Semantic Segmentation for Robot Vision with Synthetic Training Data

Alberto Guiggiani
alberto6@stanford.edu

Abstract

This project focuses on enhancing the semantic segmentation capabilities of the SegFormer model for robot navigation in warehouse environments. By utilizing synthetic data generated through nVidia ISAAC Sim and fine-tuning the model, we aim to improve the segmentation accuracy and reliability essential for autonomous robot operations. The results demonstrate a significant increase in mean Intersection over Union (mIoU) from 0.43 to 0.70, highlighting the effectiveness of synthetic data in training advanced segmentation models. Key improvements include better differentiation between similar objects and the introduction of new classes critical for navigation, such as forklifts.

1. Introduction

Image segmentation is a fundamental problem in computer vision that involves dividing an image into multiple segments or regions to simplify the representation and make it more meaningful for analysis. It is pivotal in tasks like object recognition, medical imaging, and autonomous driving, where understanding the spatial organization of different objects within an image is crucial [3, 4].

One foundational work that has been frequently cited in the context of image segmentation is Long et al.'s introduction of Fully Convolutional Networks (FCNs) for semantic segmentation [6]. This paper revolutionized the field by adapting CNNs for pixel-wise prediction without any fully connected layers, which enabled end-to-end training and inference on images of arbitrary sizes. This approach has laid the groundwork for many subsequent developments in image segmentation methodologies.

The introduction of Transformer models has brought a new perspective to handling image segmentation tasks, traditionally dominated by convolutional networks. The SegFormer paper by Xie et al. [7] integrates the Transformer architecture specifically tailored for the demands of semantic segmentation. SegFormer stands out for its hierarchical Transformer encoder which efficiently processes multiscale features, crucial for capturing detailed context at various resolutions necessary for accurate segmentation.

Another significant contribution is the Swin Transformer by Liu et al. [5], which constructs hierarchical feature maps and applies shifted windows for self-attention, enhancing modeling capability and efficiency for various vision tasks, including semantic segmentation.

As the field progresses, more innovations continue to emerge, such as the integration of Transformer models with conventional CNNs to leverage the strengths of both architectural paradigms [1, 2]. These advancements underscore the dynamic nature of the field and the ongoing efforts to improve the accuracy and efficiency of image segmentation models.

In this project, we focus on enhancing the semantic segmentation capabilities of the SegFormer model for robot navigation in warehouse environments. By utilizing synthetic data generated through nVidia ISAAC Sim and finetuning the model, we aim to improve the segmentation accuracy and reliability essential for autonomous robot operations.

2. Problem Statement

In the context of enhancing autonomous robot navigation within warehouse environments, this project addresses the critical task of semantic segmentation.

- **Objective:** Enable autonomous robots to navigate efficiently and safely in complex and dynamic warehouse settings.
- **Input:** Images captured by the robot's camera while navigating the warehouse.
- **Output:** Segmented images where each pixel is labeled with the class of the object it represents, allowing the robot to understand and interpret its surroundings.
- **Challenges:**
  - **Accurate Navigation:** Robots need to accurately identify paths, obstacles, and relevant items within the warehouse.
  - **Dynamic Environments:** Warehouses are dynamic with frequently changing layouts and ob-

jects, requiring robust and adaptable segmentation models.

- **Data Acquisition:** Obtaining a large, diverse, and accurately labeled dataset for training is often expensive and time-consuming.

The aim is to achieve a high level of precision in image segmentation to ensure robots can navigate effectively, avoiding obstacles and identifying essential items reliably.

3. Methodology Overview

To address the challenges identified, this project utilizes the following methods:

- **SegFormer Model:** Leveraging the SegFormer model, which is renowned for its efficiency and effectiveness in semantic segmentation tasks. The model processes multi-scale features, crucial for capturing detailed context at various resolutions necessary for accurate segmentation.

- **Synthetic Data Generation:** Utilizing high-fidelity graphical simulations through nVidia ISAAC Sim to generate large volumes of labeled training data. This approach mitigates the challenges of acquiring real annotated images in specialized settings by producing high-quality, diverse synthetic datasets.

- **Model Fine-Tuning and Performance Enhancement:** Fine-tuning the SegFormer model on the synthetic data to improve its performance efficiently. This strategy leverages the controlled conditions of simulated data to enhance the model's generalization capabilities in real-world scenarios.

4. The SegFormer Model for Semantic Segmentation

SegFormer is an advanced semantic segmentation model that leverages the power of Transformers to achieve efficient and accurate segmentation results. The architecture, as proposed by Xie et al. [7], integrates several innovative design elements to enhance performance and scalability.

4.1. Architecture Overview

The SegFormer architecture (shown in Figure 1) consists of two main components: the encoder and the decoder. The encoder is responsible for extracting multi-scale features from the input image, while the decoder processes these features to produce the final segmentation map.

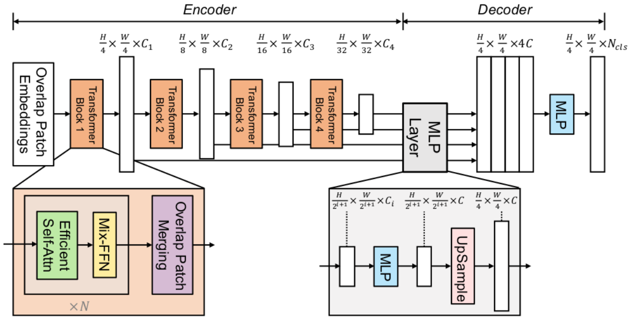

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 1. Architecture of the SegFormer model, highlighting its hierarchical Transformer encoder and efficient design. Source: Xie et al. [7]

4.1.1 Encoder

The encoder in SegFormer uses a hierarchical structure of Transformer blocks, which allows it to process input images at multiple scales. This multi-scale feature extraction is crucial for capturing both fine details and broader contextual information. Each stage in the encoder consists of:

- **Overlap Patch Embeddings**: Converts the input image into a sequence of patches with overlapping regions to retain spatial information.

- **Transformer Blocks**: Processes the patches using self-attention mechanisms and feed-forward networks to extract rich features.

- **Patch Merging Layers**: Reduces the spatial dimensions of the feature maps while increasing the depth, enabling efficient multi-scale representation.

4.1.2 Decoder

The decoder aggregates the multi-scale features from the encoder and refines them to produce a high-resolution segmentation map. Key components of the decoder include:

- **MLP Layers**: Multi-Layer Perceptron (MLP) layers are used to merge features from different scales.

- **UpSampling Layers**: These layers increase the spatial resolution of the feature maps to match the input image size, facilitating accurate pixel-wise predictions.

4.2. Benefits and Performance

The SegFormer model is designed to be both simple and efficient, making it suitable for real-time applications in environments with limited computational resources. Key benefits include:

- **High Efficiency**: The use of lightweight Transformer blocks and efficient patch embedding techniques ensures that SegFormer can process images quickly without compromising on accuracy.
- **Scalability**: The hierarchical design allows SegFormer to scale effectively, handling images of various sizes and resolutions.
- **Robustness**: The multi-scale feature extraction and merging capabilities make SegFormer highly robust to variations in object sizes and complex scene layouts.

In their experiments, Xie et al. [7] demonstrated that SegFormer achieves state-of-the-art performance on several benchmark datasets, including ADE20K and Cityscapes, outperforming many existing models in terms of both accuracy and inference speed.

5. Dataset

The dataset creation is a pivotal part of this project, focusing on generating high-quality synthetic data for training the SegFormer model tailored for warehouse navigation. The following subsections describe the environment setup, data capture, and preprocessing steps involved in creating the dataset.

5.1. Environment Setup and Data Capture

A detailed 3D warehouse environment was constructed using nVidia ISAAC Sim on an Ubuntu workstation equipped with a Geforce RTX graphics card. The realism of the environment was enhanced by the Warehouse Creator Extension, which supports hyper-realistic, ray-traced scenes.

An animation of a simulated robot camera was defined to navigate around the warehouse, capturing images from various angles and positions. This setup allows for the generation of ground-truth data for semantic segmentation, crucial for the training and evaluation of the SegFormer model.

See Figure 2 for a screenshot of the ISAAC environment. Top-left panel shows a realtime, ray-traced rendering of the 3D scene. On the bottom right we use the “Synthetic Data Recorder” tool to automatically generate the ground-truth segmentation data during the animation. The latter is controlled by the timeline at the bottom, where we defined a path of the robot around the warehouse via a series of keyframes.

5.2. Dataset Composition

The dataset comprises 100 elements, where each element includes:

- A 3-channel (RGB) rendering of the scene with dimensions 3x512x512.
- A 512x512 tensor representing the semantic segmentation ground truth, where each pixel’s value is a class ID corresponding to the object present in that pixel.
- A JSON file mapping each class ID to its respective class name, providing an understandable and traceable reference for each object in the images.

See Figure 3 for some example RGB images from the dataset, as well the ground-truth segmentation data auto-generated by the Synthetic Data Recorder tool. Notice how the images look very realistic thanks to the advanced rendering capabilities of ISAAC Sim with ray-tracing. Also note the complexity of the scenes, which would require a lengthy and expensive manual labeling process.

5.3. Preprocessing and Class Mapping

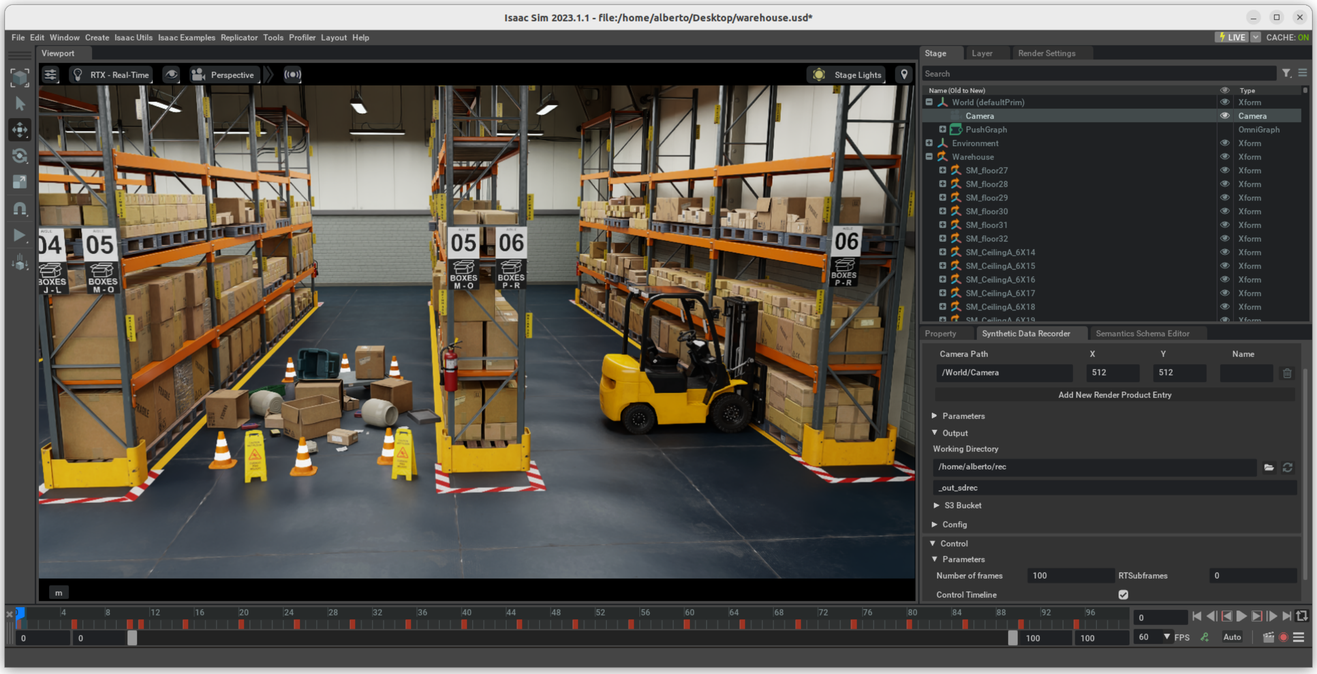
One of the more complex aspects of the dataset creation was the preprocessing of segmentation data. Each captured image came with its unique mapping from IDs to objects, which varied due to the presence of different objects across the images. To standardize this, an extensive matching process was undertaken to align each object with one of the 150 classes defined in the ADE20K dataset [8], supplemented by additional classes to cover new objects found in the warehouse environment. This was done because we chose as baseline the SegFormer model trained on ADE20K.

Ultimately, a unified dictionary was created where each “ID” was associated with a “class” name, totaling 165 unique classes. This standardized mapping ensured that all processed segmentation data had class IDs consistent with this single dictionary, simplifying the training and validation process for the SegFormer model.

Here follows a list of all the classes present in the warehouse dataset. In bold the new classes not available in the ADE20K dataset that the fine-tuned model will have to learn from scratch.

BACKGROUND
rack
paper_note
UNLABELLED
ceiling
sign
crate
lamp
bottle
bracket
forklift
cart
paper_shortcut
emergency_board

box
pillar
pallet
floor_decal
floor
fire_extinguisher
wall
wire
barel
fuse_box
cone
bucket
barcode



Isaac Sim 2023.1.1 - file:/home/alberto/Desktop/warehouse.usd*

Name (Old to New)	Type
World (defaultPrim)	Xform
Camera	Camera
PushGraph	OmniGraph
Environment	Xform
Warehouse	Xform
SM_floor27	Xform
SM_floor28	Xform
SM_floor29	Xform
SM_floor30	Xform
SM_floor31	Xform
SM_floor32	Xform
SM_CeilingA_6X14	Xform
SM_CeilingA_6X15	Xform
SM_CeilingA_6X16	Xform
SM_CeilingA_6X17	Xform
SM_CeilingA_6X18	Xform

Camera Path	X	Y	Name
/World/Camera	512	512	

Working Directory: /home/alberto/rec

_out_sdrec

Number of frames: 100

RTSubframes: 0

Control Timeline: ☑

Figure 2. Warehouse 3D environment using nVidia ISAAC Sim.

6. Finetuning the SegFormer Model

Finetuning the SegFormer model was a crucial part of this project, aimed at optimizing the model's performance for semantic segmentation in warehouse environments. The finetuning process was implemented in PyTorch and executed on a fully-specced Apple MacBook Pro with an M3 Max chip, which includes 40 GPU cores and 128GB of unified memory. Leveraging the MPS backend of PyTorch allowed us to efficiently utilize all GPU cores and most of the available memory.

6.1. Training Setup

The finetuning process spanned approximately one week, during which a total of 18 training attempts were conducted. The training experiments were meticulously logged using TensorBoard, as illustrated in Figure 4. Each training run explored different hyper-parameters and dataset augmentations to identify the optimal configuration.

6.2. Optimization and Augmentation

We utilized the AdamW optimizer in PyTorch, experimenting with various learning rates ranging from 1×10^{-7} to 1×10^{-4}, and stability parameter ϵ values from 1×10^{-8} to 1×10^{-6}. Additionally, random dataset augmentations were applied, including color jitter, rotations, and flips.

6.3. Finetuning Results and Observations

Figure 5 shows the train and test loss curves for the best finetuned model, obtained after 80 epochs. This optimal result was achieved with a learning rate of 2×10^{-5} and $\epsilon = 1 \times 10^{-8}$, without applying vertical flips or rotations to the dataset. This decision was based on the necessity to preserve the concept of "up" and "down," which provides critical information for distinguishing between the ceiling and the floor in warehouse environments.

In some training attempts, we observed a significant discrepancy between test and train loss, indicating potential overfitting. Additionally, prolonged training beyond a certain number of epochs led to performance deterioration and instability. Therefore, for the best model, we aimed to achieve a clean and consistent decrease in both train and test losses, with the test loss slightly higher but not excessively so.

The TensorBoard metrics provided valuable insights into the finetuning process, highlighting the importance of balanced training and the careful selection of hyper-parameters to avoid overfitting while achieving robust performance improvements.

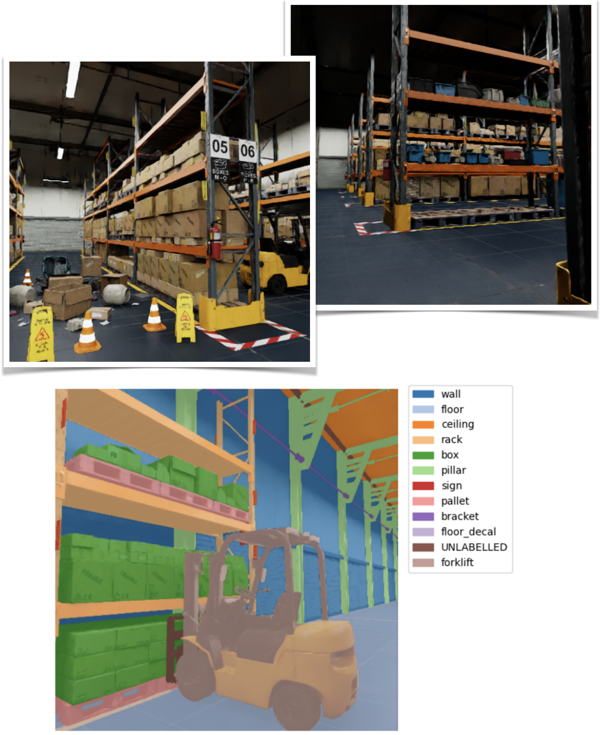


Figure 3. Samples from the generated synthetic dataset. Top: RGB images, showcasing the photo-realistic fidelity of ISAAC Sim. Bottom: overlay with ground-truth segmentation data generated via the Synthetic Data Recorder tool.

7. Results

This section presents the evaluation results of the SegFormer model before and after finetuning on the synthetic warehouse dataset. We utilize the mean Intersection over Union (mIoU) metric to quantitatively assess the model's performance.

7.1. Mean Intersection over Union (mIoU)

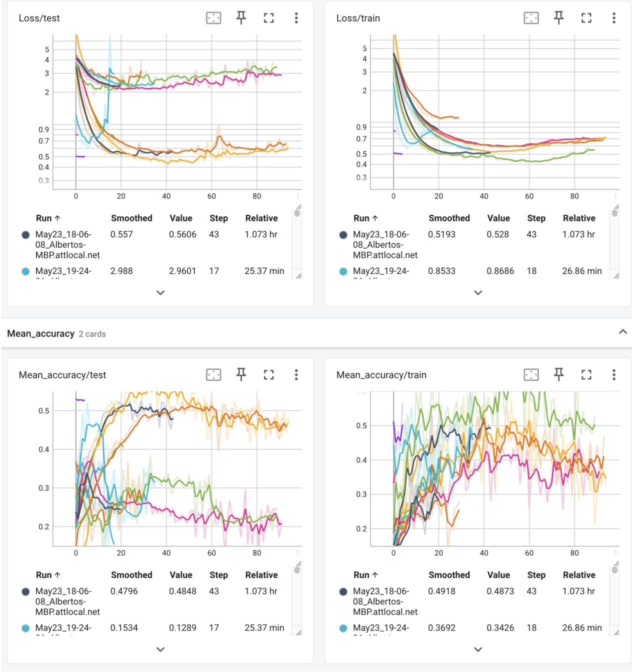
The mIoU metric is a common evaluation measure for semantic segmentation tasks. It calculates the average IoU across all classes, where IoU is defined as:

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} \tag{1}$$

For each class, the intersection is the area of overlap between the predicted segmentation and the ground truth, and the union is the total area covered by both the predicted segmentation and the ground truth. The mIoU is then computed as the mean of the IoUs for all classes.

7.2. Overall Performance

Figure 6 shows the inference performance on the entire test dataset, comparing the baseline SegFormer model (red) and the finetuned model (blue). The finetuned model demonstrates a significant improvement in mIoU, increasing from an average of 0.43 to 0.70.



Loss/test

Run ↑	Smoothed	Value	Step	Relative
● May23_18-06-08_Albertos-MBP.attlocal.net	0.557	0.5606	43	1.073 hr
● May23_19-24-	2.988	2.9601	17	25.37 min

Loss/train

Run ↑	Smoothed	Value	Step	Relative
● May23_18-06-08_Albertos-MBP.attlocal.net	0.5193	0.528	43	1.073 hr
● May23_19-24-	0.8533	0.8686	18	26.86 min

Mean_accuracy 2 cards

Mean_accuracy/test

Run ↑	Smoothed	Value	Step	Relative
● May23_18-06-08_Albertos-MBP.attlocal.net	0.4796	0.4848	43	1.073 hr
● May23_19-24-	0.1534	0.1289	17	25.37 min

Mean_accuracy/train

Run ↑	Smoothed	Value	Step	Relative
● May23_18-06-08_Albertos-MBP.attlocal.net	0.4918	0.4873	43	1.073 hr
● May23_19-24-	0.3692	0.3426	18	26.86 min

Figure 4. Tensorboard metrics for the 18 fine-tuning attempts with different hyper-parameters and dataset augmentations. Notice the attempts with a large discrepancy in train loss (top-right) vs test loss (top-left), likely indicating overfitting.

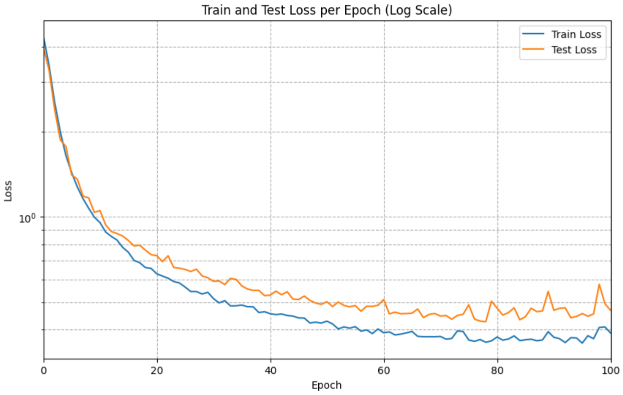


Figure 5. Train vs. Test loss for the best finetuned model, obtained at Epoch 80.

7.3. Specific Image Analysis

To further illustrate the improvements achieved through finetuning, Figure 7 presents the segmentation results for a specific test image. The finetuned model shows notable improvements in key areas:

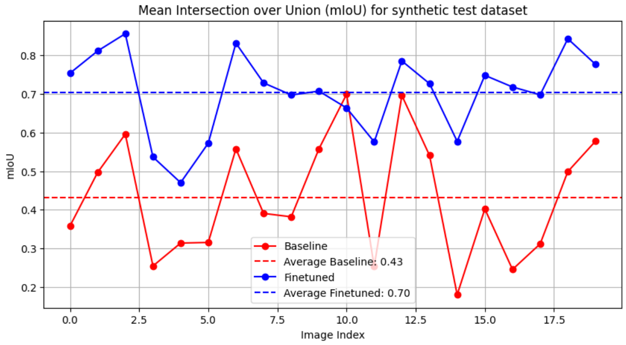


Figure 6. Inference performance on test dataset, comparing baseline SegFormer (red) vs. SegFormer fine-tuned on the warehouse dataset (blue).

- **Shelf Segmentation**: The finetuned model more accurately differentiates between boxes, pallets, and racks on the shelves. This distinction is crucial for robotic navigation and manipulation tasks within the warehouse.
- **Forklift Detection**: The finetuned model successfully detects the forklift, a new class introduced during finetuning. Detecting such obstacles is essential for safe navigation and operation within the environment.

It is interesting to note that classes such as boxes and racks were already present in the ADE20K dataset, which the baseline SegFormer model was trained on. Despite this, finetuning with the synthetic warehouse dataset led to significant improvements, highlighting the effectiveness of the finetuning process.

7.4. Case of Decreased Performance

While the finetuned model generally outperforms the baseline, there is one instance at index 10 where the finetuned mIoU is lower. Figure 8 compares the baseline and finetuned models for this specific image. In the finetuned model, the wire on the wall is misclassified as a pillar, and there is slightly more noise in the segmentation of the bottom boxes. However, the finetuned model correctly classifies the floor decal, which is essential for navigation, and the pallet on the bottom right.

Despite the lower mIoU for this particular image, the improvements in key areas for robotic operation in a warehouse, such as accurate floor decal detection and pallet recognition, demonstrate the overall effectiveness of the finetuning process.

8. Conclusion

In this project, we successfully enhanced the semantic segmentation performance of the SegFormer model for robot navigation within warehouse environments. By leveraging synthetic data generated through high-fidelity simulations and implementing an extensive fine-tuning process, we significantly improved the model's mean Intersection over Union (mIoU) from 0.43 to 0.70. The finetuned model exhibited notable advancements in accurately differentiating between similar objects and detecting new classes essential for navigation, such as forklifts.

Our findings underscore the potential of synthetic data in training sophisticated models for complex tasks where real-world annotated data is scarce or expensive to obtain. The success of this approach sets a precedent for future research in robotics and computer vision, demonstrating that synthetic data, when properly utilized, can effectively enhance model performance in real-world applications.

Future work will focus on further optimizing the model, exploring additional synthetic data generation techniques, and testing the model in more diverse and challenging real-world environments to ensure its robustness and generalizability.

References

[1] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Lin, L. Wang, and D. Tao. Pre-trained image processing transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12309, 2021.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[5] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[6] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[7] 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*, 2021.

[8] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision (IJCV)*, 127(3):302–321, 2019.

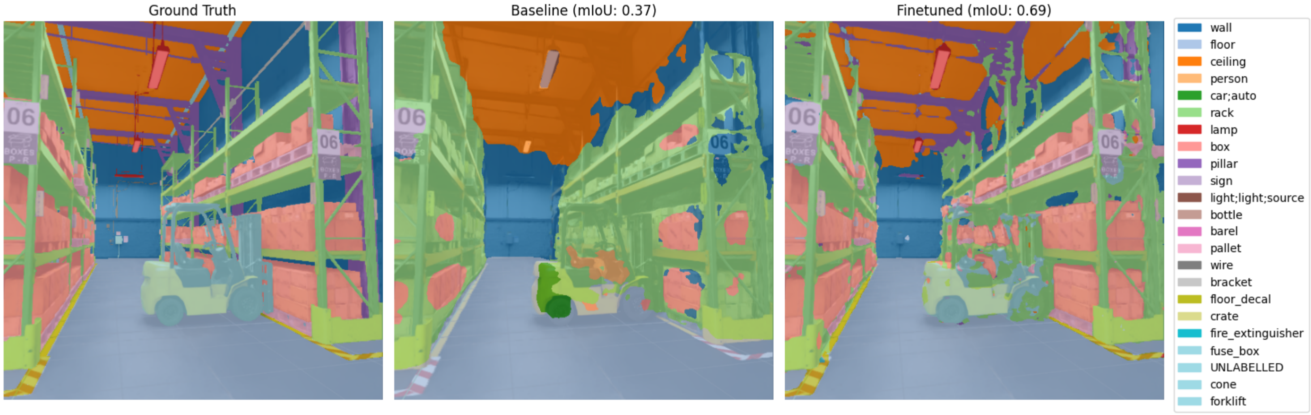


Figure 7. Inference results on challenging image. Left: ground truth. Middle: baseline SegFormer. Right: fine-tuned SegFormer (ours). A key area of improvement is discerning between boxes, pallets, and racks on both shelves. Other notable areas are the forklift, as well as the separation of pillars and lamps from the ceiling.

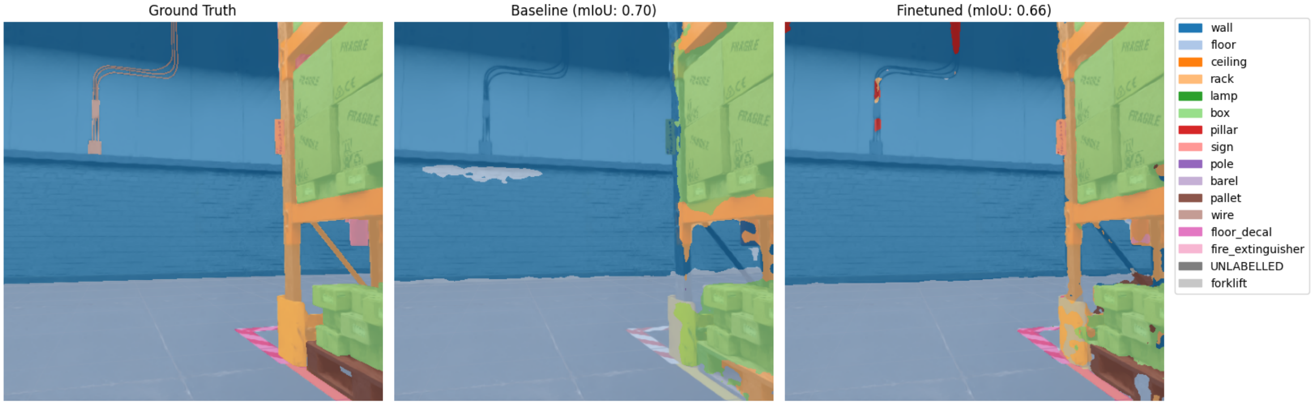


Figure 8. Comparison of segmentation results for a specific test image where the finetuned model performed worse. Left: Ground Truth. Middle: Baseline SegFormer. Right: Finetuned SegFormer.