set to 0.05, batch size 16 and optimizer used was Adam. The loss function used was Cross-Entropy loss.
Although the accuracy we got was good, the actual segmentation mask generated was not that good. So we created 4 more models and then compared those.

EPOCHS	MEAN TRAINING LOSS	VALIDATION ACCURACY
25	8.127	81.17
50	3.047	86.67
75	1.890	88.68
100	1.570	89.19

Table 1: Evaluation Metric from Base Model

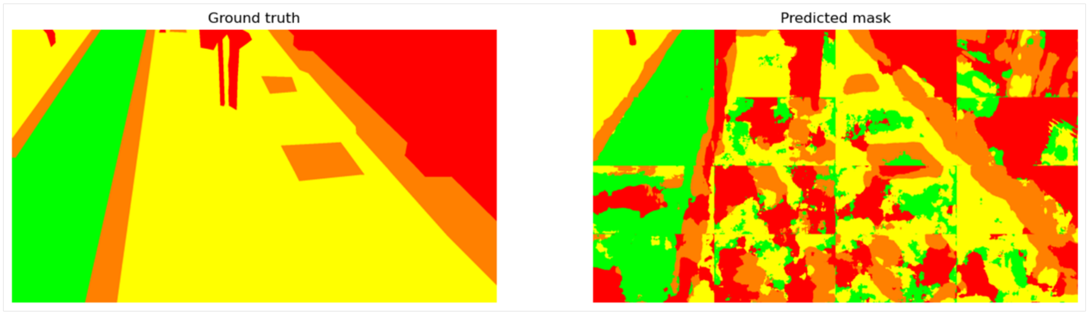

Figure 10: Example output of an image in the test

Results from different model architectures

All four models were trained for 12 epochs. The learning rate was set to 0.05, batch size 10 and optimizer used was Adam. The loss function used was Cross-Entropy loss.The model 4 was showing good performance, so we trained it for more epochs, but it was overfitting, so we trained the final model.

MODEL	MEAN DICE SCORE	ACCURACY
1	0.60	72.17
2	0.598	73.54
3	0.57	71.88
4	0.636	74

Table 2: Comparison Results of different Models

Results from the final model

The final model was trained for 50 epochs with varying learning rates. The batch size was set to 15, the optimizer used was Adam. The loss function used was Cross-Entropy loss. The model managed to get 0.8536 mean dice score on the test set.

EPOCHS	**LEARNING RATE**	**MEAN TRAINING LOSS**	**MEAN DICE SCORE (TEST)**	**ACCURACY**
0-12	0.0001	7.1437	0.6907	83.22
13-25	0.00001	4.19988	0.77148	90.55
26-39	0.000005	3.1785	0.77059	89.16
39-50	0.0000025	2.361	0.8536	94.36

Table 3: Evaluation Metric from Final Model

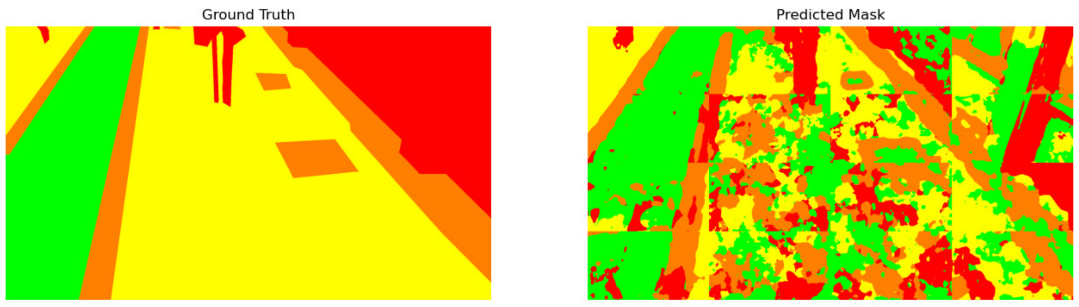

Figure 11: Example output of an image in test set from Final Model

0.6 Conclusion

To address the critical problem of semantic terrain segmentation for robots with different movement configurations, we have proposed a Deep Neural Network (DNN) based on the UNET. Our methodology employs a convolutional neural network that enables pixel-wise segmentation of terrain images into four distinct categories, considering the mobility constraints of robots.

The significance of our work lies in its contribution to the field of robotic mobility and terrain analysis. Our DNN model provides valuable information for robot navigation, path planning, and decision-making processes by accurately classifying and segmenting terrains. This has the potential to impact a wide range of applications, such as autonomous robots in outdoor environments, search and rescue missions, agricultural robotics, and exploration of challenging terrains.

While our proposed final model has shown promising results, there are still avenues for future research and improvement. One potential direction is to change the loss function to dice loss or boundary loss or a linear combination of both to improve the segmentation near boundaries and the overall image.
Furthermore, post-processing techniques such as morphological operations can be used to remove small misclassified regions and improve the visual quality of the generated segmentation masks.
Incorporating additional contextual information, or leveraging transfer learning techniques. Furthermore, exploring the applicability of our approach to different types of terrain and diverse robot configurations could broaden the scope of its practical use and facilitate the deployment of robots in various real-world scenarios.
In conclusion, our project contributes to robotic mobility and terrain analysis by developing a Deep Neural Network for semantic terrain segmentation. The potential impact of our work on various applications is significant, and further research endeavors can focus on refining the model's performance and extending its applicability to diverse terrains and robot configurations.

You can access the code on Git at the following link:
`https://gitlab.com/basu1999/improved-unet`

0.7 References

- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv:1505.04597
- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv:1502.03167
- Medium link for Attention
  https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00