UNIVERSITY OF TURKU

Self-Supervised Visual Representation Learning with Masked Autoencoders for Traversability Estimation in Outdoor Environments

Master of Science in Technology Thesis

University of Turku

Department of Computing

Robotics and Autonomous Systems

Author:

Lipei Huo

Supervisors:

Doctoral Researcher Ali Salmasi

Professor Tomi Westerlund

June 2026

Turku

Master of Science in Technology Thesis

Subject: Computer Vision and Robotics

Author(s): Lipei Huo

Title: Self-Supervised Visual Representation Learning with Masked Autoencoders for Traversability Estimation in Outdoor Environments

Supervisor(s): Doctoral Researcher Ali Salmasi, Professor Tomi Westerlund

Number of pages: 67 pages

Date: June 2026

Abstract.

Autonomous robots operating in unstructured outdoor environments require reliable perception systems to assess traversability and navigate safely. However, gathering large amounts of labeled data to train these perception models is time-consuming and expensive. Self-supervised learning has recently emerged as a promising approach for leveraging large amounts of unlabeled data to learn useful visual representations.

In this thesis, we use Masked Autoencoders (MAE) [1] for self-supervised visual pretraining to support traversability prediction. A domain-adaptive approach is applied, where a pretrained MAE model is further fine-tuned on unlabeled outdoor images to learn representations suitable for robotic navigation. The trained encoder is then adapted for downstream traversability tasks using both patch-level classification and pixel-level segmentation models. We also explore multi-scale feature aggregation and the fusion of LiDAR-derived depth data to improve the understanding of terrain geometry.

All experiments are carried out using the GOOSE dataset, which contains a variety of outdoor environments, including forests, grasslands, uneven terrain, and roads. The models are evaluated using standard metrics such as F1-score, precision, and recall. The results show that self-supervised pretraining improves performance when labeled data is limited and helps the model efficiently learn useful terrain features. At the same time, the experiments highlight a limitation of vision transformer-based encoders: patch-based tokenization can make it more difficult to preserve fine spatial details.

Overall, the experiments demonstrate the potential of self-supervised visual representation learning for traversability estimation in robotics and provide insights into important design considerations when using transformer-based architectures for perception tasks.

Keywords: self-supervised learning, traversability estimation, autonomous navigation, outdoor robotic perception, vision transformers, masked autoencoders, semantic segmentation.

Table of contents

1 Introduction

1.1 Motivation and problem statement

Autonomous robots are increasingly being used in real-world applications that require reliable navigation. Examples include agricultural robots operating on large farms, search-and-rescue robots in disaster zones, logging robots in forests, and self-driving vehicles navigating complex roads. In many of these applications, robots operate in outdoor environments where the terrain can vary significantly and remain unpredictable. Unlike indoor or structured environments, outdoor areas contain natural obstacles, uneven surfaces, and continuously changing conditions. As a result, enabling robots to navigate safely through such environments remains a significant challenge in robotics research.

A key component of autonomous navigation is determining which areas are safe to traverse, a task known as traversability estimation. Traversability estimation refers to the process of determining whether a region of terrain is safe for a robot to cross. This is often represented using traversability maps [2], where each region is classified as either traversable or non-traversable. In practice, robots must continuously analyze sensor data to determine where safe navigation is possible, where paths may be blocked, and where the terrain may be unstable. For example, mud and gravel may appear visually similar, yet misclassifying them could cause a robot to become stuck. Rocks, slopes, vegetation, and steep terrain can also present risks depending on the robot’s capabilities. Therefore, accurate traversability estimation is essential for safe autonomous navigation.

Estimating traversability in outdoor environments is highly challenging. Natural terrain is often unstructured and visually complex, and many areas contain weak visual cues. Unlike city streets or indoor floors, outdoor surfaces do not follow consistent patterns, so visual appearance alone is not always sufficient to determine whether they are safe to cross. For example, grass may conceal rocks, muddy ground can appear similar to firm soil, and potholes on roads can be difficult to detect. In addition, outdoor perception is affected by changing lighting conditions, shadows, weather, and seasonal variations. All of these factors make it more difficult for robots to accurately understand their surroundings, and therefore developing perception

systems that can robustly handle these challenges remains an active research problem.

Traditional methods for traversability estimation often rely on supervised deep learning models, particularly convolutional neural networks [3], trained on manually labeled data. Even with annotation tools, the labeling process remains time-consuming. While these methods can perform well in many perception tasks, they require large amounts of labeled training data. In robotics, collecting such data is expensive, especially for pixel-level image annotations or point-level labels for point clouds. Robotic datasets are also generally much smaller than the large-scale datasets commonly used in computer vision research, making it difficult for supervised models to generalize to new environments. These limitations highlight the need for alternative learning approaches.

Self-supervised learning first demonstrated major success in natural language processing [4] and has since become increasingly popular in computer vision, particularly following the introduction of Vision Transformers (ViTs) [5], which represented a significant advancement in visual representation learning. Unlike traditional supervised learning, self-supervised methods do not require labeled data; instead, they learn useful features from raw data by solving auxiliary tasks. This is particularly beneficial for robotics, where robots naturally collect large amounts of sensor data that is mostly unlabeled.

In this thesis, we focus on transformer-based self-supervised learning for traversability estimation in unstructured outdoor environments. We explore Masked Autoencoders (MAE) [1], a self-supervised transformer framework that learns visual features by reconstructing masked regions of input images. We use a pretrained MAE encoder and adapt it for traversability prediction using both patch-level classification and pixel-level segmentation models. To improve the model’s understanding of spatial structure and geometry, we also investigate additional components such as multi-scale feature aggregation and LiDAR-based depth fusion.

Through extensive experiments on the GOOSE dataset, our work evaluates how effectively self-supervised transformer representations can support safe robot navigation in complex unstructured outdoor environments.

1.2 Challenges

Estimating traversability in real-world outdoor environments is challenging for perception systems. Natural terrain is often irregular and visually complex, with many areas providing unclear visual cues. In addition, modern deep learning models have their own limitations in capturing fine details, which are important for accurately identifying safe and unsafe areas. This section highlights the main challenges involved in traversability estimation in unstructured outdoor environments.

Environmental Complexity

Outdoor terrain can vary significantly in its shape and physical properties, and traversability depends on many factors such as slope, surface roughness, material type, and obstacles. For example, grass, gravel, and mud may appear visually similar, yet they behave very differently when a robot moves over them. Small rocks, uneven ground, and slopes can also create hazards that are difficult to detect using only visual information. Therefore, perception systems must capture both geometric details and global contextual information to reliably make traversability decisions.

Visual Ambiguity in Natural Environments

Natural outdoor scenes often contain ambiguous visual patterns. Traversable and non-traversable areas may appear mixed together, with narrow paths and irregular boundaries. Thin structures such as tree branches or small gaps in the terrain are easy to overlook, and some regions may contain multiple terrain types within a small area. All of these factors make it difficult for perception models to accurately distinguish between safe and unsafe regions.

Spatial Precision in Deep Learning Models

Modern deep learning models, particularly transformers, face additional challenges when applied to tasks such as pixel-level segmentation. Vision transformers divide an image into fixed-size patches that are treated as tokens, while fine details within each patch are not explicitly preserved. Although this patch-based approach enables efficient global scene understanding, it also reduces spatial resolution. As a result, small obstacles or sharp boundaries between terrain types may be lost. This makes it more difficult for the model to generate accurate traversability predictions.

These challenges require perception systems capable of learning robust representations of both global scene structure and local terrain details. In this thesis, we explore self-supervised representation learning using transformer-based architectures to improve the robustness of traversability estimation in complex outdoor environments.

1.3 Research questions

The goal of this thesis is to investigate how self-supervised visual representation learning can support traversability estimation in unstructured outdoor environments. In particular, this work examines whether a transformer-based model pretrained using Masked Autoencoders (MAE) [1] can reduce the dependence on large labeled datasets while maintaining strong performance in terrain perception tasks. To address this goal, the following research questions are investigated:

RQ1: Can a self-supervised visual representation learning framework based on a transformer encoder and a lightweight classification head be effectively developed for outdoor traversability estimation?

RQ2: How effectively can pretrained transformer representations be adapted to both patch-level traversability classification and pixel-level traversability segmentation tasks?

RQ3: How well do the proposed transformer-based models perform on the GOOSE dataset, which contains diverse unstructured outdoor environments with pixel-level annotations?

RQ4: How do the proposed transformer-based methods compare with conventional supervised convolutional neural network architectures, such as U-Net and DeepLabv3, for traversability estimation?

1.4 Contributions

In this thesis, we explore the use of self-supervised visual representation learning for traversability estimation in unstructured outdoor environments. The main contributions of this work are summarized as follows:

1. Self-supervised Representation Learning for Robotic Perception

A self-supervised learning pipeline based on Masked Autoencoders (MAE) is developed to learn visual features from large amounts of unlabeled images. This approach reduces the need for expensive labeled datasets while enabling robust feature learning for robotic perception tasks.

2. Transformer-based Traversability Prediction Models

   Several models for traversability prediction are designed using pretrained transformer weights, including patch-level classification and pixel-level segmentation models. Additional architectural improvements, such as multi-scale supervision, feature pyramid structures, and LiDAR fusion, are explored to enhance spatial representation.

3. Evaluation on Real-world Outdoor Datasets

   The proposed models are evaluated on the GOOSE dataset, which covers diverse outdoor environments. The experiments assess model performance under different architectural configurations and training settings.

4. Comparison with CNN-based Segmentation Baselines

   The transformer-based models are compared with commonly used CNN architectures, including U-Net and DeepLabv3, providing insight into the strengths and weaknesses of transformer representations for terrain perception.

5. Analysis of Spatial Limitations in Patch-based Transformer Encoders

   The limitations of patch-based transformer models for fine-grained traversability estimation are analyzed, particularly in detecting small obstacles and thin terrain features. The results highlight the trade-offs between global semantic understanding and spatial precision in transformer-based perception models.

1.5 Thesis organization

The remainder of this thesis is organized as follows.

Chapter 2 reviews related work on traversability estimation, robotic perception, and self-supervised visual representation learning.

Chapter 3 introduces the datasets used in this work, including the GOOSE dataset, and describes the preprocessing steps applied to the data.

Chapter 4 presents the methodology, covering the MAE architecture and the different traversability prediction models developed in this thesis.

Chapter 5 describes the experimental setup, presents the results of the proposed approaches, and compares them with baseline models such as U-Net and DeepLabV3.

Chapter 6 discusses the experimental findings by analyzing the strengths and limitations of the proposed methods, as well as the effects of key architectural design choices.

Chapter 7 concludes the thesis and outlines potential directions for future work.

2 Related work

2.1 Traversability Estimation and CNN-based Methods

Traversability estimation is an important task in mobile robotics. The goal is to identify which parts of the environment are safe for a robot to traverse. This is particularly important in unstructured outdoor environments where there are no clearly defined roads or layouts. Unlike urban driving environments, off-road areas often contain uneven terrain such as grass, rocks, mud, and slopes, all of which make navigation more challenging. A reliable traversability system can help robots avoid obstacles, select safe paths, and operate safely in real-world conditions.

Early methods for traversability estimation were based on classical computer vision techniques. These methods typically rely on color, texture, or geometric features extracted from images or depth sensors to classify different types of terrain [6]. Although these methods can be computationally efficient, they do not generalize well to new environments because they depend on hand-crafted features.

In recent years, deep learning has significantly improved visual perception tasks. Convolutional neural networks (CNNs) [3] have become the dominant approach and are used not only for object detection but also for tasks such as semantic segmentation and traversability estimation. CNN-based models can learn complex features from data while capturing both low-level details and high-level semantic information. Popular architectures such as U-Net [7] and DeepLabv3 [8] are widely used for pixel-level segmentation tasks.

U-Net is an encoder–decoder CNN architecture originally developed for biomedical image segmentation [7]. It consists of two main parts: a contracting path that captures global context and an expanding path that recovers spatial details. One of the main features of U-Net is its skip connections, which transfer high-resolution features from earlier layers to later layers. This helps combine detailed spatial information with deeper semantic features, enabling the model to preserve fine details while still understanding the overall scene. Its design is relatively simple, yet its performance is strong, making it a widely used baseline model for segmentation tasks.

DeepLabv3 uses dilated (atrous) convolutions and Atrous Spatial Pyramid Pooling (ASPP) to capture information at multiple scales [8]. This enables the model to handle objects of different sizes and improves segmentation performance in complex scenes involving fine details. DeepLab-based models generally achieve better results than simpler architectures; however, they also require more computational resources and careful tuning.

More recently, foundation models and transformer-based architectures have also been explored for traversability estimation. In particular, self-supervised vision transformers such as DINOv2 [9] have demonstrated strong visual representation learning capabilities, where features learned from large-scale unlabeled data can be effectively transferred to downstream perception tasks, including segmentation and robotics applications. By leveraging large-scale pretraining, these models can learn general visual representations that transfer effectively to outdoor navigation tasks while reducing dependence on large labeled datasets. Recent work has demonstrated that adapting vision foundation models to unstructured outdoor environments can achieve reliable traversability estimation performance [10].

Although CNN-based methods achieve strong performance, they depend on large amounts of pixel-level labeled data. In robotics, particularly in outdoor environments, collecting and annotating such datasets is expensive. Another limitation is that CNN-based models typically require labeled data for each new domain, which reduces their scalability. These challenges motivate the need for alternative approaches that can better utilize unlabeled data and scale more effectively.

2.2 Self-Supervised Learning for Vision

Self-supervised learning has become increasingly popular for learning visual representations without requiring large amounts of labeled data. These methods learn useful features directly from raw data through the use of pretext tasks. This is particularly useful in robotics, where large amounts of sensor data are available but labeling is expensive and time-consuming.

In recent years, transformer-based models have demonstrated strong performance in visual representation learning. Several self-supervised learning methods have been developed for vision transformers, including contrastive learning and

reconstruction-based approaches. Contrastive learning methods, such as SimCLR [11] and MoCo [12], train models by encouraging different augmented views of the same image to produce similar representations while distinguishing them from representations of other images. These methods achieve strong performance but usually require large batch sizes and careful tuning. Another approach is masked image modeling, in which parts of the input image are hidden and the model learns to reconstruct the missing patches. Masked Autoencoders (MAE) [1] are a representative example of this approach. In MAE, an image is divided into fixed-size patches, which are then converted into tokens and processed by a transformer encoder.

Self-supervised learning provides several advantages for robotic perception. First, it reduces the dependence on labeled data, making it easier to scale to large datasets. Second, the learned features can be transferred to other tasks, such as segmentation or navigation, with only limited fine-tuning. Third, self-supervised models are generally more robust to environmental changes, which is important for outdoor robotics applications.

However, self-supervised transformer models also present several challenges. Because images are processed as patches, the spatial resolution is reduced, making it more difficult to detect fine details such as precise boundaries. In addition, pretraining objectives such as masked patch reconstruction do not explicitly teach the model to recognize semantic structures or edges. As a result, additional architectural modifications are often required to achieve strong performance on pixel-level tasks.

In our experiments, we use a self-supervised pretrained MAE model as the backbone for traversability estimation. The pretrained encoder is used to extract general visual features, which are then adapted to the pixel-level traversability task using a limited amount of labeled data. This approach combines the flexibility of self-supervised learning with the accuracy of supervised fine-tuning.

2.3 Transformer-based Vision Models for Segmentation

Transformer-based models have become increasingly popular in computer vision, especially for tasks that require scene understanding. Originally developed for natural language processing, transformers use self-attention mechanisms to model

relationships between elements in a sequence. When applied to images, Vision Transformers (ViTs) [5] treat an image as a sequence of patches, allowing the model to capture long-range relationships across the entire scene.

Unlike convolutional neural networks, which rely on local receptive fields, transformers can directly model interactions between distant regions of the image. This is particularly useful for scene understanding tasks. As a result, transformers have achieved strong performance in tasks such as image classification and semantic segmentation. Several transformer architectures have been specifically designed for dense prediction tasks. One early example is the Vision Transformer (ViT) [5], which demonstrated that a pure transformer architecture can compete with CNNs on large-scale image classification tasks. However, ViT does not include mechanisms for preserving spatial resolution, which can limit its ability to capture fine details.

To address this issue, hybrid and hierarchical transformer models have been introduced. For example, models such as SegFormer [13] and Swin Transformer use hierarchical feature representations to capture multi-scale features. This approach improves spatial localization and segmentation performance by combining global attention with local feature processing. Another popular direction involves adapting pretrained transformer encoders for specific segmentation tasks. In this approach, the transformer encoder is first pretrained using self-supervised learning and then paired with a task-specific decoder to generate pixel-level predictions [14]. The decoder is designed to restore spatial resolution through feature upsampling and fusion.

Despite these improvements, transformer models still face challenges in segmentation tasks. One major limitation is the patch-based representation of images. For example, when a patch size of 16×16 is used, each token represents a relatively large region of the input image. As a result, fine spatial details may be lost, making it difficult to accurately predict small objects and sharp boundaries. Therefore, transformer-based segmentation models can struggle in tasks that require precise localization.

The self-attention mechanism in transformers aggregates information across the entire image. While this helps the model understand global context, it can also blur

boundaries between different regions. Consequently, the model may struggle to distinguish small areas, such as boundaries between traversable and non-traversable terrain. To address this issue, several techniques have been proposed, including multi-scale feature aggregation [15], skip connections from earlier layers, and hybrid architectures. Feature Pyramid Networks (FPN) [16] and multi-scale supervision are commonly used strategies for improving spatial detail and segmentation performance.

In our experiments, we use a transformer-based encoder pretrained with a Masked Autoencoder (MAE) [1] as the foundation for traversability estimation. To overcome the limitations of patch-based representations, we incorporate architectural components such as multi-scale supervision and feature pyramid decoding. These modifications are designed to help the model capture finer spatial details while still leveraging the strong semantic representations learned by the transformer encoder.

2.4 Multimodal Perception and LiDAR Fusion

In outdoor robotic perception systems, relying solely on RGB images can be challenging because changes in lighting conditions, shadows, and texture variations can make scene interpretation difficult. For this reason, combining camera and LiDAR data [17] has become a widely used approach.

LiDAR sensors measure the three-dimensional structure of the environment by capturing distances to surrounding objects. This geometric information is highly useful for tasks such as terrain analysis and traversability estimation. Unlike RGB images, LiDAR provides explicit depth information, helping the system better understand the physical structure of the scene.

A common approach for combining both sensors is to project the LiDAR point cloud onto the image plane, creating a depth map aligned with the RGB image. This allows the model to utilize both visual and geometric information for each region of the scene. However, LiDAR data is often sparse, particularly for small obstacles, which makes multimodal fusion more challenging.

Several fusion strategies have been proposed for incorporating LiDAR data, including heterogeneous CNN–Transformer based fusion approaches that improve cross-modal semantic alignment and depth understanding [18]. Early fusion

combines different data modalities at the input level, for example by stacking RGB and depth channels into a single input representation [19]. This approach is simple and allows the model to learn from both modalities from the beginning. However, it may not effectively capture the differences between RGB and depth information. Mid-level (feature-level) fusion processes each modality separately using different encoders and then combines their features at an intermediate stage [20]. This allows the model to first learn meaningful features from each modality before merging them, which often leads to improved performance. Fusion can be performed in several ways, including addition, concatenation, or attention-based methods. Late fusion combines the outputs of separate models trained on different modalities [21]. However, this approach generally cannot capture detailed interactions between modalities and is less suitable for tasks requiring precise spatial alignment. In transformer-based models, fusion can also be performed at the token level. For example, RGB and depth images can be divided into patches and combined as tokens. However, this increases computational cost and can make training more difficult.

Although multimodal fusion offers several advantages, it also introduces additional challenges. First, the sensors must be carefully calibrated to ensure accurate alignment between LiDAR and camera data. Misalignment can significantly reduce performance. Second, LiDAR data is often sparse and may fail to capture small details such as thin objects. As a result, small obstacles and fine boundaries still rely heavily on visual information from RGB images. Therefore, an effective fusion strategy must balance the strengths of both modalities.

In our experiments, LiDAR information is incorporated into the pipeline using feature-level fusion. The RGB image is first processed by a pretrained transformer encoder, while the LiDAR-derived depth information is introduced at intermediate stages of the segmentation decoder. This approach preserves the strong visual representations learned by the encoder while incorporating geometric information to improve spatial understanding. We compare the LiDAR fusion model with RGB-only models to evaluate the effectiveness of multimodal fusion.

2.5 Datasets for Outdoor Navigation

In recent years, many datasets have been developed to support research in areas such as visual perception, localization, and navigation. These datasets typically include sensor data such as RGB images, LiDAR point clouds, and annotations for tasks including object detection, semantic segmentation, and traversability estimation.

One of the most widely used datasets in autonomous driving is the KITTI dataset [22]. It provides synchronized data from multiple sensors, including stereo cameras, LiDAR, and GPS/IMU, collected in urban driving environments. KITTI has been widely used for tasks such as object detection and road segmentation. However, it primarily focuses on structured environments with clearly defined roads and traffic rules, making it less suitable for unstructured outdoor navigation.

Another widely used dataset is the Oxford RobotCar dataset [23]. It contains long-term recordings of urban driving under different weather conditions. The dataset also includes multiple sensor modalities and is useful for research in localization and perception. Similar to KITTI, this dataset mainly focuses on urban environments and does not fully represent off-road terrain.

For perception tasks in unstructured outdoor environments, more specialized datasets have been developed. These environments include forests, grasslands, and uneven terrain, where there are no clearly defined road boundaries. In such environments, traversability estimation becomes more difficult because of the large variation in terrain appearance and geometry. The GOOSE dataset [24] is specifically designed for perception tasks in these environments. It includes a wide range of images captured in natural settings, such as grass, soil, gravel, rocks, and vegetation. In addition to RGB images, the dataset also provides LiDAR data, supporting multimodal research. It further includes pixel-level traversability annotations, making it useful for training and evaluating navigation models.

The GOOSE dataset is particularly challenging because the environments are less structured and more visually ambiguous. For example, surfaces such as grass and mud may appear visually similar while having very different traversability properties.

In addition, environmental factors such as lighting changes and shadows further increase the difficulty of the task.

In our experiments, the GOOSE dataset is used as the primary dataset for both self-supervised pretraining and supervised fine-tuning. Since our work focuses on unstructured outdoor environments, this dataset enables a more realistic evaluation of traversability models in real-world robotic scenarios.

2.6 Summary and Research Gap

In this chapter, we reviewed different approaches to traversability estimation and visual perception, including CNN-based methods, self-supervised learning, transformer models, and multimodal perception. These methods have achieved significant progress; however, several limitations still remain, particularly in outdoor robotic navigation.

CNN-based segmentation models such as U-Net and DeepLabv3 perform well in pixel-level tasks. However, they rely heavily on large amounts of labeled data, which limits their applicability in real-world robotics. Collecting and annotating data in outdoor environments is expensive, making fully supervised methods less practical.

Self-supervised learning, particularly transformer-based models such as Masked Autoencoders, offers a promising alternative. These methods can learn useful representations from unlabeled data, improving generalization capabilities. However, most existing work focuses primarily on representation learning or image reconstruction rather than directly addressing tasks such as traversability segmentation.

Transformer-based models are also highly effective at capturing global context and understanding scene structure. However, their patch-based design reduces spatial resolution, making it more difficult to detect fine details such as small obstacles and sharp boundaries. This can be a major limitation in outdoor environment segmentation tasks, where accurate localization is essential.

Multimodal approaches attempt to address some of these limitations by incorporating additional sensor data, such as LiDAR. Although depth information provides useful geometric cues, effectively combining it with RGB data remains challenging. Issues

such as sensor misalignment, sparse LiDAR measurements, and differences between modalities make fusion difficult. In many cases, existing fusion methods do not provide substantial improvements in fine-detail segmentation.

From the discussion above, several research gaps can be identified. First, there is a need for methods that combine the efficiency of self-supervised learning with the strong task-specific performance required for traversability estimation. Second, transformer-based models often struggle to preserve fine spatial details, which are important for accurately identifying small regions. Third, it remains unclear to what extent multimodal fusion can improve traversability prediction in unstructured outdoor environments, requiring further investigation.

To address these gaps, we propose a framework that uses a pretrained transformer encoder with self-supervised learning combined with task-specific heads for traversability segmentation. We explore different architectural designs, including patch-level classification, pixel-level segmentation, multi-scale supervision, feature pyramid decoding, and LiDAR fusion. The main objective is to investigate how self-supervised transformer representations can be adapted to improve traversability estimation while minimizing the dependence on labeled data.

3 Dataset & problem formulation

3.1 GOOSE Dataset

Our experiments in this thesis are conducted using the GOOSE dataset [24], which is specifically designed for perception tasks in unstructured outdoor off-road environments. Unlike traditional autonomous driving datasets that focus on structured urban roads, the GOOSE dataset contains images captured in natural environments such as forests, grasslands, bushes, and uneven terrain. These environments present additional challenges for robotic navigation due to the absence of clear road boundaries and the presence of visually ambiguous surfaces.

Most of the images in the GOOSE dataset are high-resolution RGB images, typically with resolutions of 1024×500 or 2048×1000, with these two resolutions being the most common. These images capture a wide variety of terrain classes, including sky, terrain, animal, water, road, construction, human, sign, vehicle, and vegetation. The dataset also provides pixel-level annotated masks for each image and LiDAR point cloud data, enabling multimodal perception experiments. Table 3.1 shows the classes in the GOOSE dataset, and Figure 3.1 presents example RGB images together with their corresponding pixel-level annotations.

Our traversability estimation task is formulated as a binary classification problem at both the patch level (16 × 16) and the pixel level, where each patch or pixel is classified as either traversable or non-traversable. In our experiments, the classes gravel, bikeway, pedestrian_crossing, road_marking, sidewalk, asphalt, cobble, snow, soil, low_grass, and leaves are considered traversable. These categories are selected based on their suitability for safe robot navigation in outdoor environments, although the set of traversable classes may vary depending on the size of the robot.

The GOOSE dataset is well suited to our task because it provides diverse outdoor scenarios as well as pixel-level annotations relevant to traversability estimation. It reflects many of the challenges encountered in real-world robotic applications.

Table 3.1: classes in GOOSE dataset.

Group	Class Labels
Animal	animal
Construction	bridge, building, container, debris, fence, guard_rail, tunnel, wall, wire
Human	person, rider
Object	barrel, obstacle, pipe, pole, rock, street_light,
Road	bikeway, curb, pedestrian_crossing, rail_track, road_marking, sidewalk
Sign	barrier_tape, misc_sign, traffic_cone, traffic_light, traffic_sign, road_block, boom_barrier
Sky	sky
Terrain	asphalt, cobble, gravel, soil, snow
Vegetation	bush, crops, forest, hedge, high_grass, leaves, low_grass, moss, scenery_vegetation, tree_crown, tree_root, tree_trunk
Vehicle	bicycle, bus, car, caravan, heavy_machinery, kick_scooter, military_vehicle, motorcycle, on_rails, trailer, truck
Void	ego_vehicle, outlier, undefined
Water	water

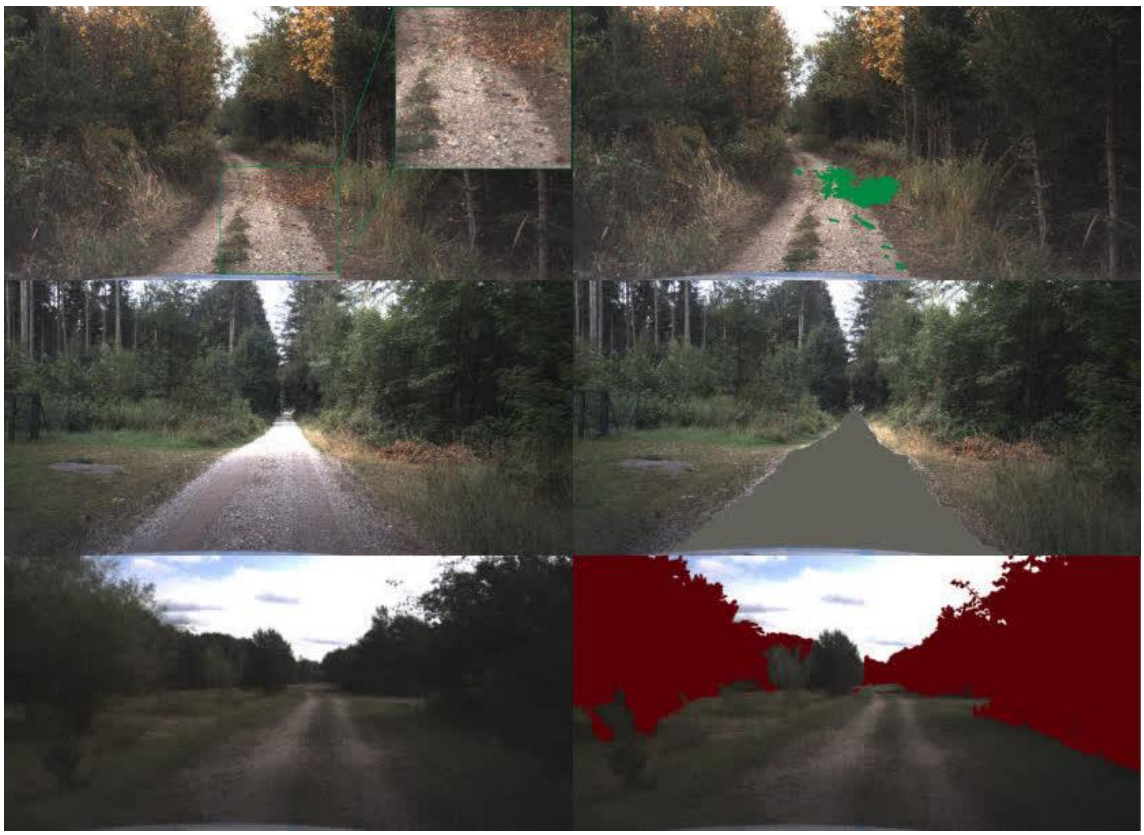

Figure 3.1: Example images from the GOOSE dataset showing different terrain types. The left column displays the original RGB image, while the right column shows the corresponding pixel-level annotations. Top to bottom: Leaves, Gravel, and Forest.

3.2 Dataset splits

The GOOSE dataset, which contains sensor data collected from the MuCAR-3 platform, is divided into training, validation, and test subsets to support both self-supervised pretraining and supervised evaluation. The GOOSE Train and GOOSE-Ex Train splits are used for self-supervised pretraining, where no labels are required. This enables the model to learn general visual representations from a large amount of unlabeled data. The GOOSE-Ex dataset, designed for fine-tuning on specific platforms, includes data collected from two additional platforms (an excavator and a quadruped robot) operating in different environments. For LiDAR-based experiments, the corresponding LiDAR point clouds are synchronized with the selected image samples using their timestamps.

For supervised traversability learning, a representative subset consisting of approximately 1,350 labeled training images and 150 validation images is initially used during model development and ablation studies. This subset corresponds to approximately 10% of the available labeled data. The subset is selected by preserving the relative proportion of images from each recording sequence in the original dataset, ensuring that the distribution of environments, terrain types, and operating conditions remains consistent with the full dataset. As a result, the subset provides a representative benchmark for comparing different model variants while significantly reducing computational cost during experimentation.

The 10% subset is primarily used to evaluate the proposed architectures, including the patch-level classification model, pixel-level segmentation model, multi-scale supervision model, feature pyramid model, and LiDAR fusion model. All models are trained and evaluated using the same subset and validation protocol, ensuring fair comparison between architectural variants.

After identifying the best-performing proposed model, additional experiments are conducted using larger portions of the labeled training data (30%, 50%) for comparison with baseline segmentation models, namely U-Net and DeepLabv3. These experiments are designed to evaluate the scalability and data efficiency of the proposed approach under different amounts of supervision. The validation and test protocols remain unchanged across all experiments to ensure a consistent evaluation setting.

The separation between self-supervised pretraining data and supervised evaluation data helps prevent overfitting to the labeled subset and reflects a realistic robotics scenario, where large quantities of unlabeled sensor data are available while labeled data remains limited.

The selection of the 10% labeled subset was guided by considerations of representativeness and research integrity. By preserving the relative proportions of images from each recording sequence, the subset maintains a distribution of environments and terrain categories that closely reflects the full dataset, thereby reducing potential sampling bias. The GOOSE dataset was used in accordance with its publicly available release terms. Furthermore, the implementation details, training procedures, and experimental settings are fully documented to support reproducibility and enable independent verification of the presented results.

3.3 Preprocessing

All input images are randomly cropped to 224 × 224 pixels and randomly flipped horizontally before being fed into the model. This resolution is chosen to match the input requirements of the Masked Autoencoder (MAE) [1] architecture, where images are divided into 16 × 16 patches, resulting in a 14 × 14 token representation.

During the initial experiments, we used the same data augmentation strategy as in the original MAE training, where images were randomly cropped with a scale range between 0.2 and 1.0 before being resized. However, this approach resulted in poor downstream performance, even after adjusting the scale range to 0.6–1.0. Further analysis revealed that this issue was caused by differences in image resolution between the ImageNet [26] dataset and the GOOSE dataset. While ImageNet images typically range from 300 to 600 pixels in size, resizing them to 224 × 224 pixels after cropping with a scale between 0.2 and 1.0 results in relatively little distortion. In contrast, GOOSE images are considerably larger, with most images having resolutions of 2048 × 500 or 1024 × 500. Consequently, aggressive cropping and resizing of these larger images lead to a significant loss of fine-grained texture and edge information. For example, directly resizing an image with a resolution of 2048 × 500 results in a substantial reduction in the preservation of texture and fine image details.

Therefore, we adopted a fixed 224 × 224 cropping strategy with random crop locations. This approach is better aligned with the characteristics of robotic image inputs, as the visual variation encountered by robotic platforms is generally lower than that found in ImageNet. For LiDAR data, we focused on increasing the density of captured points by assigning greater weight to horizontal center crops, where the LiDAR point clouds are denser. In addition, cropping was biased toward the lower central regions of the image, as the upper region often contains sky, which is not relevant for traversability estimation. This improves training efficiency by focusing on the most informative regions. Figure 3.2 shows the LiDAR depth point cloud projected onto the image, highlighting the increased point density in the horizontal center region. This observation demonstrates the importance of adapting preprocessing strategies to the characteristics of the dataset.

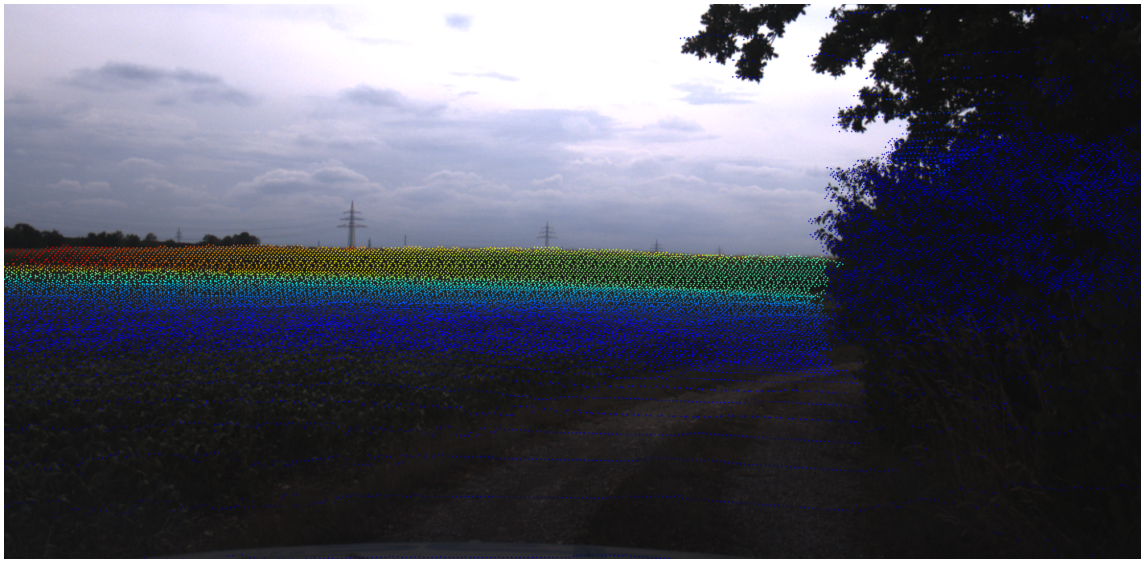

Figure 3.2: LiDAR Depth Point Cloud Overlay on Image, Highlighting Increased Point Density in the Horizontal Center Crop, Different depths are shown in different colours.

3.4 Traversable area definition

Traversability is defined as the ability of a robot to safely navigate across a given surface. In our experiments, traversability is formulated as a binary classification problem, where each patch or pixel is labeled as either traversable or non-traversable.

The labeling is based on the surface categories provided in the GOOSE dataset annotations. Surfaces such as gravel, asphalt, soil, snow, and low grass are

considered traversable, while obstacles such as rocks, forests, and structural barriers are considered non-traversable.

For patch-level classification, labels are assigned by aggregating pixel-level annotations. A patch is labeled as traversable if more than 50% of its pixels belong to traversable categories. This labeling strategy simplifies the classification task but may introduce ambiguity in regions containing mixed terrain types. At the same time, it reflects practical navigation requirements, where robots must make decisions based on both the semantic and geometric properties of the environment. Figure 3.3 shows a sample labeled image from the dataset, illustrating how traversable and non-traversable regions are annotated at the pixel level.

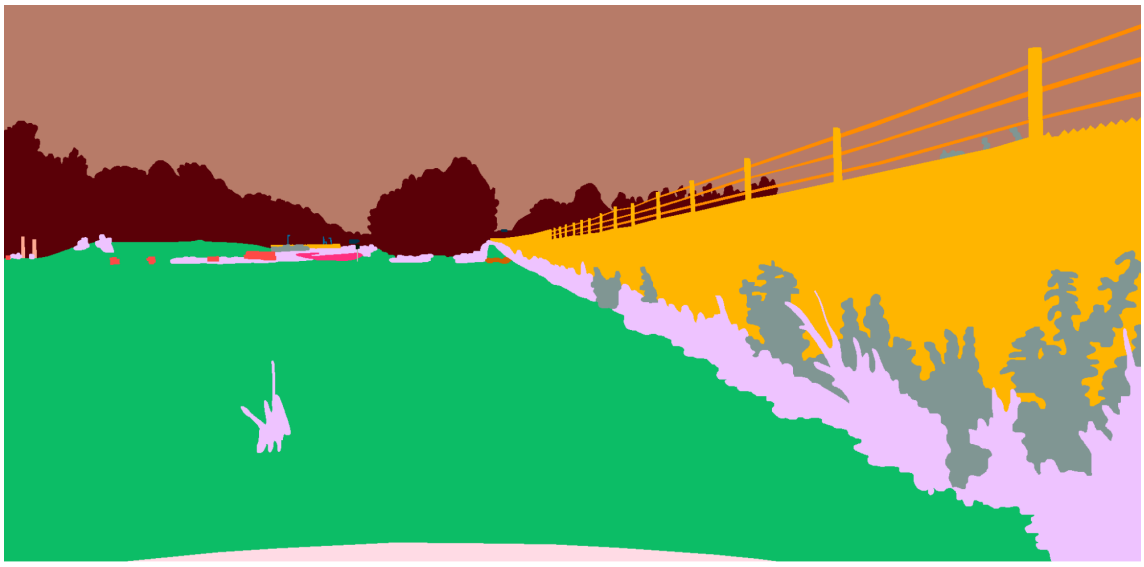

Figure 3.3: Sample labeled image showing traversable and non-traversable regions, with pixel-level annotations. traversable areas marked in green, illustrating terrain low grass.

3.5 Lidar data

In our experiments, LiDAR data is used to provide geometric information about the environment. LiDAR sensors capture 3D point clouds, which can be projected onto the image plane to generate depth maps aligned with the RGB images. This projection requires accurate calibration between the LiDAR and camera sensors, including both intrinsic and extrinsic parameters [25]. The resulting depth map has the same spatial resolution as the RGB image, enabling pixel-wise fusion.

However, LiDAR data is inherently sparse. Many pixels in the projected depth map do not contain valid depth values, particularly for distant or small objects. As a result, depth information can be incomplete or noisy, especially for thin structures such as tree stems.

The depth maps are normalized and integrated into our model through feature fusion. The RGB image is processed by the transformer encoder, while depth information is incorporated at intermediate stages of the segmentation decoder. This approach allows the model to leverage both semantic and geometric information.

Although LiDAR data does not always improve performance in fine-grained segmentation tasks, it provides valuable geometric cues. Due to its sparsity, small obstacles may not be captured reliably, which can limit its contribution in complex scenarios.

4 Methodology

4.1 Masked Autoencoder Architecture

Masked Autoencoders (MAE) [1] are an effective approach for self-supervised visual representation learning using transformer-based encoder–decoder architectures. In this work, the MAE framework is adopted as the backbone encoder for traversability estimation due to its ability to learn strong semantic representations from unlabeled data.

In MAE, given an input RGB image of size 224 × 224, the image is first divided into non-overlapping patches of size 16 × 16. Each patch is then flattened and projected into a latent embedding, forming a sequence of tokens. For a 224 × 224 image, this results in a total of 14 × 14 = 196 tokens. A diagram illustrating the overall architecture of the MAE framework is shown in Figure 4.1, highlighting the processes of patch tokenization, masking, and encoding.

During self-supervised training, a large portion of the tokens (typically 75%) is randomly masked. Only the remaining visible tokens are passed through the transformer encoder. This design significantly reduces computational cost while forcing the model to learn meaningful representations from partial observations. The encoder consists of multiple transformer blocks (24 blocks in the large model), each containing self-attention and feed-forward layers that enable global context modeling across different regions of the image.

A lightweight transformer-based decoder consisting of eight transformer blocks is then used to reconstruct the masked image patches. The decoder takes both the encoded visible tokens and mask tokens as input and predicts the original pixel values of the missing patches. The training objective is defined as a pixel-wise reconstruction loss computed over the masked regions, where the reconstructed pixels are compared with the corresponding ground-truth image pixels.

It is important to note that the MAE model is optimized to minimize pixel-level reconstruction error rather than perceptual image quality. As a result, the reconstructed images may appear visually blurry even when the reconstruction loss is low. However, for downstream tasks such as traversability estimation, the primary

objective is not image reconstruction but the quality of the learned feature representations. Therefore, in our task, the encoder is used as a feature extractor, while the decoder is discarded during downstream training. The feature representations produced by the MAE encoder are directly fed into the classification head instead of being processed by the MAE decoder.

The MAE framework is particularly well suited for robotic perception tasks. By learning from large-scale unlabeled image data, it can capture general visual structures such as textures, edges, and spatial relationships. These properties are essential for understanding outdoor environments, where semantic labels are limited and visual variability is high.

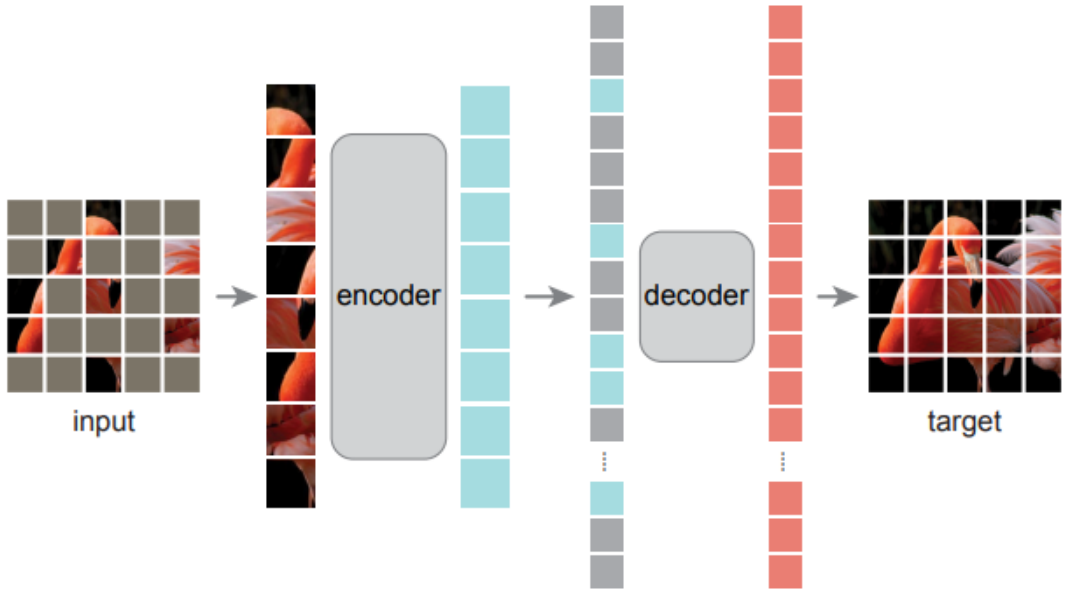

Figure 4.1: Overview of the Masked Autoencoder (MAE) architecture, showing the process: Image → Patchify → Mask → Encoder → Decoder → Reconstruction (adapted from MAE: Masked Autoencoders Are Scalable Vision Learners, He et al., 2021).

4.2 Self-Supervised Pretraining

To take advantage of self-supervised learning, we performed domain-adaptive pretraining using the MAE framework [1]. The model was first initialized with pretrained weights obtained from large-scale training on the ImageNet dataset [26], which contains approximately 1.2 million images with diverse visual content. This provided the model with strong general visual representations.

However, ImageNet images differ significantly from the outdoor robotic scenes used in this work. ImageNet contains a wide variety of objects and textures, whereas our target dataset primarily consists of outdoor environments such as roads, grass, soil, and vegetation, with scenes often dominated by sky and grass. To reduce this domain gap, additional self-supervised pretraining was performed using unlabeled images from the GOOSE dataset.

During this stage, the MAE encoder and decoder were trained jointly using the same reconstruction objective described in Section 4.1 and a very low learning rate. This allowed the model to adapt its learned representations to outdoor environments by capturing terrain textures, natural lighting conditions, and environmental structures while retaining the knowledge acquired from ImageNet.

An important aspect of this pretraining stage is the image scaling and augmentation strategy. The original MAE training procedure uses random cropping with a scale range between 0.2 and 1.0, followed by resizing the cropped images to 224 × 224. While this approach works well for ImageNet, it is less suitable for the GOOSE dataset. GOOSE images are considerably larger (for example, most images have resolutions of 1024 × 500 or 2048 × 500), and aggressive cropping and resizing can remove important texture and edge information. To address this issue, random scaling was removed. This modification helps preserve fine-grained details that are important for traversability estimation.

The overall training process consists of two stages, as illustrated in Figure 4.2.

The first stage is domain-adaptive self-supervised pretraining. In this stage, the MAE model is initialized with ImageNet-pretrained weights, and the encoder and decoder are trained on unlabeled GOOSE images. The objective is to learn visual representations that are specific to outdoor environments.

The second stage is downstream task training. In this stage, the MAE decoder is removed, the encoder is used as a feature extractor, and a task-specific prediction head is trained using labeled data.

It is important to note that self-supervised pretraining does not require manual annotations. This substantially reduces the need for labeled data, which is

particularly beneficial in robotics, where data annotation is often expensive and time-consuming.

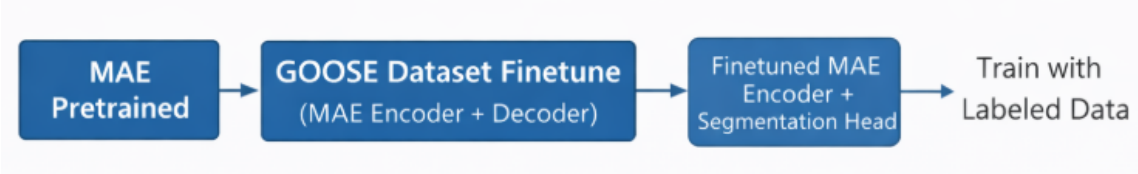

Figure 4.2: Two-stage training pipeline: domain-adaptive MAE pretraining followed by supervised fine-tuning.

4.3 Patch-Level Traversability Classification Model

To establish a baseline for traversability prediction, we first developed a patch-level classification model using the features learned by the MAE encoder. In this setup, each token produced by the encoder corresponds to a 16 × 16 image patch, and the task is to classify each patch as either traversable or non-traversable.

For an input image of size 224 × 224, the MAE encoder produces a 14 × 14 feature map, resulting in 196 tokens. Each token contains context-aware information due to the global self-attention mechanism of the transformer encoder. These token features are then fed into a lightweight classification head to generate patch-level predictions. Figure 4.3 illustrates the patch-level classification pipeline.

The classification head is a simple multi-layer perceptron (MLP) consisting of a linear layer, a GELU [27] activation function, and a second linear layer that outputs a binary classification score for each token. The design is intentionally simple because the MAE encoder already captures rich spatial and semantic relationships. The primary role of the classification head is to map these learned representations to traversability labels rather than learn complex spatial interactions, which are already modeled by the MAE encoder.

Our training procedure follows a two-stage fine-tuning strategy. In the first stage, the MAE encoder is kept frozen, and only the classification head is trained using labeled data. A relatively high learning rate is used to enable the newly initialized head to converge quickly. In the second stage, the last four layers of the MAE encoder are unfrozen and trained jointly with the classification head, while the earlier layers

remain frozen. This allows the model to adapt to the traversability task without significantly altering the pretrained representations.

Patch-level labels are generated from pixel-level annotations. A patch is labeled as traversable if more than half of the pixels within its 16 × 16 region belong to traversable classes. This provides a straightforward way to align the supervision with the patch-token-based features produced by the MAE encoder.

The patch-level model is computationally efficient and requires only a small amount of labeled data for training. However, because each patch covers a 16 × 16 pixel region, the spatial resolution of the predictions is limited. Small structures that occupy only a portion of a patch may not be captured accurately, which motivates the pixel-level segmentation model introduced in the following section.

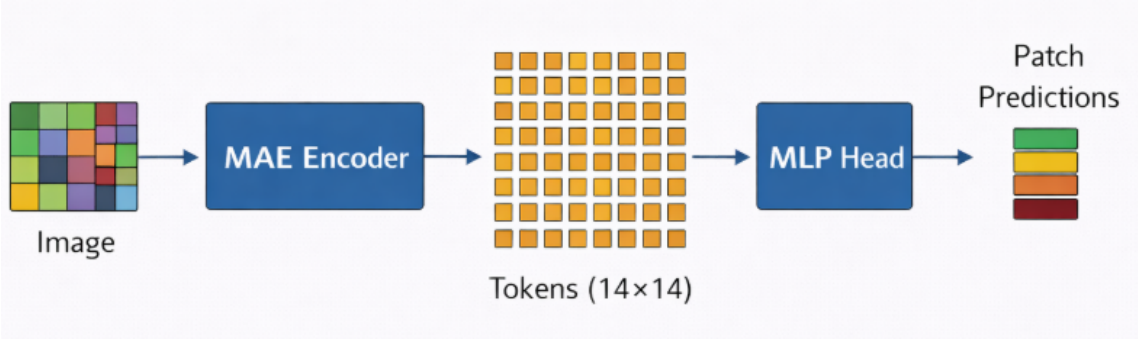

Figure 4.3: Patch-level traversability classification pipeline:
Image → MAE Encoder → Tokens (14×14) → MLP Head → Patch Predictions
Each MAE token corresponds to a 16×16 image region and is classified independently using a lightweight MLP head.

4.4 Pixel-Level Segmentation Model

Although our patch-level model is simple and efficient, its limited spatial resolution restricts its applicability in real-world scenarios. Therefore, we developed a pixel-level segmentation model to produce detailed traversability predictions at the original image resolution.

The MAE encoder outputs a feature map of size 14 × 14, representing the image at the patch level. To recover the original resolution, we employed a decoder network that gradually upsamples these features to produce a 224 × 224 prediction map. The decoder consists of several convolutional layers and upsampling stages, with each

stage increasing the spatial resolution and refining the feature representations. First, the 14 × 14 feature map is processed using convolutional layers to enhance local feature interactions. The features are then progressively upsampled through intermediate resolutions (14 × 14, 28 × 28, 56 × 56, 112 × 112, and 224 × 224) until the original image size is reached. The final output is a dense prediction map in which each pixel is classified as either traversable or non-traversable, matching the resolution of the input image.

It should be noted that the decoder used in this work differs from the original MAE decoder. The MAE decoder is designed for image reconstruction, where the objective is to predict missing pixel values. In contrast, our segmentation decoder is designed to predict a semantic class label for each pixel. Because of this difference, the original MAE decoder is not suitable for segmentation and is therefore replaced with a task-specific decoder.

Segmentation requires spatial continuity and the ability to capture fine boundaries between different regions. However, the patch-based design of the MAE encoder limits spatial resolution, as each token represents a 16 × 16 image region. This can lead to the loss of fine details, particularly when traversable and non-traversable areas are closely intermixed. Therefore, the decoder plays an important role in recovering spatial information and refining the final predictions.

Figure 4.4 illustrates the overall model pipeline. The input image is first processed by the MAE encoder to generate feature representations, which are then passed to the decoder to produce pixel-level traversability predictions. This design allows the model to benefit from strong representation learning while adapting effectively to dense prediction tasks.

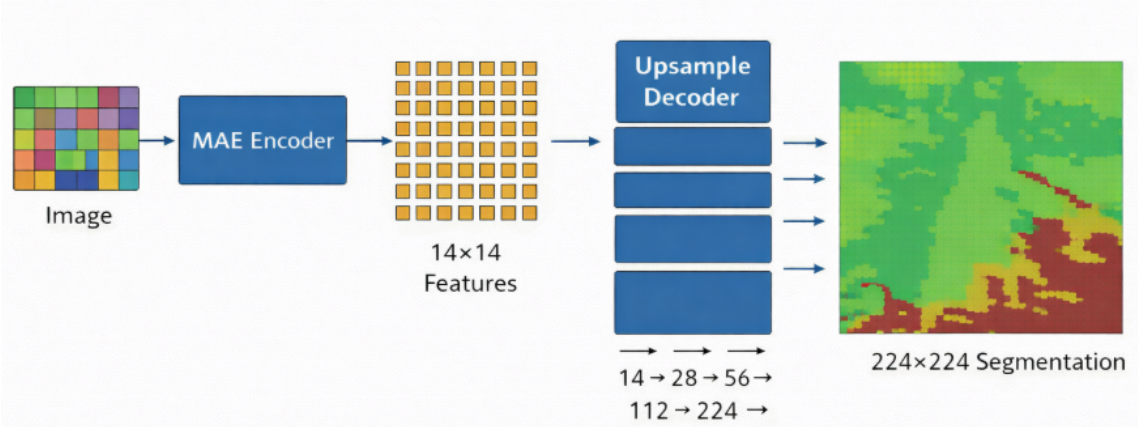

Figure 4.4: Pixel-level segmentation model:
Image → MAE Encoder → 14×14 Features → Upsample Decoder → 224×224 Segmentation ,The MAE encoder produces low-resolution feature maps, which are progressively upsampled by a convolutional decoder to generate dense traversability predictions.

4.5 Multi-Scale Supervision

To improve the spatial accuracy of pixel-level segmentation, we introduced a multi-scale supervision strategy, inspired by recent multi-branch and multi-scale learning approaches such as the Triple-Branch Multi-Scale Network [28]. The goal is to help the model preserve spatial information at different resolutions during decoding, particularly for fine details and small regions.

In the baseline pixel-level model, supervision is applied only at the final output resolution of 224 × 224. However, during upsampling, the decoder generates intermediate feature maps at lower resolutions, such as 28 × 28, 56 × 56, and 112 × 112. These intermediate feature maps contain useful spatial information that can also be supervised to guide the learning process.

In our multi-scale framework, auxiliary losses are applied to both the final output and the intermediate predictions. Lightweight prediction layers are used to generate predictions at each intermediate resolution, which are then compared with downsampled ground-truth labels to compute the auxiliary losses. The total training loss is defined as a weighted combination of the losses at different scales:

$$L_{total} = \lambda_1 L_{224} + \lambda_2 L_{112} + \lambda_3 L_{56}$$

where L_{224} represents the loss at the final resolution, and L_{112} and L_{56} represent the auxiliary losses at the intermediate resolutions. The loss weights are selected such that the final output receives the highest importance, while the intermediate outputs provide additional guidance during training.

The intuition behind this approach is that supervision at lower resolutions helps the model retain coarse structural information, while supervision at higher resolutions refines fine details such as the boundaries between traversable and non-traversable regions. By enforcing consistency across multiple scales, the model learns more robust spatial representations.

This multi-scale supervision strategy is lightweight and can be easily integrated into the existing decoder architecture described in Section 4.4. It introduces only a small computational overhead. The overall model pipeline is illustrated in Figure 4.5.

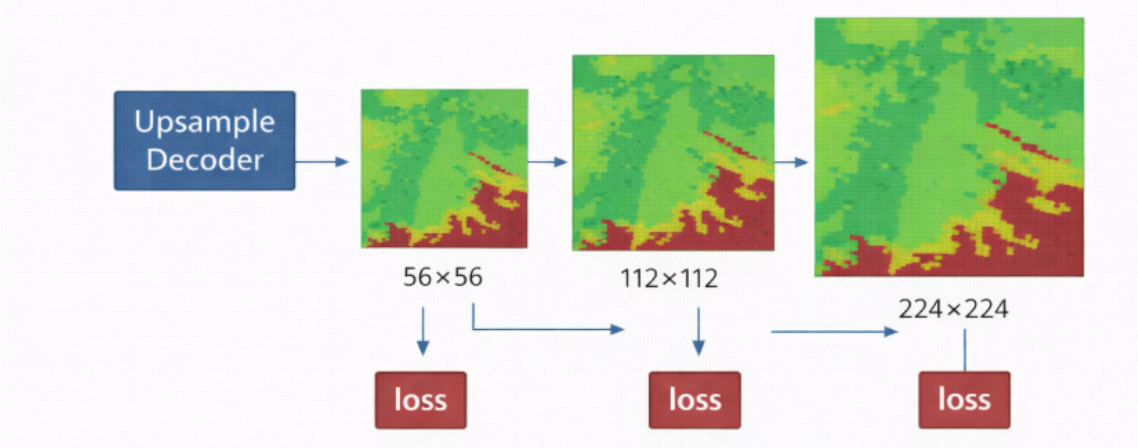

Figure 4.5: Multi-scale supervision strategy. Auxiliary losses are applied at intermediate resolutions to guide the learning of spatial features during upsampling.

4.6 Feature Pyramid Model

To improve our model's ability to capture spatial information, we introduced a feature pyramid architecture. The main idea is to utilize feature maps from multiple layers of the MAE encoder rather than relying solely on the final layer. The overall model pipeline is shown in Figure 4.6.

In transformer models, different layers capture different types of information. Early layers tend to preserve low-level details such as textures and edges, while deeper

layers focus on high-level semantic information. Relying only on the final layer can cause the model to lose fine spatial details that are important for pixel-level traversability prediction.

Feature maps are extracted from several stages of the MAE encoder, including early, intermediate, and final layers. These features are then combined within a feature pyramid decoder, which fuses information across different scales.

The feature pyramid decoder progressively combines features from different levels. High-level features provide strong semantic understanding, such as identifying terrain types, while low-level features provide detailed spatial information, including edges and textures. By merging these features, the model can better capture both global context and fine local details.

We also employed specialized loss functions to improve segmentation quality. In addition to the standard pixel-wise classification loss, Dice loss [29] is incorporated to address class imbalance and improve region-level consistency. An edge-aware loss [30] is also applied to emphasize boundary regions, helping the model produce sharper transitions between traversable and non-traversable areas.

By combining multi-level features with these complementary loss functions, the model can better detect both large continuous traversable regions and small obstacles or boundaries. This is particularly important in outdoor environments, where terrain characteristics can vary substantially across different spatial scales.

Overall, the feature pyramid model extends the baseline segmentation approach by explicitly leveraging multi-scale feature representations, enabling more accurate traversability predictions.

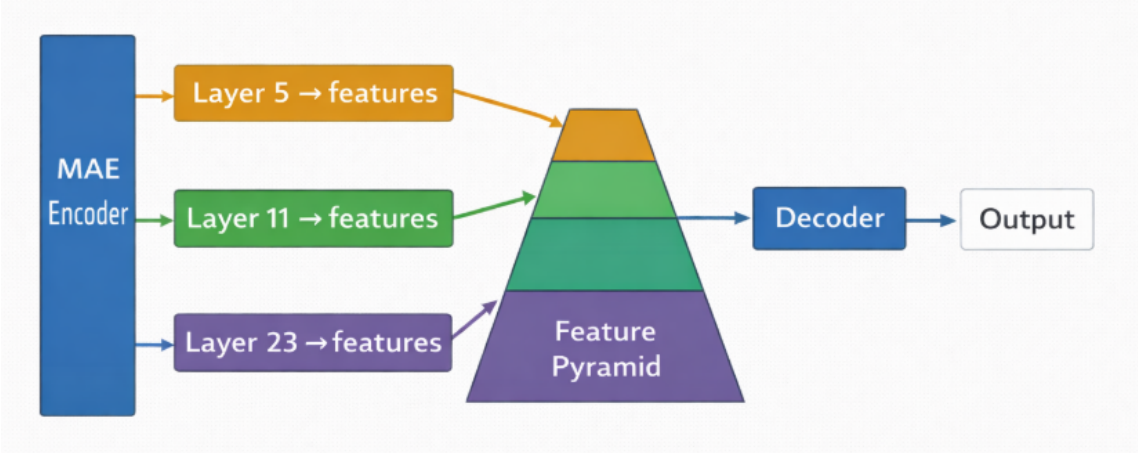

Figure 4.6: Feature pyramid architecture. Features from multiple layers of the MAE encoder are fused to combine low-level spatial information with high-level semantic representations.

4.7 LiDAR Fusion

To improve the model's understanding of spatial information, LiDAR depth information is incorporated into the feature pyramid framework. RGB images provide rich semantic and texture information, but they do not provide explicit 3D cues such as terrain height and slope. These properties are important for determining traversability in outdoor environments. Incorporating LiDAR data provides complementary geometric information that can help the model make more accurate predictions.

In this model, LiDAR point clouds are first projected into the camera coordinate system using calibration parameters, creating a depth map aligned with the RGB image. The depth values are normalized and then fused with visual features at intermediate feature-map resolutions (28 × 28 and 56 × 56) within the decoder. The fusion is performed at the feature level using element-wise addition:

$$F_{fused} = F_{rgb} + \alpha F_{depth}$$

where F_{rgb} represents features from the MAE encoder, F_{depth} represents features extracted from the depth map, and α is a learnable parameter that controls the contribution of the depth features. This fusion strategy enables the network to combine semantic information from RGB images with geometric cues from LiDAR during training.

An important design choice when incorporating LiDAR data is the resolution at which the depth features are fused with the RGB features. The MAE encoder produces a low-resolution feature map of size 14 × 14, where each token corresponds to a 16 × 16 pixel region of the original image. At this scale, fine details such as small obstacles may already be lost due to patch tokenization. To address this limitation, depth fusion is applied at higher-resolution stages within the decoder, specifically at the 28 × 28 and 56 × 56 feature maps. At a resolution of 56 × 56, each feature corresponds to approximately 4 × 4 pixels, preserving substantially more spatial detail and providing a better representation of geometric structures such as obstacle boundaries. In contrast, fusion at very high resolutions, such as 112 × 112, is less effective because LiDAR depth maps become sparse and noisy due to missing measurements and interpolation artifacts.

Figure 4.7 illustrates the feature fusion strategy used in the decoder, where depth features are combined with RGB features at the 28 × 28 and 56 × 56 stages to improve the model's understanding of scene geometry for traversability estimation.

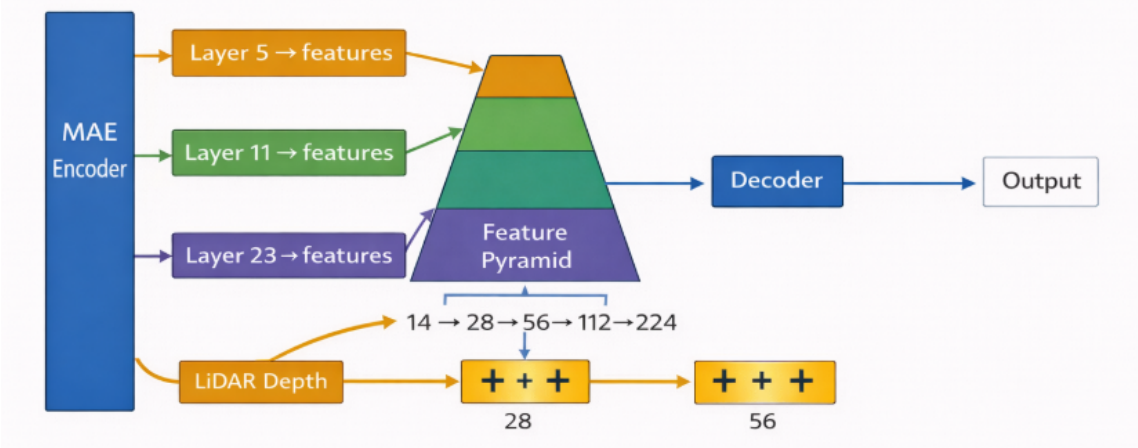

Figure 4.7: LiDAR Fusion Strategy at feature maps 28x28 and 56x56.

There are several approaches for combining RGB and depth information. Early fusion methods incorporate the depth channel directly into the RGB input, allowing the model to learn from both modalities from the beginning. However, this approach may not be well suited to our setting because the MAE encoder is pretrained on RGB images and may not effectively utilize the additional depth channel.

Another approach is the dual-encoder architecture, in which RGB and depth data are processed separately and fused at a later stage. While this can improve

modality-specific feature learning, it also increases model complexity. Token-level fusion is another alternative, where depth information is incorporated directly into the transformer tokens. However, this approach is computationally expensive.

The mid-level fusion approach used in this work provides a practical balance between performance and complexity. It preserves the pretrained MAE representations while allowing depth information to refine the spatial reconstruction process in the decoder. This approach improves performance without substantially increasing model complexity.

We also observed that LiDAR depth maps are inherently sparse, with missing values in regions where the sensor does not receive valid returns. Small or thin objects may not be captured reliably. As a result, LiDAR primarily contributes information about large-scale terrain geometry, while fine details, such as thin obstacles, must still be inferred from the RGB images.

5 Experiments & Results

5.1 Experimental setup

All experiments were conducted on a Google VM equipped with an NVIDIA Tesla T4 GPU, 2 CPU cores, and 12 GB of system memory. The models were implemented using the PyTorch framework, along with additional libraries for data processing and visualization.

The training procedure consists of a two-stage pipeline: self-supervised pretraining followed by supervised fine-tuning.

Stage 1: Domain-Adaptive MAE Pretraining

The Masked Autoencoder (MAE) [1], initialized with pretrained ImageNet weights, is further trained on the GOOSE dataset (including both GOOSE and GOOSE-Ex) using a self-supervised reconstruction objective. During this stage, both the encoder and decoder are trained jointly without requiring labeled data.

Stage 2: Traversability Model Training

After pretraining, the MAE decoder is removed and replaced with task-specific prediction heads for traversability estimation. The fine-tuning procedure is performed in two steps. First, the MAE encoder is kept frozen, and only the classification or segmentation head is trained. A relatively higher learning rate is used during this stage to enable rapid adaptation to the downstream task.

Second, partial fine-tuning is performed, in which the final four layers of the MAE encoder are unfrozen and trained jointly with the prediction head. A smaller learning rate is applied to the encoder to ensure that the pretrained representations are preserved. This strategy helps prevent catastrophic forgetting while allowing the model to adapt to the characteristics of the target domain.

Training Hyperparameters

The models were trained using standard optimization settings. The main hyperparameters are summarized in Table 5.1.

Table 5.1: Training hyperparameters.

Optimizer	LR	Batch Size	Epochs
AdamW [31]	Encoder: 3e-6 Head: 1e-4	16	70 (with early stopping)

The MAE encoder was found to be highly sensitive to the learning rate during fine-tuning. Even small increases in the learning rate can negatively affect the pretrained representations. Therefore, careful learning rate selection is required, and very low learning rates are used to ensure stable training. We also observed that full fine-tuning of all encoder layers is unstable, particularly given the size of our dataset. For this reason, most experiments fine-tune only the final four transformer layers.

Evaluation Metrics

To evaluate model performance, both patch-level and pixel-level metrics are used. For patch-level classification, precision, recall, and F1-score are reported. For pixel-level segmentation, Intersection over Union (IoU) is additionally included alongside the metrics used for patch-level classification. Qualitative evaluation is also performed by overlaying the predicted traversability maps on the input images to visually assess the segmentation results.

Experimental Reliability Considerations

A note regarding the interpretation of the reported results should be made. All performance metrics presented in this chapter are derived from single training runs and are not averaged across multiple random seeds. Furthermore, confidence intervals and variance measures are not reported. Therefore, small performance differences between models should be interpreted with caution. For example, the IoU difference between the feature pyramid model and U-Net when trained on 10% of the dataset (0.7676 versus 0.7634) is relatively small and may be influenced by run-to-run variability. In contrast, larger and more consistent performance differences, such as the superior boundary segmentation performance of DeepLabv3 observed across multiple training data proportions, are less likely to be explained solely by random variation. A more comprehensive statistical analysis based on repeated experiments and variance reporting would provide stronger evidence for the observed trends and is identified as an important direction for future work.

5.2 Patch-level results

The first set of experiments focuses on evaluating the patch-level traversability classification model. In this experiment, each 16 × 16 image patch is classified as either traversable or non-traversable based on the features extracted by the MAE encoder. The model performs well on the validation set, achieving an F1-score greater than 0.84. Table 5.2 summarizes the results, including the precision, recall, and F1-score.

Table 5.2: Patch-level classification performance.

Model	Precision	Recall	F1-score
Patch-level(MAE)	0.7977	0.8980	0.8449

Figure 5.1 shows example predictions produced by the patch-level model. The model correctly identifies traversable areas such as roads, grass, and open ground, often forming large continuous regions. These results indicate that the model effectively captures the overall scene context and distinguishes between different terrain types.

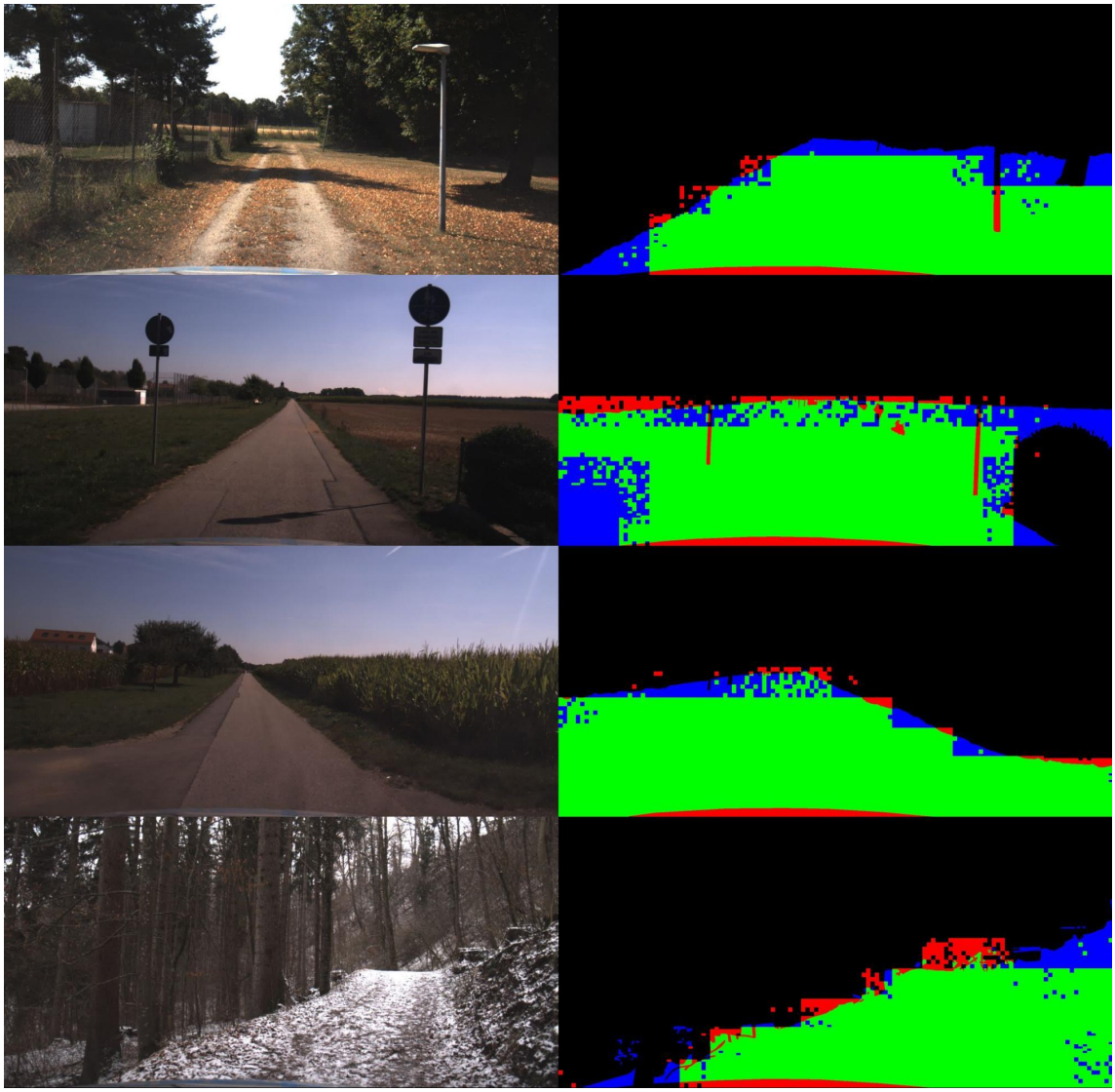

Figure 5.1: Patch-Level Traversability Prediction with Error Visualization
Each row shows the input image (left) and the corresponding prediction overlay (right), where true positives are shown in green, false positives in red, and false negatives in blue.

Although the patch-level model shows strong quantitative performance, it has several limitations due to its relatively low spatial resolution. Since each prediction corresponds to a 16 × 16 pixel region, the model struggles to capture fine details such as thin obstacles, small objects, and narrow boundaries. As a result, the predicted traversability maps appear blocky and lack precise boundary information. This limitation highlights the need for pixel-level segmentation models, which can provide higher-resolution predictions and capture finer spatial details.

5.3 Pixel-Level Segmentation Results

In this section, we evaluate the performance of the pixel-level segmentation model. Unlike the patch-level model, which predicts one label per 16 × 16 patch, the pixel-level model generates detailed traversability maps at the original image resolution. This enables more accurate localization of traversable and non-traversable regions, particularly along object boundaries.

The segmentation model employs a decoder that upsamples the feature representations produced by the MAE encoder. The encoder first generates a feature map of size 14 × 14, which is then progressively upsampled to 224 × 224 using convolutional and interpolation layers. This design enables the model to generate full-resolution segmentation masks while leveraging the semantic features learned during MAE pretraining.

The performance of the pixel-level model is evaluated using standard segmentation metrics, including Intersection over Union (IoU), F1-score, precision, and recall. The results are summarized in Table 5.3. To further analyze the model's behavior, qualitative results are presented in Figure 5.2, which shows example predictions from the test set.

Table 5.3: Pixel-level segmentation base upsample model performance.

Model	IoU	F1-score	Precision	Recall
MAE(pixel) upsample	0.7337	0.8464	0.8318	0.8615

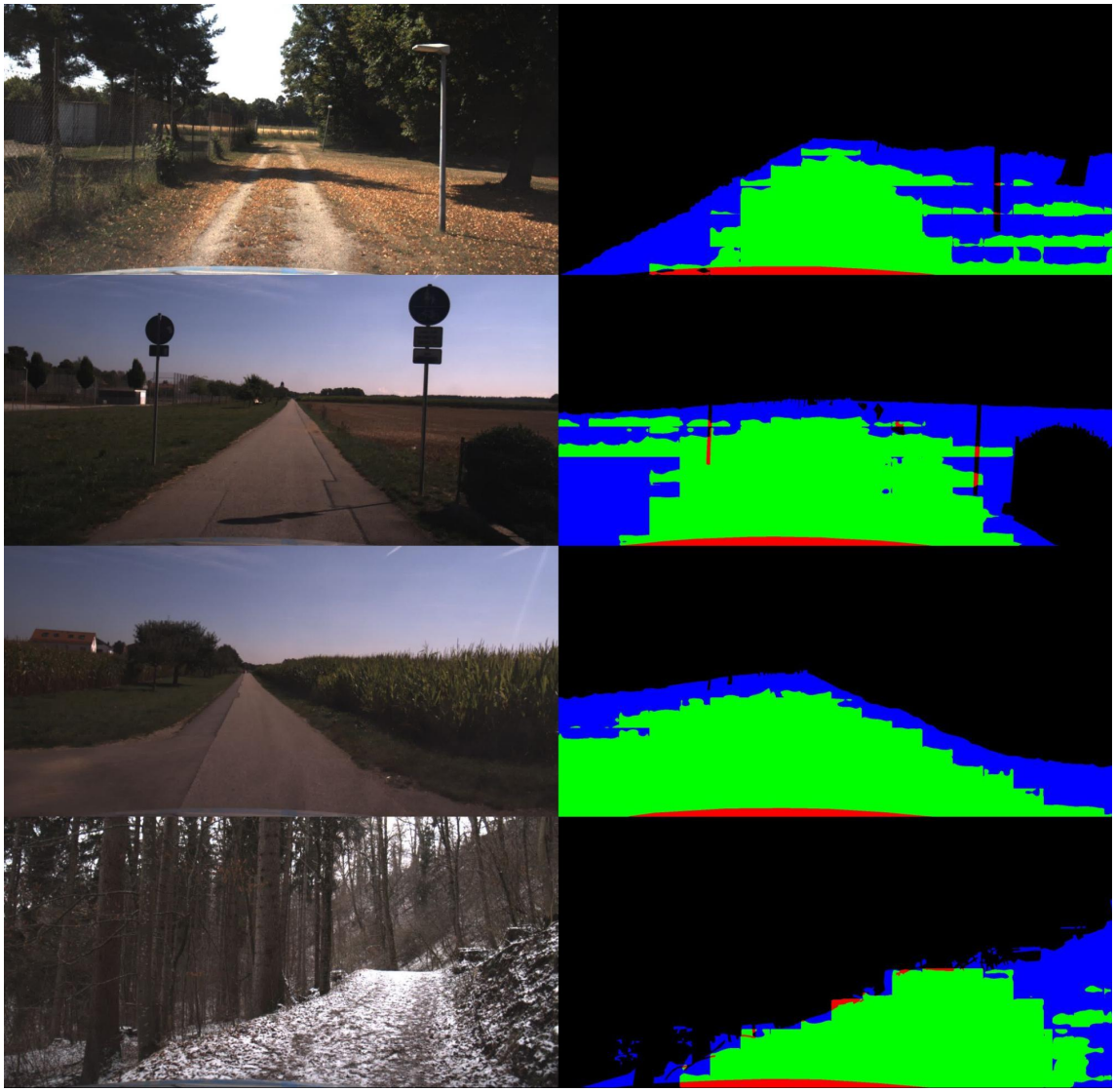

Figure 5.2: Pixel-Level(Base upsample) Traversability Prediction with Error Visualization.
Each row shows the input image (left) and the corresponding prediction overlay (right), where true positives are shown in green, false positives in red, and false negatives in blue.

As shown in Figure 5.2, the model accurately identifies large traversable regions such as roads and open terrain. However, errors are more noticeable in regions with complex boundaries and small obstacles. The pixel-level model shows clear improvements over the patch-level approach, particularly in the segmentation of large traversable areas, where the patch-level model tends to produce noisier predictions. It accurately segments large regions such as roads, grass fields, and open terrain, suggesting that the MAE encoder captures strong semantic features.

However, several limitations remain. For example, thin structures, such as small obstacles, are often missed, boundary regions may appear imprecise, and areas containing a mixture of traversable and non-traversable terrain are more difficult to distinguish. These regions remain challenging to segment accurately. This is primarily due to the limited spatial resolution of the MAE encoder. Since the input image is divided into 16 × 16 patches, some fine-grained details are lost early in the encoding process. As a result, the decoder cannot fully recover precise boundaries during feature upsampling.

These results highlight a common limitation of patch-based transformer encoders. While they are effective at capturing the global semantic context of a scene, they often struggle to preserve the fine spatial details required for precise segmentation.

5.4 Multi-Scale Supervision and Feature Pyramid Results

To reduce the spatial limitations observed in the baseline pixel-level model, we evaluate two improvement strategies: multi-scale supervision and feature pyramid modeling. These methods aim to improve boundary accuracy and capture finer details by leveraging features at different resolutions. The performance of these enhanced models is presented in Table 5.4, where the following configurations are compared: the baseline pixel-level model, the pixel-level model with multi-scale supervision, and the pixel-level model with feature pyramid modeling.

Table 5.4: Comparison of Pixel-Level Segmentation Models with Multi-Scale Supervision and Feature Pyramid.

Model(MAE pixel-level)	IoU	F1-score	Precision	Recall
Baseline	0.7337	0.8464	0.8318	0.8615
Multi-scale supervision	0.7915	0.8836	0.9070	0.8614
Feature pyramid	0.7676	0.8685	0.8266	0.9149

The multi-scale supervision model introduces additional loss terms at intermediate resolutions (56 × 56 and 112 × 112), which encourages the model to learn useful representations at multiple scales. The results show a slight improvement in overall performance; however, it does not lead to a significant improvement in boundary accuracy.

The feature pyramid model, in contrast, combines features from different layers of the encoder. This approach enables the model to leverage both high-level semantic information and low-level spatial details. Compared to the baseline model, the feature pyramid approach better detects small regions, produces more detailed segmentation masks, and generates sharper boundaries, although the overall F1-score remains approximately unchanged.

As shown in Figure 5.3, the feature pyramid model produces clearer boundaries and better preserves small structures compared to both the baseline model and the multi-scale supervision model.

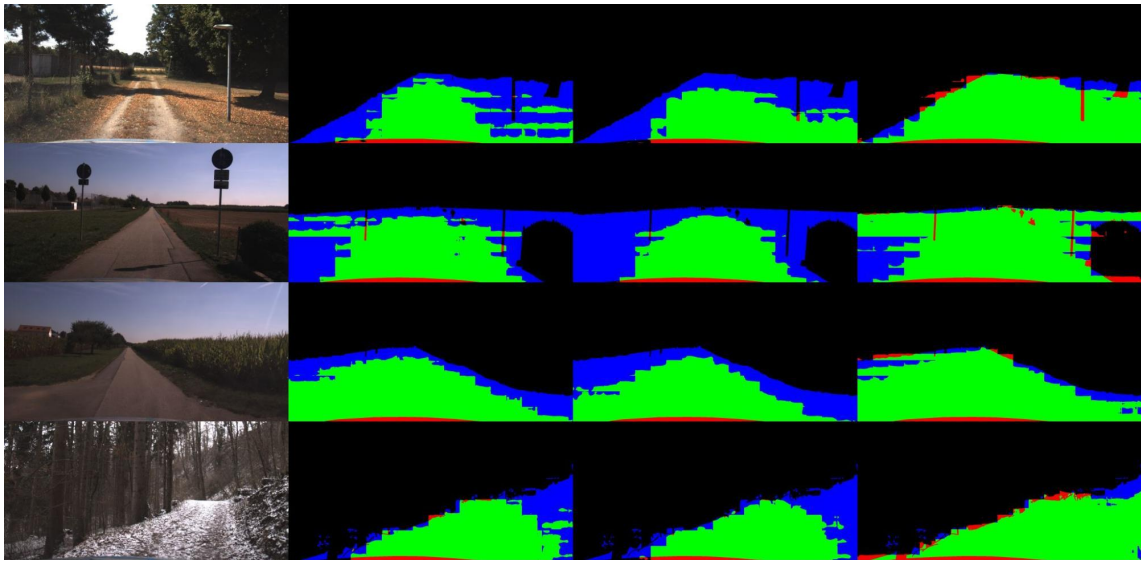

Figure 5.3: Qualitative comparison of pixel-level segmentation models. Each row contains four images: the input RGB image (left), followed by predictions from the baseline model, the multi-scale supervision model, and the feature pyramid model (right). True positives are shown in green, false positives in red, and false negatives in blue.

Despite these improvements, several challenges remain in the feature pyramid model. For example, thin objects are still difficult to detect, particularly those smaller than the patch size. Mixed regions are sometimes incorrectly classified, and some boundaries remain slightly blurred.

These results indicate that, although multi-scale learning improves training, it does not fully resolve the limitations caused by the encoder’s spatial resolution. In general, multi-scale supervision and feature pyramid architectures improve performance; however, they cannot recover details that are lost during the initial patch embedding

process in the MAE encoder. This highlights a trade-off in transformer-based models: while they are effective at capturing global scene context, they may struggle with precise spatial details.

5.5 LiDAR Fusion Results

In this section, we evaluate the effect of incorporating LiDAR depth information on the performance of the traversability model. The motivation is that LiDAR provides complementary geometric information to RGB images, which helps the model better understand terrain structure and detect obstacles.

In our experiments, depth information is incorporated at the decoder stage of the network, after RGB features have been processed by the MAE encoder. The depth map is aligned with the camera view and passed through a convolutional layer for feature extraction. These depth features are then fused with visual features using feature-level fusion. The performance of the model, with and without LiDAR fusion, is presented in Table 5.5. To further analyze the impact of LiDAR fusion, qualitative results are shown in Figure 5.4.

Table 5.5: Comparison of Pixel-Level Segmentation Feature Pyramid Models with and without LiDAR fusion.

Model(Feature pyramid)	IoU	F1-score	Precision	Recall
RGB only	0.7676	0.8685	0.8266	0.9149
RGB + LiDAR	0.7404	0.8509	0.7717	0.9481

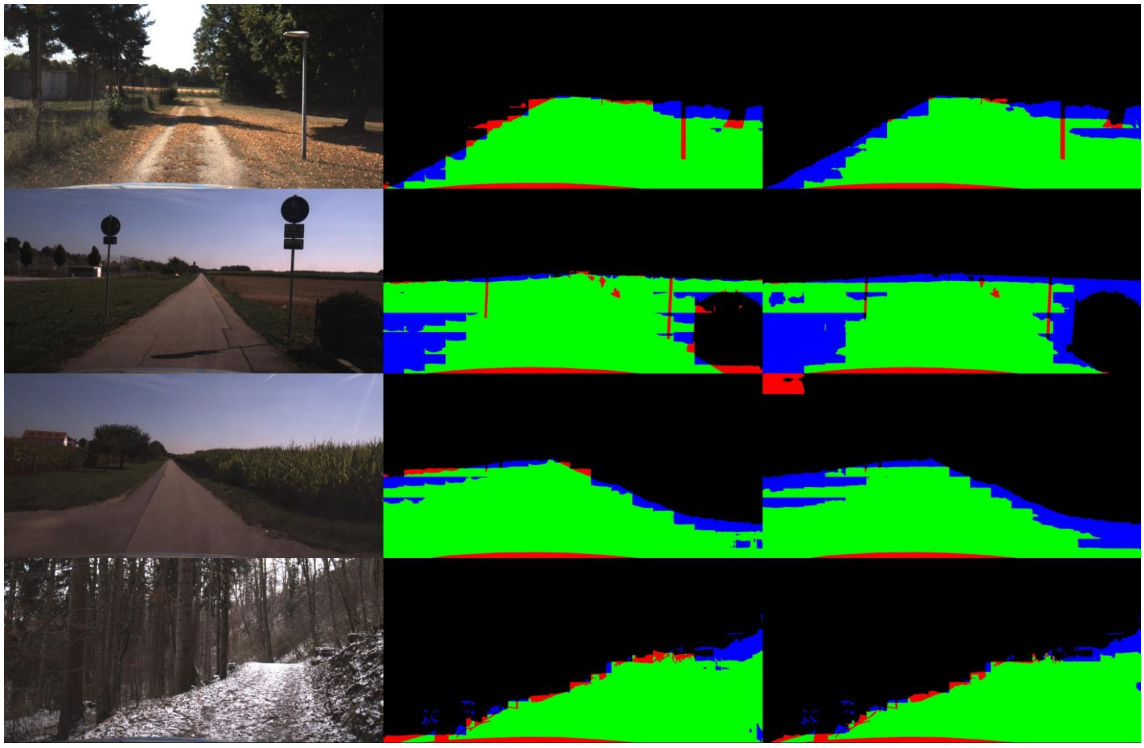

Figure 5.4: Qualitative comparison of RGB-only and RGB+LiDAR fusion segmentation results. Each row contains three images: the input RGB image (left), the RGB-only feature pyramid model prediction (middle), and the RGB+LiDAR fusion model prediction (right). True positives are shown in green, false positives in red, and false negatives in blue.

The results show that the LiDAR fusion model performs similarly to the feature pyramid model, with only minor differences in overall performance. Although LiDAR provides useful geometric information for larger structures, its impact is limited because the feature pyramid model already handles these regions effectively. In addition, LiDAR data is sparse for small or thin objects, and therefore provides limited improvement for fine-grained details. As a result, the model still struggles with thin structures, small obstacles, and boundary regions.

One important limitation arises from the fusion location in our architecture. The MAE encoder reduces the input image to a low-resolution feature map of size 14 × 14, where each token represents a 16 × 16 pixel region. Consequently, fine details are already partially lost before LiDAR information is introduced.

Since LiDAR is integrated at the decoder stage, it primarily contributes to understanding the overall scene geometry rather than improving fine boundary details. Furthermore, LiDAR depth maps can be noisy, particularly in regions

containing small or thin objects, which further limits their contribution to fine-grained segmentation.

From the experimental results, it can be observed that while LiDAR provides useful geometric cues, its impact is limited when introduced after spatial resolution has already been reduced by the encoder. This suggests that incorporating LiDAR earlier in the network, or fusing it at multiple stages, may be more effective for preserving fine spatial details.

5.6 Baseline Comparison (U-Net, DeepLabv3)

To evaluate the effectiveness of the proposed approach, we compare the best-performing model, namely the feature pyramid model, with two widely used semantic segmentation architectures: U-Net [7] and DeepLabv3 [8]. U-Net is a convolutional encoder–decoder network that utilizes skip connections to preserve spatial information, while DeepLabv3 employs atrous spatial pyramid pooling (ASPP) to capture multi-scale contextual information.

To provide a more comprehensive evaluation, the comparison is conducted using different proportions of the labeled training dataset (10%, 30%, and 50%). This allows us to assess not only the segmentation performance of each model but also their ability to learn from varying amounts of training data. The quantitative segmentation results are presented in Table 5.7. In addition, the computational efficiency of each model, including inference speed and model size, is reported in Table 5.6. Qualitative prediction examples are shown in Figure 5.5.

Table 5.6: Computational Efficiency Comparison of the Feature Pyramid Model, U-Net, and DeepLabv3.

Model	FPS(T4 GPU)	Params (M)
Feature Pyramid	22.22	304
U-Net	58.82	31
DeepLabv3	41.67	39.6

Table 5.7: Segmentation Performance Comparison of the Feature Pyramid Model, U-Net, and DeepLabv3 Using Different Proportions of the Training Dataset.

Training Data	Model	IoU	F1-score	Precision	Recall
10%	Feature Pyramid	0.7676	0.8685	0.8266	0.9149
10%	U-Net	0.7634	0.8658	0.8276	0.9077
10%	DeepLabv3	0.8366	0.9110	0.8989	0.9235
30%	Feature Pyramid	0.7785	0.8754	0.8491	0.9035
30%	U-Net	0.7312	0.8447	0.8058	0.8876
30%	DeepLabv3	0.8084	0.8941	0.8445	0.9498
50%	Feature Pyramid	0.7981	0.8877	0.8556	0.9224
50%	U-Net	0.7799	0.8764	0.8815	0.8713
50%	DeepLabv3	0.8124	0.8965	0.8476	0.9513

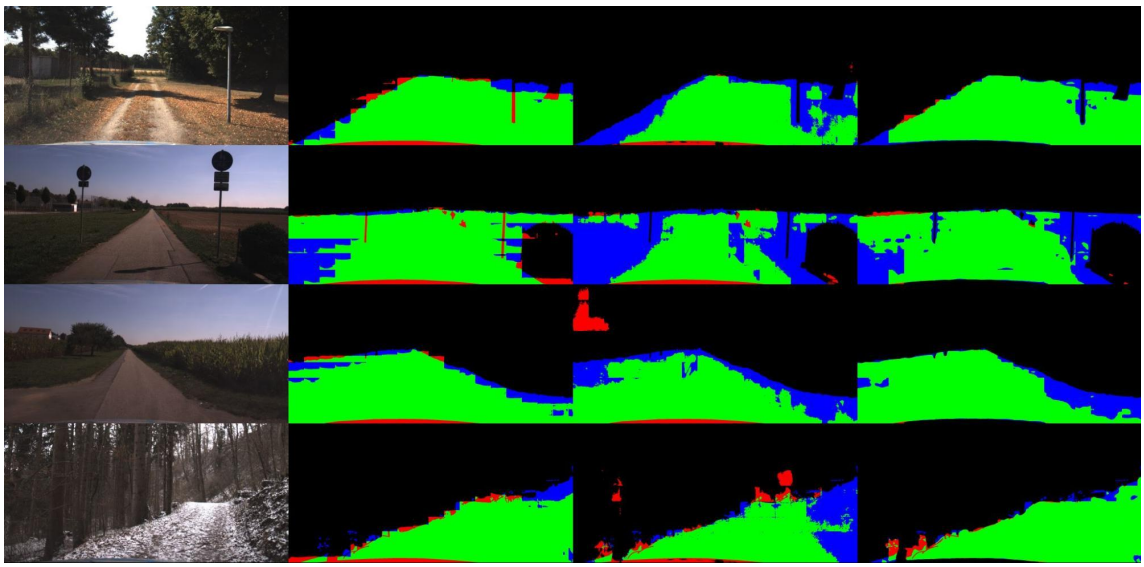

Figure 5.5: Qualitative comparison of segmentation results across different models. Each row contains four images: the input RGB image (left), followed by predictions from the proposed feature pyramid model, U-Net, and DeepLabv3 (right). True positives are shown in green, false positives in red, and false negatives in blue.

The results in Table 5.7 show clear differences between the proposed transformer-based approach and conventional CNN-based segmentation models.

Across different training data sizes, the relative performance characteristics remain similar.

U-Net preserves high-resolution spatial information through its skip-connection design and achieves competitive segmentation performance. Nevertheless, its overall accuracy remains lower than that of both the feature pyramid model and DeepLabv3. Qualitative results in Figure 5.5 show that U-Net produces more false positives and false negatives, particularly in complex outdoor scenes containing mixed traversable and non-traversable regions.

DeepLabv3 achieves the highest quantitative performance among the evaluated models. Its ASPP module effectively captures multi-scale contextual information, allowing it to segment both large traversable regions and fine boundary structures accurately. In particular, DeepLabv3 performs well around object boundaries and small structures, where preserving local spatial information is important.

An interesting observation from Table 5.7 is the different performance trends exhibited by the evaluated models as the amount of labeled training data increases. The proposed feature pyramid model shows a consistent improvement in IoU, increasing from 0.7676 at the 10% setting to 0.7981 at the 50% setting. In contrast, DeepLabv3 achieves its highest IoU when trained on 10% of the data (0.8366) and exhibits lower performance at both the 30% (0.8084) and 50% (0.8124) settings. Several factors may contribute to this behaviour. First, the 10%, 30%, and 50% training sets are independently sampled subsets rather than nested subsets, meaning that each configuration may contain slightly different data distributions. Consequently, the 10% subset may have been particularly well suited to the characteristics of DeepLabv3. Second, all experiments were conducted using a single training run, and therefore the reported results may be influenced by random initialization and training variability. Finally, it is possible that DeepLabv3 reaches its optimal performance relatively early on this binary segmentation task and becomes more sensitive to annotation noise or ambiguous samples as additional data is introduced. Determining the exact cause would require repeated experiments with controlled random seeds and multiple dataset samplings, which are left for future work. Nevertheless, this observation should be considered when interpreting the comparative performance of the evaluated models.

The proposed feature pyramid model demonstrates strong performance across different training data sizes and consistently outperforms U-Net. The model is particularly effective at identifying large continuous traversable regions such as roads, grass fields, and open terrain. These results suggest that the MAE encoder provides robust semantic representations that help the model understand the overall scene structure. As shown in Figure 5.5, the feature pyramid model often produces more coherent predictions in large terrain regions, although some fine boundary details remain less precise than those produced by DeepLabv3.

Table 5.6 highlights the computational trade-offs between the evaluated models. The proposed feature pyramid model contains substantially more parameters than both U-Net and DeepLabv3 and achieves the lowest inference speed on the Tesla T4 GPU. In contrast, U-Net is the most computationally efficient model, while DeepLabv3 provides a balance between segmentation accuracy and computational cost.

It is also useful to place these results in the context of the original GOOSE benchmark [24]. A direct numerical comparison with the published benchmark results is not appropriate for several reasons. First, the original GOOSE benchmark evaluates a multi-class semantic segmentation task across the complete label taxonomy, whereas this work reformulates the problem as a binary traversable/non-traversable segmentation task. This changes both the class distribution and the overall difficulty of the problem. Second, the experiments presented in this thesis are intentionally conducted using reduced labeled training subsets (10%, 30%, and 50% of the available training data) in order to evaluate the effectiveness of self-supervised representation learning under limited-label conditions. Third, differences in dataset splits, traversability label definitions, preprocessing procedures, and evaluation protocols further limit direct comparability. Therefore, the benchmark results reported in the original GOOSE paper should be viewed as a reference point for fully supervised multi-class segmentation, while the results presented in this thesis demonstrate the effectiveness of the proposed approach in a label-efficient binary traversability estimation setting.

Overall, the comparison reveals a trade-off between convolution-based and transformer-based approaches. CNN-based models, particularly DeepLabv3, are

effective at preserving fine spatial details and object boundaries. In contrast, the MAE-based feature pyramid model benefits from strong pretrained representations and demonstrates a superior understanding of global scene structure. Although the proposed model achieves competitive segmentation performance, its accuracy is still constrained by the reduced spatial resolution introduced by the patch-based tokenization process of the MAE encoder.

5.7 Error Analysis and Limitations

In this section, we look at the failure cases of our traversability model, focusing on thin structures, boundary areas, and regions where traversable and non-traversable parts are mixed.

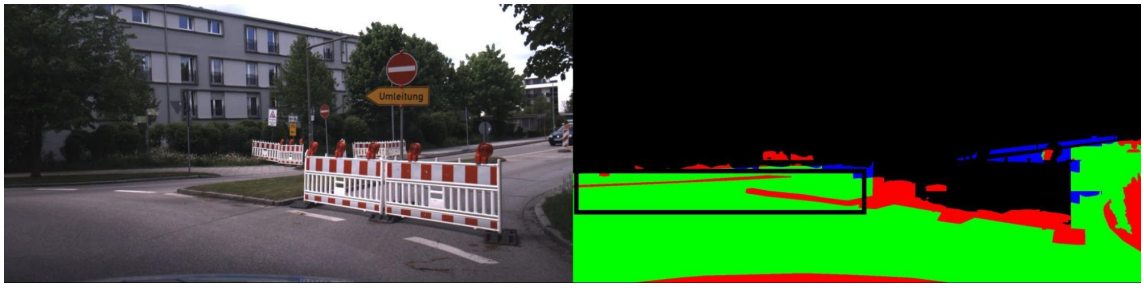

Figure 5.6: Examples of failure cases, highlighting incorrect predictions in thin and interlaced regions, which are marked with black rectangles. True positives are shown in green, false positives in red, and false negatives in blue.

As shown in Figure 5.6, our model works well on large, continuous traversable and non-traversable areas. However, its performance drops in some challenging cases, such as thin structures (for example narrow obstacles) and boundary areas with sharp changes. This result matches the quantitative results, that good overall metrics (like F1 score) do not always mean the model performs well on fine details.

Spatial Resolution Limitation of MAE

One of the main limitations is how the MAE encoder represents the image using patches. The input image (224×224) is divided into a grid of 14×14 tokens, where each token corresponds to a 16×16 pixel region. This creates an important limitation that any structure smaller than 16×16 pixels is merged into a single token, and fine details are lost during encoding(such as edges and small obstacles). Because of this, encoders cannot clearly represent small features like tree roots and narrow

obstacles. This explains why small objects and mixed regions are usually misclassified. Figure 5.7 clearly shows that smaller obstacles are embedded inside a patch.

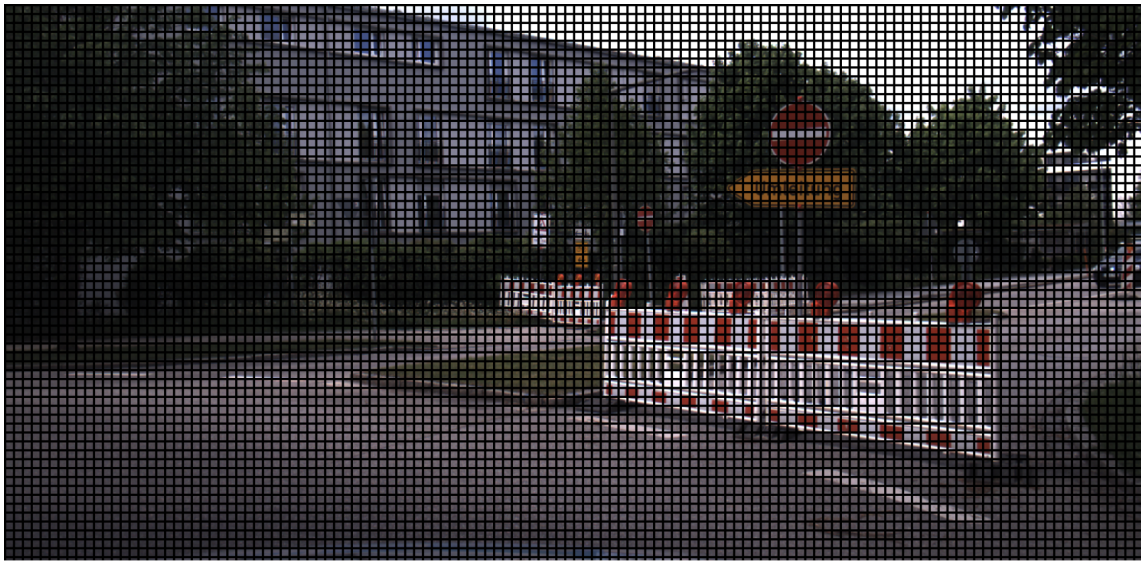

Figure 5.7: Illustration of patch tokenization in MAE, where each token corresponds to a 16×16 pixel region(marked by black grid).

Limitations of Loss Functions

We used different loss functions, such as Dice loss and edge-aware loss, and tried to improve boundary quality. But these methods mainly help the model learn better decision boundaries and improve classification near edges, however they cannot bring back spatial details that were already lost during encoder tokenization. This tells an important point that loss functions cannot recover information that is missing from the feature representation. Because of this, even with more advanced loss functions, our model still struggles with fine boundary details.

MAE Training Objective

The Masked Autoencoder (MAE) [1] is designed to reconstruct missing image patches and learn global semantic features. It is not specifically trained to preserve boundaries and capture fine-grained details. So the features it learns focus on overall semantic consistency and global context. But in our pixel segmentation task, this can lack sharp edges and precise spatial details.

Effect of Global Attention

The transformer model self-attention mechanism allows features from across the whole image to interact. This helps the model understand the overall scene, but it also has some drawbacks for our task. For example, features from distant areas get mixed together, local boundaries can be smoothed out and small spatial details are lost. So for the segmentation task mixed regions are averaged out and the model is less sensitive to small structures.

Class Imbalance

Another factor affecting performance is class imbalance in the GOOSE dataset. Traversable areas make up only about 35% of the total area, and within these, grass terrain dominates. So in the end, the model can be biased toward the dominant class(even if we use weight loss for different classes), rare classes are under-represented (such as small obstacles) and predictions become conservative. This makes it harder for our model to perform well in challenging areas.

Implications for Traversability Detection

The results show that our model works well for coarse traversability estimation, especially when regions are large and continuous and understanding the overall scene is enough. But its performance drops in situations that require fine spatial precision, such as avoiding small obstacles and detecting narrow paths.

Design Trade-off

A key trade-off in this architecture comes from the patch size (16×16). It is efficient and captures strong semantic features, but it limits spatial resolution. Reducing the patch size (for example, to 8×8, or even 4x4) could improve details, but it would also increase computational cost (more patches per input) and increase memory usage, it could also require more training data to converge.

The main limitation of our current approach is not the lack of geometric information, but the loss of spatial resolution during patch-based encoding. Once fine details are compressed into tokens, they cannot be recovered later, no matter what loss function or fusion strategy is used.

6 Discussion

In this chapter, we present and analyze the results of our experiments, highlighting key performance trends, limitations of the proposed approach, and their implications for robotic perception in unstructured outdoor environments. We examine the performance of self-supervised transformer-based models, particularly in comparison with traditional convolutional methods, and investigate how different architectural and data-related factors influence model performance.

6.1 Interpretation of Results

The experimental results show that the MAE-based framework performs well in segmenting large and continuous traversable regions. The model effectively captures global scene structure and maintains semantic consistency, which enables accurate identification of terrain categories such as roads, grassland, and open terrain.

This performance is likely attributed to the ability of transformer-based encoders to model long-range dependencies and capture global contextual information. In addition, self-supervised pretraining allows the model to learn general visual representations without requiring large amounts of labeled data, which is particularly beneficial in robotics applications where annotation is costly.

Despite these advantages, the model still struggles with fine-grained details such as thin structures, small obstacles, and complex boundaries. This indicates a trade-off between global contextual understanding and local spatial precision.

6.2 Strengths of Self-Supervised Transformer Representations

One of the main advantages of this approach is the use of self-supervised pretraining. The MAE encoder, initially pretrained on large-scale datasets such as ImageNet, provides strong visual representations that can be effectively transferred to downstream tasks such as segmentation.

Our results indicate that:

- The encoder is able to learn strong semantic features, even with limited labeled data.
- The model generalizes well across different outdoor environments.

- It reliably segments large-scale structures.

Notably, even when reconstruction quality degrades during domain adaptation, the learned representations still improve performance on the downstream task. This suggests that reconstruction quality is not always a reliable indicator of downstream performance in self-supervised learning. Instead, the encoder appears to prioritize features that are more relevant for scene understanding, such as texture and structural patterns.

6.3 Limitations of MAE for Fine-Grained Segmentation

One of the main limitations observed is that the MAE-based architecture struggles to capture fine-grained spatial details. As discussed in Chapter 5, this issue is primarily caused by the patch-based tokenization process. Each token covers a relatively large spatial region (16×16 pixels in our case), which leads to loss of small-scale features within each patch, blurred object boundaries, and reduced sensitivity to thin structures.

This limitation is further influenced by the MAE training objective, where the reconstruction loss focuses on recovering masked image patches rather than preserving fine geometric structures or sharp edges. In addition, the global self-attention mechanism in transformers tends to smooth local variations in early layers, emphasizing semantic consistency over precise spatial localization. While this improves global scene understanding, it negatively affects segmentation performance in regions requiring fine spatial detail.

6.4 Effect of Dataset Size and Domain Adaptation

Another factor affecting performance is the size and diversity of the GOOSE dataset. The original MAE model was pretrained on a large-scale dataset containing approximately 1.2 million images (ImageNet-1K), providing substantial visual diversity. In contrast, the domain-specific GOOSE dataset used in this work is considerably smaller, with approximately 12,000 images, and is primarily focused on outdoor terrain scenes. These differences introduce several challenges, including reduced variability in textures and visual patterns, an increased risk of overfitting to dataset-specific features, and limited generalization during fine-tuning.

As the model adapts to the target domain, it may shift from learning general representations toward more domain-specific features, a phenomenon often referred to as feature drift. While this adaptation can improve performance in the target domain, it may also reduce texture diversity, limit reconstruction variability, and negatively affect generalization ability.

In addition, repetitive patterns such as grass, sky, and ground may encourage the model to rely on shortcut representations, resulting in overly smooth predictions and reduced sensitivity to fine-grained structural details.

6.5 Analysis of LiDAR Fusion

We also investigated the use of LiDAR depth information to improve geometric understanding, although it did not lead to significant improvements in overall segmentation performance. Several factors may explain this result. First, LiDAR data is sparse and noisy in many regions, particularly for small or thin structures. Second, depth information is introduced after the encoder stage, where a substantial amount of spatial detail has already been compressed. Third, the model primarily relies on RGB features for fine-grained segmentation. As a result, LiDAR contributes mainly to coarse geometric understanding of the scene, but provides limited improvement in boundary precision.

Despite these limitations, the fusion strategy still offers certain benefits. It improves training stability and introduces complementary geometric information. It also provides flexibility for integrating different sensor modalities. The experimental results suggest that more effective fusion strategies, such as early fusion or multi-level fusion within the encoder, may be required to fully exploit depth information.

6.6 Architectural Trade-offs

Our results highlight a key trade-off in transformer-based vision models. They are effective at capturing global scene context, but they are less effective in modeling fine-grained spatial details. This is particularly relevant for traversability estimation. While understanding the overall scene is essential for identifying large terrain regions, precise navigation requires accurate detection of small obstacles.

In contrast, CNN-based models such as DeepLabV3 achieve better performance in fine-grained segmentation, as they are better able to preserve spatial details through skip connections. This observation suggests that hybrid architectures combining transformers and CNNs may provide a more balanced solution.

6.7 Practical Implications for Robotics

Our approach is well-suited for tasks where coarse traversability estimation is sufficient, such as path planning in open areas or general terrain classification. However, for safety-critical applications that require precise obstacle detection, the current limitations need to be addressed. In particular, the model's inability to detect small obstacles may lead to unsafe navigation decisions in certain scenarios.

Several strategies can be considered to mitigate this issue, such as high-resolution perception modules or sensor fusion with additional modalities.

6.8 Future Improvements

Based on the experimental results, several potential directions for improvement can be identified:

- Higher-resolution encoding: using smaller patch sizes or hybrid architectures (e.g., 8×8 or 4×4 patches) to improve spatial detail preservation.
- Improved decoders: incorporating CNN-based refinement modules to enhance boundary accuracy.
- Enhanced multimodal fusion: introducing depth information earlier in the pipeline, for example by integrating RGB-D inputs into the MAE encoder.
- Attention refinement: exploring local attention mechanisms or boundary-aware attention strategies.
- Increased data diversity: expanding the dataset with more varied environments and scene conditions.

These directions may improve the model's ability to capture fine-grained spatial details while maintaining its strong global scene understanding.

7 Conclusion & Future Work

This chapter summarizes the main findings and contributions of this thesis and outlines potential directions for future research in self-supervised perception.

7.1 Summary of Findings

This thesis explores self-supervised visual representation learning for traversability estimation in unstructured outdoor environments. First, a transformer-based Masked Autoencoder (MAE) [1] is used for representation learning, followed by supervised fine-tuning for downstream classification and segmentation tasks at both patch-level and pixel-level.

The experimental results show that the proposed method is effective in capturing global structure and semantic information. The model achieves strong performance in identifying large, continuous traversable regions such as grass fields and open terrain. A comparison with convolutional baselines shows that transformer-based models exhibit strong semantic understanding, CNN-based models perform better on fine-grained spatial details, and the proposed method achieves competitive overall performance.

However, the results also indicate limitations in fine-grained segmentation, particularly in thin structures and complex boundary regions. These limitations are primarily attributed to the patch-based tokenization in the MAE encoder, which reduces spatial resolution within each patch.

7.2 Contributions

The main contributions of this work are as follows:

- A self-supervised representation learning framework for traversability estimation based on Masked Autoencoders is proposed, reducing reliance on labeled data.
- Multiple architectures for traversability prediction are developed, including patch-level classification, pixel-level segmentation, multi-scale supervision, feature pyramid modeling, and LiDAR fusion.

- A comprehensive evaluation of different models is conducted on an outdoor navigation dataset, including quantitative comparisons with baseline methods such as U-Net and DeepLabv3.
- An analysis of transformer-based limitations in fine-grained segmentation is provided, particularly regarding the impact of patch-based tokenization on spatial precision.

These contributions demonstrate the potential of self-supervised transformer models for robotic perception, while also highlighting key challenges that must be addressed for real-world deployment.

7.3 Practical Implications

The findings of this work have important implications for robotic navigation in unstructured outdoor environments.

The proposed models are well-suited for coarse traversability estimation, path planning, and large-scale terrain detection. However, for safety-critical applications that require precise obstacle detection, the current limitations of the approach should be considered, particularly the reduced ability to accurately detect small obstacles.

In practice, the proposed method can be combined with complementary approaches, such as geometry-based obstacle detection or high-resolution perception modules. For example, combining RGB-based segmentation with LiDAR-derived dense height maps may further improve the detection of small obstacles.

7.4 Future Work

Based on the findings of this thesis, several directions for future work can be identified:

Improve Spatial Resolution

The current approach suffers from loss of spatial detail due to MAE patch-based encoding. Future work could explore smaller patch sizes (e.g., 8×8 or 4×4) combined with shallower transformer configurations, hybrid CNN–Transformer architectures with a CNN stem for feature extraction prior to the MAE encoder, or enhanced decoder designs for higher-resolution feature recovery. In addition, more advanced

multimodal fusion strategies could be investigated. Although LiDAR fusion was explored in this work, its effectiveness was limited by late-stage integration. Future approaches may consider early fusion of RGB and depth data, multi-level feature fusion, or dedicated depth encoders.

Real-World Deployment

All experiments in this thesis were conducted on an x86-based computing platform. A natural extension of this work is deployment on real robotic edge systems. This includes real-time inference on embedded hardware, integration with navigation and planning pipelines, and evaluation in real-world outdoor environments.

Uncertainty and Safety

The current work does not explicitly model prediction uncertainty. Future research could investigate uncertainty estimation, confidence-aware prediction mechanisms, and integration with safety constraints for more reliable decision-making.

7.5 Final Remarks

This work demonstrates that self-supervised transformer-based models represent a promising direction for robotic perception in scenarios with limited labeled data. Although challenges remain in preserving fine-grained spatial precision, the results highlight the potential of combining self-supervised learning with modern vision architectures for scalable and robust navigation systems.

References

[1] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16000–16009.

[2] C. Sevastopoulos and S. Konstantopoulos, “A Survey of Traversability Estimation for Mobile Robots,” arXiv:2204.10883, 2022.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Advances in Neural Information Processing Systems (NeurIPS), vol. 25, 2012.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv:1810.04805, 2019.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale,” in International Conference on Learning Representations (ICLR), 2021.

[6] J. Cui, Y. Guo, H. Zhang, K. Qian, J. Bao, and A. Song, “Support Vector Machine Based Robotic Traversability Prediction with Vision Features,” Journal of Intelligent & Robotic Systems, 2013.

[7] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Medical Image Computing and Computer-Assisted Intervention (MICCAI), VOL 9351, Springer, 2015, pp. 234–241.

[8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” arXiv:1706.05587 [cs.CV], Dec. 2017.

[9] M. Oquab, T. Darcet, T. Moutakanti, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve et al., “DINOv2: Learning Robust Visual Features without Supervision,” Transactions on

Machine Learning Research (TMLR), 2024. https://doi.org/10.48550/arXiv.2304.07193.

[10] J.-H. Hwang, J. Bae, D.-W. Kim, Y. Lee, and S.-W. Seo, “From General Vision to Reliable Traversability Estimation: Adapting Vision Foundation Models for Unstructured Outdoor Environments,” arXiv:2605.29565 [cs.RO], 2026.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in International Conference on Machine Learning (ICML), vol. 119, 2020, pp. 1597–1607.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9729–9738.

[13] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 34, 2021.

[14] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, “Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 6877–6886.

[15] Y. Bai, Z. Huang, L. Shen, H. Guo, M. H. Ang Jr, and D. Rus, “Multi-Scale Feature Aggregation by Cross-Scale Pixel-to-Region Relation Operation for Semantic Segmentation,” arXiv:2106.01744, 2021.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” arXiv:1612.03144 [cs.CV], 2016.

[17] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-View 3D Object Detection Network for Autonomous Driving,” arXiv:1611.07759 [cs.CV], 2017.

[18] Y. Chen, Q. Yin, L. Zhao, et al., “Enhancing Long-Range Depth Estimation via Heterogeneous CNN-Transformer Encoding and Cross-Dimensional Semantic

Fusion,” Scientific Reports, vol. 16, 9396, 2026. https://doi.org/10.1038/s41598-026-36755-0.

[19] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, “Learning Rich Features from RGB-D Images for Object Detection and Segmentation,” arXiv:1407.5736 [cs.CV], 2014.

[20] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation with Transformers," IEEE Transactions on Intelligent Transportation Systems, 2023.

[21] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, “Deep Multispectral Semantic Scene Understanding of Forested Environments Using Multimodal Fusion,” Springer, 2016, pp. 465–477. DOI: 10.1007/978-3-319-50115-4_41.

[22] A. Geiger, P. Lenz, and R. Urtasun, “Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[23] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000km: The Oxford RobotCar Dataset,” The International Journal of Robotics Research, vol. 36, no. 1, pp. 3–15, 2017. DOI: 10.1177/0278364916679498.

[24] P. Mortimer, R. Hagmanns, M. Granero, T. Luettel, J. Petereit, and H.-J. Wuensche, “The GOOSE Dataset for Perception in Unstructured Environments,” in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2024.

[25] Z. Zhang, "A Flexible New Technique for Camera Calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[27] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," arXiv:1606.08415 [cs.LG], 2016.

[28] Y. Zhao, G. Zhang, N. Hu, et al., “A Triple-Branch Multi-Scale Network for Real-Time Semantic Segmentation,” Scientific Reports, 2026. https://doi.org/10.1038/s41598-026-48759-x.

[29] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," Ecology, vol. 26, no. 3, pp. 297–302, 1945.

[30] J. Yi, X. Zhong, W. Liu, W. Zhu, Z. Wu, and Y. Deng, "Edge-aware Plug-and-play Scheme for Semantic Segmentation," arXiv:2303.10307, 2023.

[31] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” in International Conference on Learning Representations (ICLR), 2019.

Appendices

The source code used in this thesis, including model implementations and training scripts is publicly available at: https://github.com/l-huo/mae-traversability-segmentation.