Image segmentation of off-road Unstructured environment

BACHELOR OF TECHNOLOGY

IN

ROBOTICS AND AUTOMATION

SUBMITTED BY

ARNAV DEOGHARE

ADITYA SULE

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

॥वसुधैव कुटुम्बकम्॥

Pune – 412115

THE YEAR 2021-25

INDEX

Abstract

Image segmentation of off-road unstructured environments is a challenging task due to the presence of diverse and unpredictable conditions such as uneven terrain, varying illumination, and occlusions. Traditional segmentation methods often struggle to achieve satisfactory results in such scenarios.

Recent research has focused on developing novel approaches that can overcome these challenges. One promising direction is the use of deep learning-based methods. Deep learning models can be trained on large datasets of off-road images to learn robust features that are invariant to changes in lighting, pose, and appearance. Additionally, deep learning models can be combined with other techniques such as attention mechanisms and domain generalization to further improve performance.

The development of effective image segmentation methods for off-road environments is essential for a wide range of applications, such as autonomous driving, robotic navigation, and remote sensing. By accurately segmenting off-road images, these applications can gain a better understanding of the surrounding environment and make more informed decisions.

Introduction

In the current age marked by remarkable progress in the fields of computer vision and image processing, the demand for precise and efficient image segmentation methods has never been more pronounced. Image segmentation holds a pivotal role across a wide spectrum of domains, encompassing areas as diverse as autonomous vehicles, environmental surveillance, and precision agriculture. Within this report, we embark on a thorough exploration of image segmentation, with a specialized emphasis on the intricate challenges presented by the dynamic and demanding off-road environments.

Off-road environments are a complex and ever-changing landscape, presenting unique challenges for image segmentation. The terrain can vary from rugged wilderness to agricultural fields, and often, conventional segmentation methods fall short when confronted with these complexities. Understanding and effectively processing images in these environments have paramount importance, especially as off-road navigation continues to gain importance in applications such as autonomous vehicles, forestry management, and environmental monitoring.

This report serves as a comprehensive exploration of image segmentation in off-road environments, detailing the methodologies, challenges, and promising solutions that have emerged in recent years. We will examine the significance of image segmentation in this context, highlight the hurdles that must be overcome, and survey the innovative techniques that have been developed to address these challenges. By delving into this topic, we aim to provide a valuable resource for researchers, engineers, and professionals working in various fields that require robust image analysis in off-road settings.

1.1Computer Vision

Computer vision represents an interdisciplinary domain that converges at the junction of computer science, artificial intelligence, and image processing. Its core purpose is to empower computers with the ability to not only perceive but also comprehend and derive valuable insights from visual data present in our environment. In essence, computer vision strives to replicate and elevate the perceptual capabilities of human vision by creating intricate algorithms and systems geared toward the extraction and interpretation of rich information embedded within images and videos.

1.2 Semantic segmentation

Semantic segmentation is a fundamental task in computer vision that seeks to understand the scene by assigning a semantic label to each pixel in an image. This contrasts with image classification, which only assigns a single label to the entire image. For instance, in an image of a cityscape, semantic segmentation would label each pixel as belonging to a class such as road, building, vegetation, or person. This fine-grained understanding of the scene is essential for a wide range of applications, such as autonomous driving, robotic navigation, and medical image analysis.

Manual methods for image segmentation are based on the idea of manually defining rules or criteria for grouping pixels. Some common manual segmentation methods include:

- **Thresholding Segmentation**: Thresholding is one of the simplest segmentation techniques. It involves setting a threshold value and classifying each pixel as either foreground or background, based on its intensity or color compared to the threshold. Thresholding is commonly used for binary segmentation tasks, such as separating objects from the background in simple images.
- **Region-based Segmentation:** • Region-based segmentation, groups pixels into regions based on similarity criteria, such as color, texture, or intensity. Common algorithms include region growing, which starts from seed pixels and expands regions based on similarity, and mean-shift, which iteratively updates the center of regions.

  Region-based segmentation is useful for segmenting objects with relatively uniform appearance. and can be used in image segmentation, medical image analysis, and image segmentation with interactive user guidance. It can be used in image segmentation, medical image analysis, and image segmentation with interactive user guidance.
- **Clustering-based Segmentation:** Clustering algorithms, like k-means, group pixels with similar features into clusters, representing different segments. Clustering-based segmentation is suitable for images with clear separations between segments based on feature space, such as color or texture.
- **Edge-based Segmentation:** Edge-based segmentation aims to detect sharp transitions or edges between different regions in an image. Techniques like the Canny edge detector or the Sobel operator are used to identify edges. Edge-based segmentation is commonly used in

applications where precise boundary detection is crucial, such as object detection and boundary extraction.

- **Graph-based Segmentation:** graph-based segmentation formulates the image as a graph, where pixels are nodes, and edges represent pairwise similarities. Algorithms like normalized cut use graph partitioning to segment the image. Graph-based segmentation is useful for complex images with non-uniform regions

Manual methods for image segmentation can be time-consuming and labor-intensive, and they may not be able to achieve the same level of accuracy as deep learning-based methods. However, they can be useful for small datasets or for cases where it is important to have fine-grained control over the segmentation process.

1.3 Deep learning-based methods

For semantic segmentation have achieved state-of-the-art results on a wide range of benchmarks. These methods typically use a convolutional neural network (CNN) to extract features from the image and then use a decoder to generate a segmentation map. Some common deep-learning architectures for semantic segmentation include:

- U-Net: The U-Net is a widely used architecture for semantic segmentation that consists of an encoder-decoder structure. The encoder is responsible for extracting features from the image, while the decoder is responsible for generating the segmentation map.
- Fully convolutional networks (FCNs): FCNs are a class of CNNs that have been adapted for semantic segmentation. FCNs use transposed convolutions to upsample the feature maps produced by the encoder, allowing them to generate segmentation maps at the same resolution as the input image.
- DeepLab: Deep Lab is a family of deep learning architectures for semantic segmentation that use atrous convolutions to enlarge the field of view of the network. This allows the network to capture long-range dependencies in the image.
- Deep learning-based methods for semantic segmentation are effective in a wide range of applications. However, they can be computationally expensive to train and may require large amounts of labeled data.

In conclusion, semantic segmentation is a challenging task that has been addressed using both manual and deep learning-based methods. Manual methods are typically time-consuming and labor-intensive, but they can be useful for small datasets or for cases where it is important to have fine-grained control over the segmentation process. Deep learning-based methods have achieved state-of-the-art results on a wide range of benchmarks, but they can be computationally expensive to train and may require large amounts of labeled data.

1.4 Convolutional Neural Networks (CNNs)

Deep learning methods have emerged as a powerful tool for image segmentation, particularly in challenging off-road environments. Their ability to learn hierarchical representations of images enables them to extract salient features that are invariant to variations in lighting, pose, and appearance.

In the context of image segmentation, CNNs are typically employed in a supervised learning setting. A CNN is first trained on a dataset of labeled images, where each pixel is assigned a semantic label. During training, the CNN learns to associate specific patterns of features with corresponding labels. Once trained, the CNN can be used to segment new images by assigning labels to each pixel based on the learned patterns.

The effectiveness of CNNs for image segmentation stems from their ability to learn both local and global features. In the initial layers of a CNN, convolutional filters are applied to extract low-level features such as edges, textures, and simple shapes. As the network deepens, subsequent layers learn increasingly complex features that capture the global structure of the image. This hierarchical representation allows the CNN to identify both fine-grained details and large-scale patterns.

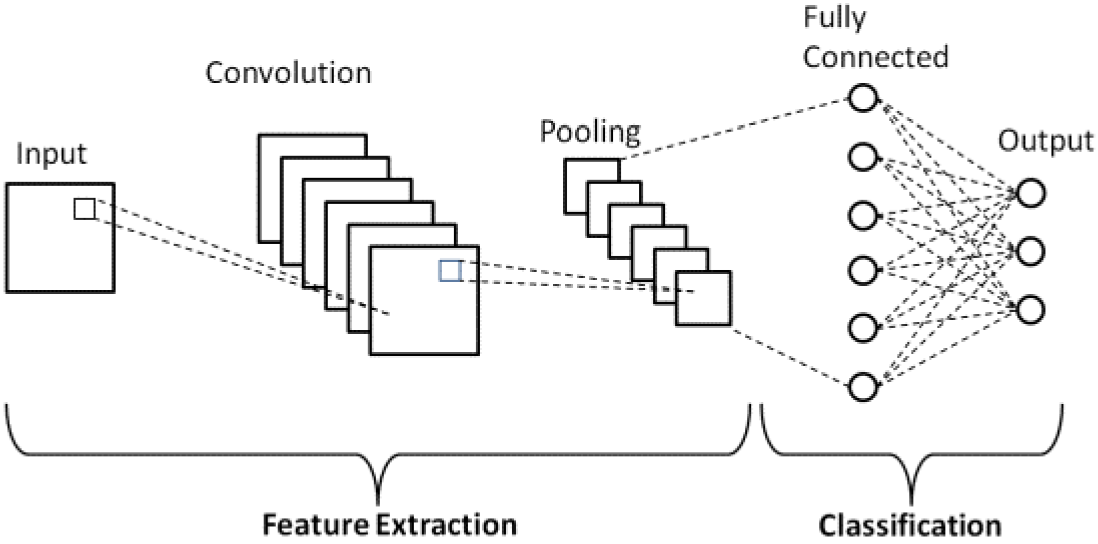


Fig: Diagram of working of CNN

In addition to their superior performance, CNNs offer several other advantages for image segmentation. CNNs can learn complex features from data without the need for explicit feature engineering. This makes them well-suited for tasks where traditional methods struggle, such as segmenting images with complex or irregular shapes. Additionally, CNNs can be trained on relatively small datasets, making them a viable option for applications where labeled data is scarce.

Overall, CNNs have proven to be a powerful tool for image segmentation. Their ability to learn hierarchical representations of images enables them to extract salient features that are invariant to variations in lighting, pose, and appearance. As a result, CNNs have become the state-of-the-art for image segmentation in a wide range of applications.

1.5 U-Net

The U-Net architecture, a type of CNN, has become popular for image segmentation tasks. It combines convolutional layers for feature extraction with upsampling layers to refine segmentation masks, resulting in highly accurate segmentation results.

U-Net Architecture

U-Net is a convolutional neural network (CNN) architecture that was developed for image segmentation tasks, particularly in the field of medical image analysis. It was introduced by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015, and it has since become a fundamental building block in various computer vision applications, not limited to medical imaging.

The U-Net architecture derives its name from its characteristic U-shaped design, which resembles an encoder-decoder network with skip connections. This unique structure allows U-Net to effectively segment and classify objects within images by preserving high-resolution spatial information while gradually learning abstract features through downsampling and upsampling.

Key components of the U-Net architecture:

1. Encoder Path: The top part of the U-Net forms the encoder path. It is responsible for capturing features from the input image at multiple scales. This path typically consists of multiple convolutional layers followed by max-pooling operations, which reduce the spatial dimensions of the input image while increasing the depth of feature maps.

2. Bottleneck: The encoder path eventually leads to a bottleneck layer, where the spatial resolution is at its minimum. This layer captures the most abstract and high-level features learned from the image.

3. Decoder Path: The decoder path is the bottom part of the U-Net, and it performs the task of upsampling and reconstructing the segmented output. It consists of transposed convolution layers (also known as deconvolution or upsampling layers) and skip

connections. These skip connections connect layers from the encoder path to equivalent layers in the decoder path. The purpose of these connections is to enable the network to merge low-level details from the encoder with high-level context from the bottleneck.

4. Final Output: The final layer of the decoder path often employs a 1x1 convolution layer followed by a softmax activation function to produce the segmentation map. The segmentation map has the same spatial dimensions as the original input image, with each pixel indicating the class or object label.

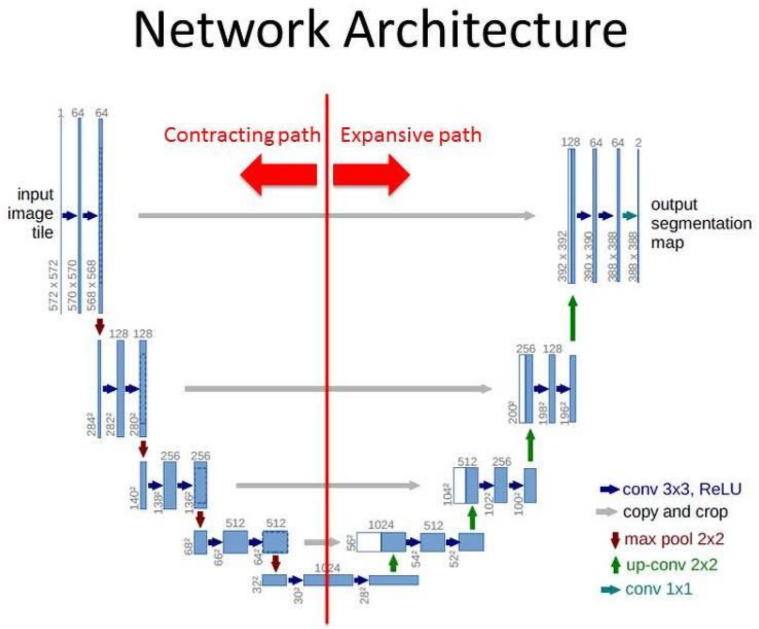


Fig:Unet architecture

The skip connections are a key innovation of the U-Net architecture. They enable the network to retain fine-grained details during the upsampling process, improving segmentation accuracy. These connections also mitigate the vanishing gradient problem by providing a direct path for gradients to flow from the output to the input.

U-Net has been widely adopted in various image segmentation tasks, including medical image segmentation (e.g., tumor detection, organ segmentation, and cell counting), and other applications such as remote sensing, image-to-image translation, and more. Its success can be attributed to its ability to handle relatively small datasets effectively, its architectural simplicity, and its capacity to capture both local and global features in an image.

Since its inception, researchers have developed many variants and improvements of the U-Net architecture to address specific challenges in image segmentation, making it a versatile and valuable tool in the field of computer vision.

.

Literature Review

SL No.	Name Of Author(s)	**Methodology**
[1]	Joshua Knights, Kavisha Vidanapathirana, Milad Ramezani, Sridha Sridharan, Clinton Fookes, Peyman Moghadam	● In this paper, the first extensive lidar dataset captured using a handheld sensor payload is being introduced for long-term place recognition in unstructured, natural surroundings. ● The given dataset is suitable for intra-sequence and inter-sequence place recognition tasks, and establish training and testing splits for benchmarking. ● The goal was to make a dataset for unstructured natural environments
[2]	Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha	● A new architecture is introduced which merges multiscale features and transformers. Visual features and improved

		accuracy are combined with the segmentation head. ● The authors have integrated GA-NAV with TERP and it shows 10% higher success compared to alternative methods ● It is also shown that GA-NAV lowers the false positive rates of forbidden areas by 37%
[3]	Kasi Viswanath , Kartikeya Singh , Peng Jiang, P.B. Sujit and Srikanth Saripalli	● The off-road driving necessity is addressed by grouping the 20 classes in RELLIS-3D and the 24 classes in RUGD into 4 classes, effectively countering the challenge of class imbalance. ● Color layers and K-means clustering are employed by the researchers to identify RGB clusters, aiding in the detection of intricate details within the traversable region.
[4]	Anukriti Singh, Kartikeya Singh , and P.B. Sujit	● In this paper the authors have made a model for semi-supervised semantic

		segmentation in unstructured outdoor environments using self-attention in vision transformers. ● This model has significantly reduced the amount of labeled data required for training and evaluation using an algorithm to automatically select the most diverse set of images per sequence. Less than 25 annotated images during training were used.

2.1 Image segmentation in off-road environments

Image segmentation in off-road environments presents a unique and challenging scenario that demands robust solutions. Off-road environments encompass a wide range of natural and man-made terrains, from rugged wilderness to agricultural fields, construction sites, and beyond. The distinctive features and complexities of these landscapes necessitate specialized image segmentation techniques tailored to the challenges they present. Here, we delve deeper into image segmentation in off-road environments:

2.2 Challenges in Off-Road Image Segmentation:

Varied Terrain: Off-road environments can include a vast array of surfaces, including dirt, rocks, sand, vegetation, and water bodies. These diverse terrains exhibit different textures, colors, and structures, making it challenging to create a uniform segmentation approach.

Obstacles and Occlusions: Off-road environments often feature obstacles like bushes, trees, rocks, and other vehicles. These objects can occlude the view of the terrain and pose difficulties in segmenting the ground and non-ground components.

Dynamic Lighting and Weather Conditions: Off-road environments are subject to dynamic lighting conditions due to changes in sunlight, weather, and shadows. Such variations can impact the appearance of the terrain and affect segmentation accuracy.

Noise and Artifacts: Off-road images can contain a high degree of noise, including sensor noise, lens flares, and reflections, which can interfere with segmentation algorithms.

2.3 Innovations in Off-Road Image Segmentation:

To address the challenges specific to off-road environments, researchers and engineers have developed innovative approaches and technologies:

Sensor Fusion: Off-road vehicles often use a combination of sensors, including cameras, LiDAR, radar, and GPS, to enhance image segmentation. Sensor fusion techniques integrate data from multiple sensors to create a more comprehensive and robust understanding of the environment.

Semantic Segmentation: Deep learning-based techniques, such as convolutional neural networks (CNNs), have been employed for semantic segmentation in off-road settings. These models can classify every pixel in an image into meaningful categories, distinguishing between terrain types, obstacles, and other objects.

Adaptive Algorithms: Adaptive image segmentation algorithms have been designed to handle dynamic lighting conditions. These algorithms adjust segmentation parameters in real time based on the current lighting environment, improving accuracy.

Machine Learning for Object Detection: Object detection techniques are used to identify and track obstacles and vehicles in off-road environments. Combining object detection with image segmentation allows for a more comprehensive understanding of the scene.

Terrain Classification: Image segmentation in off-road environments often involves terrain classification to distinguish between traversable and non-traversable areas. This is particularly important for autonomous vehicles and robotics.

Real-Time Processing: In applications like autonomous driving, real-time image segmentation is essential. Efficient algorithms and hardware accelerators are developed to ensure low-latency segmentation results

2.4 Research Gap

- Handling class imbalance:

  Handling class imbalance in image segmentation is a significant challenge, particularly when some classes are substantially more prevalent than others. In scenarios like road and vegetation segmentation, the high-class imbalance can skew the performance of segmentation models, leading to suboptimal results. Here, we'll expand on the issues associated with the class imbalance and strategies to address them:

  Challenges of Class Imbalance:

Model Bias: Class imbalance can introduce bias in segmentation models. They tend to favor the majority class (e.g., road and vegetation) while neglecting the minority classes (e.g., cars and trucks). This can result in poor recognition of critical objects in the image.

Loss of Information: With imbalanced classes, the network may not receive enough examples of underrepresented classes to learn their distinctive features effectively. This can lead to the loss of crucial information.

Evaluation Biases: Traditional evaluation metrics like accuracy can be misleading in imbalanced scenarios. Even a poorly performing model may achieve high accuracy by predicting the majority class for most pixels.

- Fusing multimodal data:

Fusing multimodal data, such as LiDAR point clouds, RGB images, and semantic annotations, is a pivotal aspect of modern computer vision and perception systems. The integration of multiple data sources allows for a more comprehensive and robust understanding of the environment, making it particularly valuable in applications like autonomous navigation, object recognition, and scene understanding.

Methodology

3.1 Dataset Used

Semantic scene understanding is of paramount importance for ensuring the robust and safe autonomous navigation of vehicles, particularly in challenging off-road terrains. Recent advancements in deep learning techniques for 3D semantic segmentation heavily rely on extensive training datasets. Unfortunately, many of the existing datasets primarily focus on urban environments and lack the diversity and complexity needed for off-road scenarios. To bridge this gap, we introduce the **"RELLIS-3D"** dataset, a comprehensive and multimodal dataset collected in a demanding off-road setting.

The RELLIS-3D dataset was meticulously gathered on the Rellis Campus of Texas A&M University and includes annotations for 13,556 LiDAR scans and 6,235 images. This rich dataset poses unique challenges to existing algorithms, primarily related to class imbalance and complex environmental topography. Furthermore, we evaluated the current state-of-the-art deep learning models for semantic segmentation using this dataset.

Our experimental results demonstrate that the RELLIS-3D dataset presents significant challenges for algorithms originally designed for segmenting scenes in urban environments. Beyond its annotated data, this dataset also offers comprehensive sensor data in ROS bag format, encompassing RGB camera images, LiDAR point clouds, stereo image pairs, high-precision GPS measurements, and IMU data.

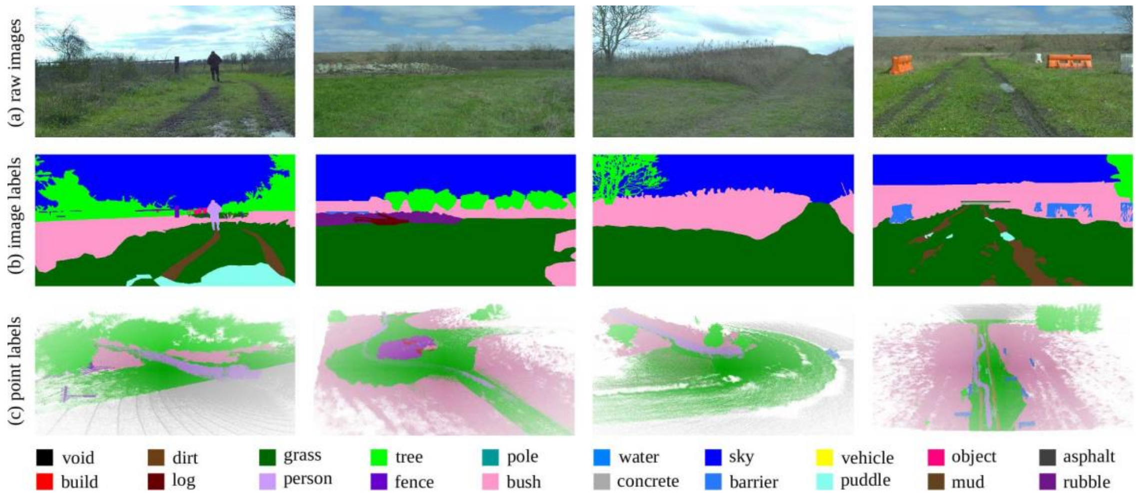
This innovative dataset serves as a valuable resource for researchers, providing the necessary tools to develop more sophisticated algorithms and explore new research avenues aimed at enhancing autonomous navigation in off-road environments.

The RELLIS-3D dataset is a groundbreaking achievement as the first multi-modal off-road navigation dataset, providing synchronized raw sensor data and a substantial number of ground truth annotations. The primary contributions of this dataset can be summarized as follows:

1. The dataset includes five sequences of synchronized sensor data captured during off-road driving, stored in Robot Operating System (ROS) bag format. This data encompasses RGB camera images, LiDAR point clouds, stereo image pairs, high-precision GPS measurements, and IMU data.

2. Across the five sequences, the dataset offers 6,235 pixel-wise image annotations and semantic labels for 13,556 complete LiDAR point cloud scans.

3. A benchmark has been established for the dataset, defining training, validation, and testing sets. An initial analysis has been conducted using state-of-the-art semantic segmentation algorithms for both images and point clouds. These results highlight the complex challenges associated with semantic segmentation in off-road environments and identify areas of research that can be further developed using the RELLIS-3D dataset.



i

Fig:the format of the dataset along with the classes

3.2 Dataset Description

RELLIS-3D comprises five journeys or sequences, and these were recorded on three different unpaved paths at the Ground Research facility located on the Rellis Campus of Texas A&M University. Specifically, three of these sequences took place on the first trail, which had bushes and a few sparse trees. These sequences vary based on the direction the robot moved on the trail and the day the data was collected. Another sequence was captured on the second trail, which passes through a pasture and a forested area. The final sequence was recorded on a hill surrounded by a lake and a highway. These sequences were gathered by manually guiding the robot to follow the trail, and each sequence contains approximately five minutes of data.

To enhance autonomous navigation in off-road scenarios, an ontology of object and terrain classes was established. This classification system is largely based on the RUGD dataset but also includes some unique terrain and object categories that are not found in RUGD. Notably, this dataset introduces classes like mud, man-made barriers, and rubble piles. Furthermore, it offers a more detailed classification for water sources, distinguishing between puddles and deep water, as these two classes present different challenges for most robotic systems. In total, there are 20 classes, including an empty or void class, in the dataset.

3.3 LiDAR Point Cloud Semantic Segmentation

There are the same data splits as used for image data but with lidar scans. The training set comprises 7,800 LiDAR scans, the validation set encompasses 2,413 scans, and the testing set involves 3,343 scans. the point cloud experiments consider only the 15 available classes for segmentation.

Semantic segmentation of LiDAR scans is a critical aspect of understanding and interpreting the 3D environment captured by LiDAR sensors. This process involves assigning a semantic label or category to each point in the LiDAR point cloud, thereby enabling an autonomous system to identify and distinguish different objects and surfaces in its surroundings. However, the lidar scan data has not been used.

3.4 Preprocessing

The data split for the dataset is given in a .lst file. The 3 different lst files contain information on which image file corresponds to which label. The 3 files include train, test, and validation. These files are used along with aid from libraries like OS,shutil, and pandas to create separate folders for train and testing and each having folders for inputs and labels.

We first extract information from the lst file and input them into a pandas data frame, with the column names being either input or labels This data frame is then put into an iterative process and each path is opened and location is pasted into the newly created folder using the shutil code "shutil. copyfile(image_path, destination_path)".This iterative process then slowly pastes the whole series of images into a new file path.

Preprocessing and data augmentation are essential steps in image segmentation. Preprocessing involves transforming the images into a format that is suitable for the segmentation model. This may include tasks such as resizing, normalizing, and converting the images to a specific color space.

Data augmentation is a technique used to artificially increase the size of a training dataset by creating new training data from existing images. This is done by applying various transformations to the images, such as rotating, flipping, cropping, and adding noise. By augmenting the training data, it is possible to make the segmentation model more robust to variations in the input data.

The ImageDataGenerator class in TensorFlow provides a convenient way to apply data augmentation to images. The ImageDataGenerator class can be used to create a generator that yields batches of augmented images. The generator can then be used to train a segmentation model.

The ImageDataGenerator class provides several parameters that can be used to control the data augmentation process. These parameters include:

- rotation_range: The range of degrees by which images can be rotated.
- width_shift_range: The range of values by which images can be horizontally shifted.
- height_shift_range: The range of values by which images can be vertically shifted.
- shear_range: The range of shear angles that can be applied to images.
- zoom_range: The range of zoom factors that can be applied to images.
- horizontal_flip: Whether or not images can be horizontally flipped.
- vertical_flip: Whether or not images can be vertically flipped.
- fill_mode: The mode used to fill in pixels that are exposed by transformations.

By carefully selecting the values of these parameters, it is possible to create a data augmentation strategy that is tailored to the specific needs of the segmentation task.

3.5 Model creation

The U-Net architecture is a convolutional neural network (CNN) that is widely used for image segmentation tasks. The architecture consists of an encoder path and a decoder path. The encoder path is responsible for extracting features from the input image, while the decoder path is responsible for generating a segmentation map.

The encoder path in the given code consists of a series of convolution and max pooling layers. The convolution layers are used to extract features from the input image, while the max-pooling layers are used to reduce the spatial size of the feature maps. The number of convolution

layers and max pooling layers in the encoder path can be varied depending on the specific needs of the image segmentation task. For example, if the task requires the model to extract high-level features from the input image, then a deeper encoder path can be used.

The decoder path in the given code consists of a series of upsampling and convolution layers. The upsampling layers are used to increase the spatial size of the feature maps, while the convolution layers are used to refine the segmentation map. The number of upsampling layers and convolution layers in the decoder path can also be varied depending on the specific needs of the image segmentation task. For example, if the task requires the model to generate a segmentation map with fine-grained details, then a deeper decoder path can be used.

The skip connections between the encoder path and the decoder path are a key feature of the U-Net architecture. The skip connections allow the decoder path to access features from the encoder path at different scales. This helps the decoder path to generate a segmentation map that is both accurate and detailed. The skip connections can be implemented in several ways. For example, the skip connections can be implemented by concatenating the feature

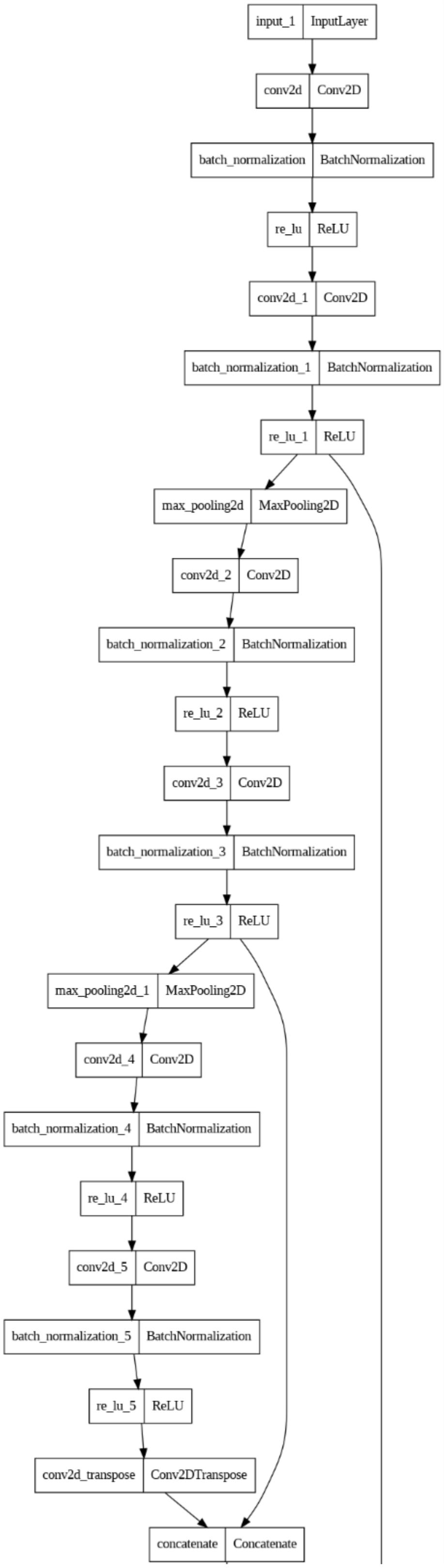

input_1 InputLayer
conv2d Conv2D
batch_normalization BatchNormalization
re_lu ReLU
conv2d_1 Conv2D
batch_normalization_1 BatchNormalization
re_lu_1 ReLU
max_pooling2d MaxPooling2D
conv2d_2 Conv2D
batch_normalization_2 BatchNormalization
re_lu_2 ReLU
conv2d_3 Conv2D
batch_normalization_3 BatchNormalization
re_lu_3 ReLU
max_pooling2d_1 MaxPooling2D
conv2d_4 Conv2D
batch_normalization_4 BatchNormalization
re_lu_4 ReLU
conv2d_5 Conv2D
batch_normalization_5 BatchNormalization
re_lu_5 ReLU
conv2d_transpose Conv2DTranspose
concatenate Concatenate

Fig:Downsampling part of the model

maps from the encoder path and the decoder path, or the skip connections can be implemented by adding the

feature maps from the encoder path and the decoder path.

The given model implements a U-Net model with the following features:

The model uses a series of convolution and max pooling layers in the encoder path to extract features from the input image.

The model uses a series of up-sampling and convolution layers in the decoder path to generate a segmentation map.

The model uses skip connections between the encoder path and the decoder path to allow the decoder path to access features from the encoder path at different scales.

The given creates a U-Net model for image segmentation of an unstructured environment.

3.6 Evaluating the performance.

Evaluating the performance of a machine learning model is an essential step in the development process. Several metrics can be used to evaluate the performance of a machine learning model, such as accuracy, precision,

recall, and F1 score. The choice of metric will depend on the specific needs of the task.

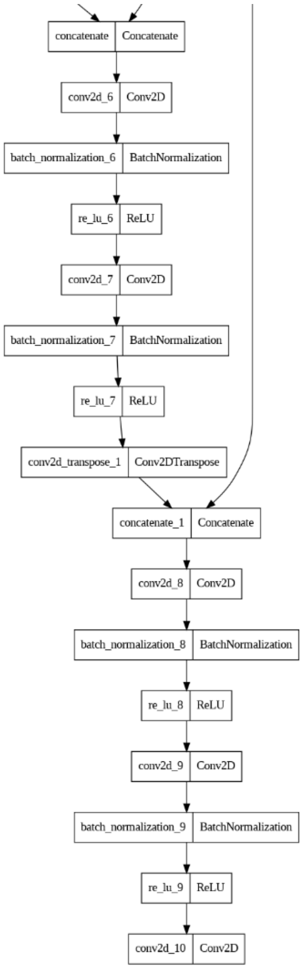

concatenate | Concatenate
conv2d_6 | Conv2D
batch_normalization_6 | BatchNormalization
re_lu_6 | ReLU
conv2d_7 | Conv2D
batch_normalization_7 | BatchNormalization
re_lu_7 | ReLU
conv2d_transpose_1 | Conv2DTranspose
concatenate_1 | Concatenate
conv2d_8 | Conv2D
batch_normalization_8 | BatchNormalization
re_lu_8 | ReLU
conv2d_9 | Conv2D
batch_normalization_9 | BatchNormalization
re_lu_9 | ReLU
conv2d_10 | Conv2D

Fig:Upsampling to get

Segmentation in unet

To get a more complete picture of the performance of a machine learning model,

it is important to use a variety of metrics.

Early stopping is a technique that can be used to prevent a machine-learning model from overfitting the training data. Overfitting occurs when a model learns the noise in the training data too well, and is no longer able to generalize to new data. Early stopping works by monitoring the performance of the model on a validation set. If the performance of the model on the validation set stops improving, then the training process is stopped.

Checkpointing is a technique that can be used to save the state of a machine learning model during the training process. This can be useful if the training process is interrupted, or if the model needs to be restarted from a previous point. Checkpointing works by saving the weights of the model at regular intervals. If the training process is interrupted, then the weights can be loaded from the most recent checkpoint.

In the case of a U-Net model, early stopping and checkpointing can be used to improve the performance of the model. Early stopping can be used to prevent the model from overfitting the training data, while checkpointing can be used to save the state of the model during the training process. This can be useful if the training process is interrupted, or if the model needs to be restarted from a previous point.

We incorporate both techniques to finally improve our accuracy.

Results and Observations

The implementation of image segmentation using U-Net in unstructured environments can yield promising results. However, the performance of the model will depend on several factors, such as the quality of the environment, the choice of hyperparameters, and the complexity of the unstructured environment.

In general, U-Net models can achieve good results on image segmentation tasks. However, the performance of U-Net models can be degraded in unstructured environments due to factors such as noise, occlusion, and variability in the appearance of objects.

To improve the performance of U-Net models in unstructured environments, several techniques can be used. For example, data augmentation can be used to increase the size and diversity of the training data. Additionally, hyperparameters such as the learning rate and the number of epochs can be tuned to improve the performance of the model.

In some cases, it may be necessary to modify the U-Net architecture to improve its performance in unstructured environments.

How these algorithms can be implemented for the use of mobile robots?

U-Net models can be implemented for the use of mobile robots in several ways. For example, a U-Net model can be used to segment the robot's surroundings into different classes, such as obstacles, walls, and free space. This information can then be used to plan a safe path for the robot to navigate.

In addition to obstacle avoidance, U-Net models can also be used for other tasks such as object detection and tracking. For example, a U-Net model could be used to detect and track pedestrians in the robot's surroundings. This information could then be used to avoid collisions or to aid pedestrians.

Mean intersection over union (mIoU) is a measure of the overlap between the predicted segmentation and the ground truth segmentation. It is calculated by taking the mean of the intersection over union (IoU) for each class in the dataset. The IoU for a particular class is calculated by dividing the number of pixels that are correctly segmented for that class by the number of pixels that are either correctly segmented or belong to that class.

The highest accuracy received from unet for the training dataset is **93.4%**

The following is how the model predicts the given classes as per the input classes

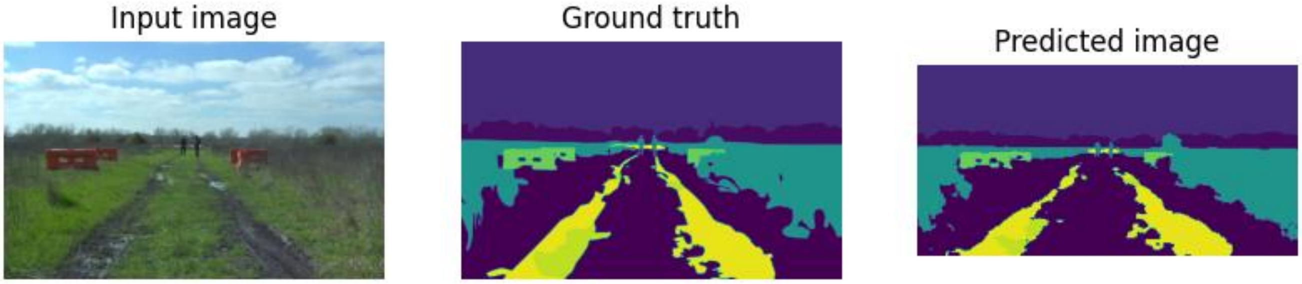


Fig:predicted image generated by the model compared to the ground truth

Next, we compare the accuracy we got per epoch of training in the model training process with the training loss we get. The training accuracy and training loss curves are two important metrics that can be used to monitor the training process of a machine learning model. The training accuracy curve shows how well the model is performing on the training data as the training process progresses. The training loss curve shows how well the model is fitting the training data as the training process progresses.

In general, a good training accuracy and training loss curve will show a steady increase in training accuracy and a steady decrease in training loss. However, the shape of the curves will vary depending on the specific model and the training data.

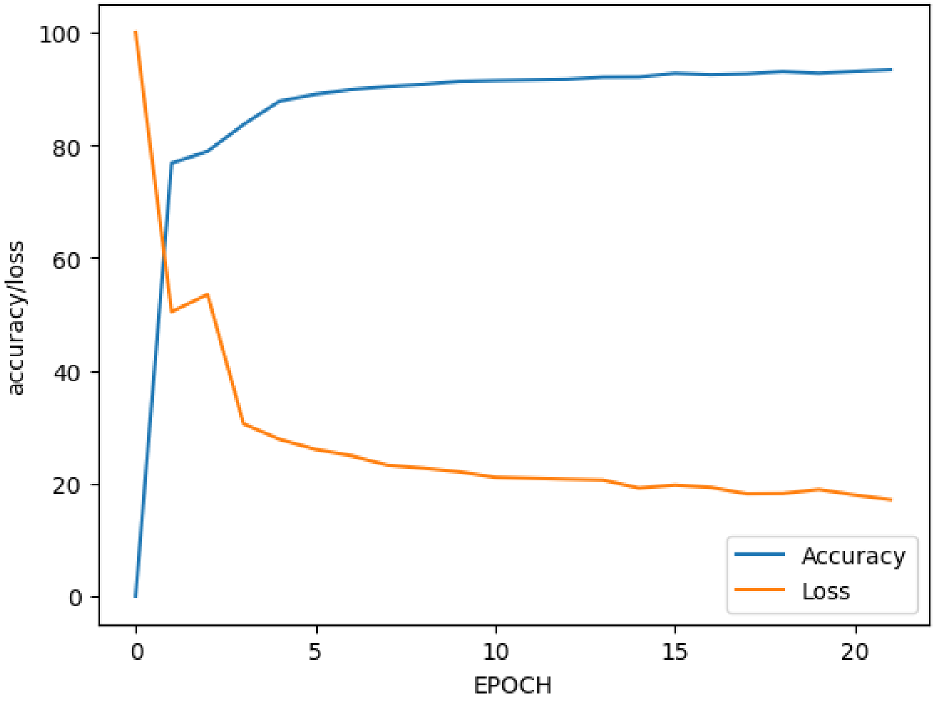


Fig:training accuracy and loss curve

References

[1] RELLIS-3D Dataset: Data, Benchmarks and Analysis. Jiang, P., Osteen, P., Wigness, M., & Saripalli, S. (2020). *ArXiv*. /abs/2011.12954

[2] Guan, T., Kothandaraman, D., Chandra, R., Sathyamoorthy, A. J., Weerakoon, K., & Manocha, D. (2021). GANav: Efficient Terrain Segmentation for Robot Navigation in Unstructured Outdoor Environments. *ArXiv*. /abs/2103.04233

[3] Viswanath, K., Singh, K., Jiang, P., P., S., & Saripalli, S. (2021). OFFSEG: A Semantic Segmentation Framework For Off-Road Driving. *ArXiv*. /abs/2103.12417

[4] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, ´ and A. Joulin, "Emerging properties in self-supervised vision transformers," arXiv preprint arXiv:2104.14294, 2021.

[5] Singh, A., Singh, K., & Sujit, P. B. (2021). OffRoadTranSeg: Semi-Supervised Segmentation using Transformers on OffRoad environments. *ArXiv*. /abs/2106.13963