Indoor Scene Understanding via 2D and 3D Semantic Segmentation: Integrating Depth for Geometry-Aware Reconstruction

Karthik Pythireddi
Stanford University
karthik9@stanford.edu

Arya Bakthiar
Stanford University
aryabakthiar@stanford.edu

Abstract

Understanding of the indoor scene benefits from combining the appearance of a frame with volumetric geometry. We first benchmark with off the shelf 2D segmentaion models - DeeplabV3+ and Segformer-B0 in a Zero-Shot setting on ScanNet Data, then refine one of the models by incorporating depth as an additional input channel—effectively leveraging 3D geometry cues to improve per-frame segmentation. Finally, we fuse all RGB-D frames into a volumetric (3D) representation and train a 3D U-Net to predict voxel-level semantic labels, comparing the performance when using either ground-truth sensor depth or self-calibrated depth from MASt3R. Our experiments reveal how the addition of geometric depth information can boost segmentation accuracy in 2D.

1. Introduction

The Indoor Scene understanding is a fundamental problem in the computer vision with latest applications in the robotics and augmented reality. Since the introduction of RGB-D sensors, such as Microsoft Kinect, the capture and analysis of indoor 3D scenes [18] has gained significant attention, opening up a wider range of details. However, the higher level of semantic understanding in cluttered indoor environments remains challenging. A key insight from recent research is that combining 2D appearance cues with 3D geometric information can substantially enhance the interpretation of the scene. Depth data provide complementary shape information that helps in distinguishing objects and surfaces, addressing the limitation of RGB based vision [4]. In particular, geometric depth cues enhance the delineation of object boundaries and remain robust under conditions such as low illumination or similar textures where RGB-based methods struggle.

Over the last 10 years, we have seen significant advances in semantic segmentation driven by large-scale 2D benchmarks like Cityscapes [20] and ADE20K [27]. The state of the art CNN architectures like DeepLabV3+ have achieved higher pixel-level accuracy by combining multi-scale contextual features with the encoder-decoder enhancement [25]. Transformer-based models such as Segformer [27] further improve performance with efficient attention blocks. Models trained on generic 2D datasets often face the issue of domain gaps compared to indoor scenes. Hence, RGB-D datasets like ScanNet [2] have gained traction, which contains approximately 1500 scans of various indoor environments. This data set comprises the RGB, depth, pose, intrinsics, and semantic information about the particular scene.

In this project, we explore how geometric depth cues and volumetric representations can enhance the semantic understanding of Indoor Scenes. We benchmarked off-the shelf DeepLabV3+ and Segformer-B0 on ScanNet in a Zero-Shot setting and found that DeepLabV3 + outperformed Segformer-B0 in multiple apartment scenes. The fine-tuning segformer showed an improvement in the segmentation quality, although the validation data leak affected the metric reliability. To asses the role of geometry, we comapared the segmentation performance using the ground truth depth from ScanNet and self-calibrated depth from MASt3R [7]. Despite the minimal accuracy gaps, MASt3R showed the competitive pose and depth estimation, and 3D segmentation with voxel-level labeling using a 3D U-Net architecture demonstrated that incorporating depth significantly improves metrics such as mIoU, Dice, and Precision. Our findings highlight the value depth-aware representations in indoor scene understanding and MASt3R's viability for calibration-free reconstruction pipelines.

2. Background and Related Work

Our work builds on the idea of combining the appearance of geometry for scene segmentation, starting from per-frame RGB segmentation and extending into the volumetric domain. We first benchmark the off-the shelf 2D semantic segmentation models like DeepLabv3+ [25] and Segformer-B0 [27] on the ScanNet [2]dataset. The models

provide insight into how well 2D pretrained networks generalize to the indoor environments. Next, we enhance one of these models by introducing depth as an additional input channel to create a RGB-D segmentation network, leveraging 3D cues at the image frame level. Similar efforts have shown the depth encoding improves object boundary segmentation [4].

To move beyond the predictions per image frame, we construct a global volumetric representation by fusing RGB-D frames. We adopt the 3D U-Net [21] architecture to predict voxel-level semantic labels in this volumetric space. This allows the network to leverage spatial continuity and context beyond individual frames. Prior work such as SemanticFusion [8], ElasticFusion [12] and PanopticFusion [9] demonstrated the importance of fusing appearance and geometry for real-time 3D scene mapping. We also compare the impact of different depth inputs like sensor ground truth from ScanNet Dataset, estimated depth from the MASt3R [7] pipeline to understand how the depth quality affects the segmentation performance. Our experiments show the accurate geometric information not only improves per-frame results but also enhances the consistency and completeness of 3D semantic maps.

3. Methods

In this work, several segmentation experiments were conducted to investigate the semantics of indoor scenes. First, we evaluated an off-the-shelf DeepLabV3+ model to gauge the effectiveness of a purely 2D segmentation approach compared to our volumetric methods. Then we went ahead to evaluate the 2D segmentation on the Segformer-B0 which is a transformer-based model.

Second, we trained a standard 3D U-Net on fused volumetric data using both ground-truth and MASt3R-predicted depth to compare the semantic accuracy with and without explicit calibration, providing a direct performance reference for our integrated geometry semantic pipeline.

3.1. 2D Segmentation - Zero-shot

We employed an off-the shelf DeepLabV3+ model [25], originally trained on the COCO dataset [26], for 2D semantic segmentation on ScanNet [2] in a zero shot manner. Each input was a single 512×512 RGB image, and the model produced a 2D segmentation mask on 41 labeled ScanNet Categories. We chose DeepLabV3+ for its strong performance in large-scale segmentation tasks and its coverage of diverse object categories, many of which overlap with ScanNet's labeled class categories.

We applied the ScanNet dataset to a Segformer-B0, which is a lightweight transformer-based encoder-decoder model pre-trained on the ADE20K [27] dataset. We ran it directly on each ScanNet RGB frame to produce the 2D mask over the 41 ScanNet classes. The input RGB image is given in the size of 512×512 using the SegformerImageProcessor function.The logits we calculated are upsampled to 512×512,so that the arg-max yields a label map per pixel in the 41-class space of ScanNet. We also evaluated the fine-tuning of Segformer-B0 on the ScanNet dataset by tweaking the hyperparameters.

3.2. 2D Segmentation - Finetuning

We began with a pre-trained SegFormer-B0 model, originally trained on the ADE20K dataset for three-channel (RGB) segmentation across 150 classes. We formed a four-channel input (R, G, B, Depth) by concatenating the depth map to the RGB data. Because the original SegFormer-B0 backbone handles only three channels, we introduced an additional Conv2D embedding layer for the depth channel and updated the model configuration to a four channel input. We froze the backbone for the initial five epochs to stabilize learning, then unfroze it to fine-tune the feature extractor for the next 20 epochs. To further sharpen segmentation at object boundaries, we included a boundary-aware loss term leveraging the Sobel operator on the depth channel. This two-stage freeze–unfreeze training regimen improved overall stability while adapting the entire network to the ScanNet data.

3.3. 3D U-Net for 3D Segmentation

We train a standard 3D U-Net on volumetric data, which is obtained by fusing each ScanNet RGB-D sequence into a four-channel truncated signed distance function (TSDF) grid at a resolution of 4.8 cm voxel, following common practice in ScanNet [2], using a depth limit of 5.0 m and a truncation threshold of 5.0 cm. The input to the 3D U-Net is a 4D tensor with dimensions $(4, D, H, W)$, where 4 corresponds to the channels (R, G, B, O) and (D, H, W) are the spatial dimensions with a resolution of 4.8 cm. The channels (R, G, B) represent the color values, and O represents the occupancy of the voxel.

The 3D U-Net is configured with five downsampling stages, residual blocks, and skip connections, and is trained with a combined Dice and cross-entropy loss. The network outputs a 4D volumetric tensor of voxelwise logits (N, D, H, W), where N is the number of classes. Two training variants are compared: one using ground-truth depth images to build the TSDF volume, and another relying on MASt3R-generated depth maps [7] to evaluate the effect of self-calibrated geometry on segmentation accuracy.

To generate the 3D voxel representation, we use the Open3D [1].Using RGB images, camera poses, and depth maps, we load the intrinsic matrix from the file and extract the focal lengths (f_x, f_y) and the main point (c_x, c_y). We create a Scalable TSDF Volume object in Open3D by specifying voxel size and truncation distance, enabling global 3D reconstruction.

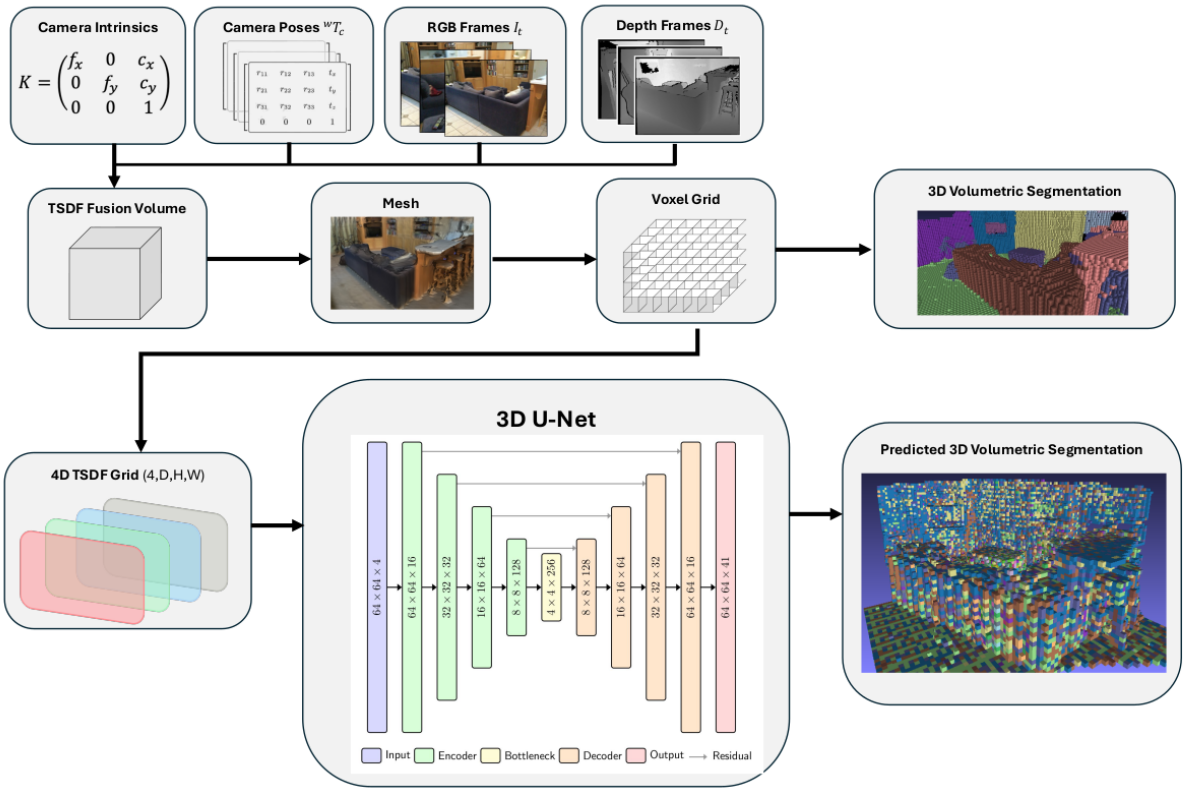

Figure 1. 3D U-net architecture, the numbers shown are the output dimenston of each layer.

For each frame of color, depth and pose, we convert the data into an Open3D RGBDImage and integrate it into the TSDF volume. This maintains a globally consistent 3D scene representation [18]. Across all frames, we fuse the scene into one 3D volume, leveraging known pose transformations. Each point map $X \in \mathbb{R}^{W \times H \times 3}$ associates the pixel (u, v) with a 3D point $X_{u,v}$ via:

$$X_{u,v} = D_{u,v} K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{1}$$

To ensure consistent mapping of 2D pixels to 3D points across multiple viewpoints, we express points from camera n to m, we use pose matrices $P_n, P_m \in \mathbb{R}^{3\times4}$:

$$X^{n,m} = P_m(P_n^{-1} \cdot h(X^n)), \quad \text{where } h(x) = \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{2}$$

During TSDF integration, the per-voxel color is updated with a running average. Once complete, a triangle mesh is extracted and then sampled into a uniform voxel grid preserving the origin and voxel size.

To assign voxel-level semantic labels, we project each voxel into 2D segmentations of all frames. The center of the voxel is transformed into the camera coordinate frame using inverse pose, projected using intrinsics, and validated by comparing depth values. The class label is added to a histogram, and the majority label is assigned per voxel.

Finally, the 4D volume (R, G, B, Occupancy) and 1D label volume are compiled as input for the 3D semantic segmentation model.

4. Dataset and Features

Our primary data set comprises the ScanNet Dataset [2],a large-scale collection of RGB-D video sequences from indoor environments. We selected three distinct apartment scenes to serve as our main training and evaluation environment, designated as Apartment 1 (5578 frames), Apartment 2 (1988 frames) and Apartment 3 (4439 frames).Each frame in these scenes consists of corresponding RGB images, depth data, camera pose information, camera intrinsics, and semantic labels. Figure 2 shows an example of RGB-D frames from Apartment 1 with their corresponding overlay segmentation and Figure 3 the overall mesh. The original ScanNet RGB images have a resolution of 1296 × 968, while depth data were at a lower resolution of 640 × 480. To account for this difference for model training and evaluation, we modified the resolution of the depth maps

to match the RGB image resolution using nearest- neighbor interpolation. For each apartment, we normalized each RGB channel to achieve standardized pixel intensities. For simplicity, we follow the NYU40 standard [4], which collapsed the hundreds of ScanNet class labels into 41 classes (the label 0 being void). Similarly, SegFormer was pretrained on the ADE20K dataset (150 classes) and DeeLabV3+ (21 classes) was trained on the Pascal 21 dataset. We used the ScanNet spreadsheet of label mappings to construct segmentation images in the NYU40 label space to ensure consistent class definitions and manageable training. We performed the same dictionary mapping for SegFormer and DeepLabV3+. This mitigates problems such as class imbalance and noisy tail categories that arise when training hundreds of classes. To enrich the training set, we ran MASt3R [7] on every consecutive frame pair in each apartment scene, recorded its dense depth estimates and refined relative poses, and compared these predictions with the original ScanNet annotations. We generated these additional 1296 × 968 depth maps and poses so that our 3D segmentation network could train on MASt3R's geometry and so we could later quantify how the accuracy of MASt3R's predictions influences downstream semantic performance. Figure 5 shows an example comparison of a depth map generated by MASt3R compared to the original ScanNet Image frame.

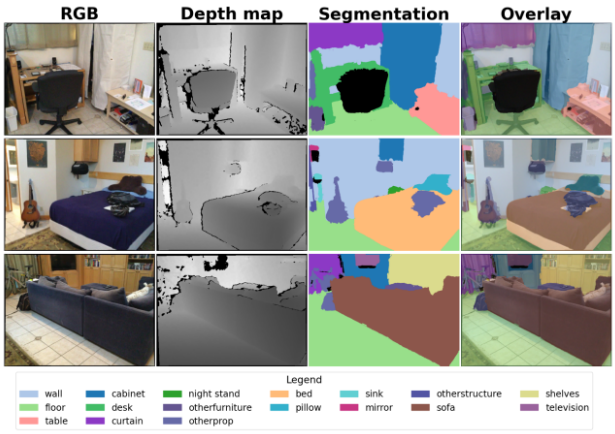

Figure 2. **Scene-based segmentation.** From left to right, each row shows original RGB image, depth map, color-coded semantic segmentation via NYU40 labels, and overlay of segmentation atop RGB image. Best viewed zoomed-in in electronic version.

Together, our complete data set is roughly 200 GB in size. In our model experiments, the dataset for each apartment was divided into 90% for training and 10% for validation. For our 2D segmentation experiments, we randomly assigned individual RGB-D images to training and validation sets. For 3D segmentation, we had to divide the volumetric representation of each scene itself, masking out specific portions of the volume for training and reserving different, non-overlapping regions for validation to ensure that the validation set covers spatial areas unseen by the model during training. Figure[5] illustrates a sample volumetric segmentation from Apartment 1 incorporated in the training pipeline.

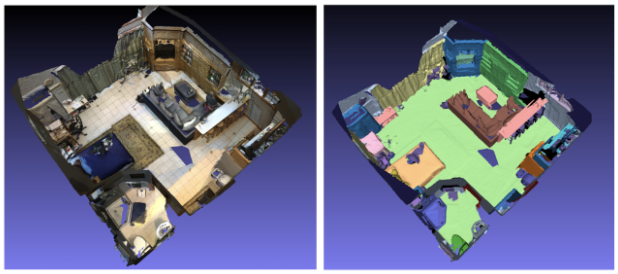

Figure 3. **Apartment Mesh.** Visualization of surface mesh (left) and corresponding semantic labels (right) for Apartment 1.

5. Experiments

5.1. Experimental Setup

For our Segformer-B0 fine-tuning, we applied the input data comprising the 4-channel RGB-D tensor. The modification we did for fine tuning was to replace the first convolution layer to accept 4 channels instead of 3 channels, Initialize the 4th Channel weight as the mean of RGB weights. For the backbone, we used the learning rate as 1×10^{-5} and for the decoder, we used 5×10^{-4}, we set the batch size in the configuration to be 2 and the total number of epochs to be 25.

To adapt it for ScanNet, we applied random (512×512) crops to the RGB, depth, and label images. During validation, we used the same (512×512) center-crop strategy. The 3D U-Net was trained in PyTorch [23] using the open-source MONAI framework [19] and configured with an encoder–decoder depth of five stages (channel widths of 16, 32, 64, 128, and 256; stride of 2 at each downsampling), providing a receptive field large enough to capture room-scale context while keeping the memory footprint manageable. Input volumes were cropped to $64 \times 64 \times 64$ voxels and stacked into four channels (RGB + TSDF), with a batch size of 8. Optimization was performed with Adam [24] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and an initial learning rate of 1×10^{-4}, a value that proved stable during a preliminary sweep over $\{1 \times 10^{-5}, 1 \times 10^{-3}\}$. The model was trained for 100 epochs.

5.2. Metrics

To enable consistent evaluation across all three experiments, we adopt a shared set of metrics for semantic segmentation performance which include Cross-Entropy Loss, Dice Loss, and Dice-Cross-Entropy Loss (DCE Loss) for

training objectives, as well as mean Intersection-over-Union (mIoU), F1 Score, Accuracy, Precision, Recall, and Dice Coefficient for evaluation.

Cross-entropy loss promotes accurate voxel-level classification by penalizing incorrect predictions, while dice loss emphasizes spatial overlap between predicted and ground-truth labels, making it well-suited for segmentation tasks with class imbalance. To jointly capture both aspects we use the combined DCE Loss defined:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbf{1}\big(y_i = c\big)\,\log p_{i,c} \tag{3}$$

$$\mathcal{L}_{\text{Dice}} = 1 \;-\; \frac{2\sum_{i=1}^{N} p_i\, y_i \;+\; \epsilon}{\sum_{i=1}^{N} p_i \;+\; \sum_{i=1}^{N} y_i \;+\; \epsilon} \tag{4}$$

$$\mathcal{L}_{\text{DCE}} = \lambda_1\,\mathcal{L}_{\text{CE}} + \lambda_2\,\mathcal{L}_{\text{Dice}} \tag{5}$$

We propose to extend MASt3R's geometric reconstruction pipeline by adding a lightweight 3D decoder that predicts per-voxel class labels directly in the volume. In particular, we append a UNet-style decoder to the multi-scale feature representation learned by MASt3R, and we train this decoder on 20 object categories provided by the ScanNet dataset. We employ a cross-entropy loss on each voxel logit, which is defined as

where N is the total number of voxels in a batch, p_i and y_i are the predicted and ground truth labels for the pixel (or voxel) i, ϵ is a small constant to avoid division by zero, and λ_1, λ_2 balance the contribution of the two terms. This loss has been shown to be effective for volumetric segmentation tasks, particularly in class-imbalanced settings [22]. For our setup, $\lambda_1 = \lambda_2 = \frac{1}{2}$.

The mean intersection-over-union (mIoU) quantifies the degree of spatial overlap between predicted regions and ground-truth labels across all classes. Unlike accuracy, which can be misleading in class-imbalanced settings, mIoU evaluates both false positives and false negatives, thus penalizing over-segmentation and under-segmentation. This makes it important to evaluate the consistency per class and the overall quality of segmentation in dense scene understanding tasks such as 3D reconstruction indoors [20, 2]. For class c, IoU is defined as:

$$\text{IoU}c = \frac{\text{TP}c}{\text{TP}c + \text{FP}c + \text{FN}_c} \tag{6}$$

Both mIoU and Dice-based losses are particularly appropriate for our task given their robustness to class imbalance and their emphasis on spatial overlap, which are essential in dense 3D scene understanding tasks [22]. In addition to evaluating semantic segmentation, we compare intermediate predictions of MASt3R against the ground truth of ScanNet to asses depth and pose precision, which are critical for reliable 3D semantic understanding [2]. Accurate pose estimation and depth prediction are essential for classifying semantic labels and 3D reconstructions, as inaccuracies in either can propagate errors throughout the voxel integration and segmentation pipelines [18, 5].

For pose, we measured the Absolute Trajectory Error (ATE), which is the average difference between the each estimated camera pose and the corresponding ground truth pose over the trajectory; a lower value indicates better global alignment. We also computed the relative pose error (RPE) in both its translational and rotatinal components to quantify frame-to frame drift where consistently low RPE values imply more accurate local alignment across sequential frames. To assess the depth map reconstruction, we reported the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) in meters, which capture the magnitude of depth deviations, and we included the Mean Relative Error (REL) to account for variations in the absolute scale. Low RMSE, MAE and REL values demonstrate that the predicted depths remain close to those acquired by the real sensor recording.

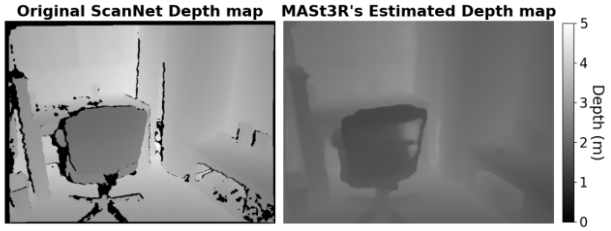

Figure 4. **Depth Map Comparison**Two depth maps for the same frame in Apartment 1: left: original ScanNet depth; right:MASt3R-estimated depth.

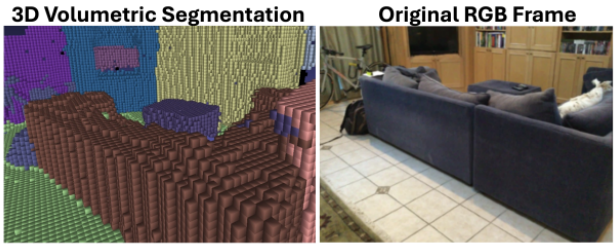

Figure 5. **Voxel Segmentation**Comparison of a voxel-based 3D segmentation (left) and original RGB image of the same scene (right). Each color in 3D volume corresponds to a distinct class from NYU40 class. Best viewed zoomed-in in electronic version..

Scene ID	Model	mIoU	F1	Accuracy	Precision	Recall	Dice	DCE Loss
Apartment 1	DeepLabV3+	0.034	0.041	0.095	0.077	0.060	0.057	11.164
	SegFormer	0.039	0.001	0.006	0.004	0.001	0.003	11.761
Apartment 2	DeepLabV3+	0.0034	0.0054	0.140	0.071	0.079	0.075	9.367
	SegFormer	0.011	0.014	0.069	0.015	0.014	0.018	9.617
Apartment 3	DeepLabV3+	0.033	0.052	0.219	0.060	0.074	0.079	10.732
	SegFormer	0.022	0.012	0.089	0.013	0.011	0.027	11.405

Table 1. 2D Segmentation Metrics Across Apartment Scenes, rounded to three decimal places

5.3. Results and Evaluation

From the 6 we can interpret that DeepLabv3 + and Segformer predicted the indoor environment without so clear segmentation compared to ScanNet Ground truth segmentation.

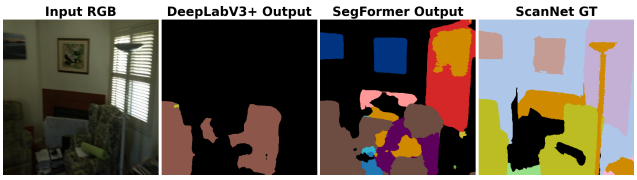

Figure 6. **Scene-based segmentation.** Visualization of our RGB Image from a Scene and respective Segmentation from the DeepLabV3+, Segformer and Scannet Ground Truth.

We were able to fine-tune the segformer as we can visualize from Figure 7. The results from the fine-tuning are very well classified in terms of the masking of the image frames. In the code i did notice that there was a bug with the validation data leak to the training image frames, but it was very last moment to fix it. From 2 we can infer that the three apartments show a similar order of magnitude for each of these metrics; at the same time, MASt3R behaves consistently in all these spaces with a minor accuracy.

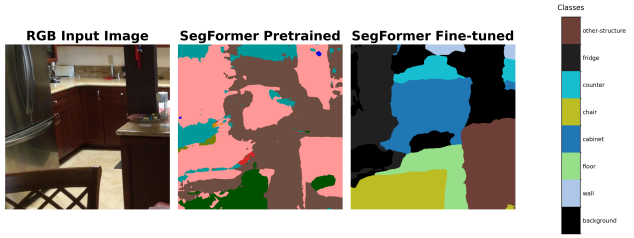

Figure 7. **Scene-based segmentation.** Visualization of our RGB Image from a Scene and respective Segmentation from the fine-tuned Segformer for Apartment3

Apartment 2 in general has poorer performance due to fewer image frames and noise from the data set capturing. Usually, this happens when the scene has a lot of elements that cause clutter, making these metrics perform poorer. Overall, the Apartment 1 has better performance according to the metrics because of the size of the data set and smoother images that provide the chance for overlap for dense fusion, reducing pose and depth errors.

Metric	Apartment 1	Apartment 2	Apartment 3
ATE [m]	3.177	2.987	2.981
RPE Trans [m]	1.442	1.673	1.585
RPE Rot [°]	23.660	24.743	23.911
Mean Tran [m]	6.724	7.912	7.314
Mean Rot [°]	122.00	127.92	123.71
RMSE [m]	0.2616	0.2993	0.2904
MAE [m]	0.1934	0.2217	0.2331
REL	0.0950	0.1193	0.1320

Table 2. **Pose and Depth Reconstruction Metrics**: Quantitative evaluation of MASt3R-predicted camera poses and depth maps across three ScanNet apartment sequences. Lower values across all metrics indicate more precise geometric reconstruction.

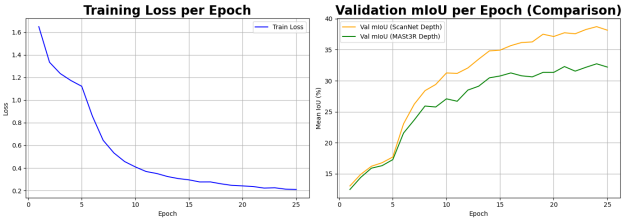

Figure 8. **Training and Validation Loss** Visualization of the Training and Validation Loss per Epoch for the Apartment3

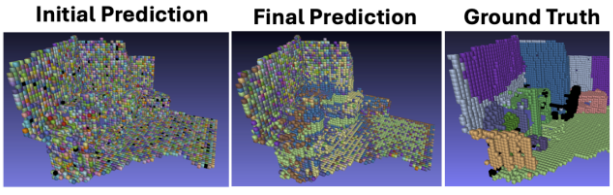

Figure 9. Results from the Final Prediction of the 3D Unet

Scene	Depth Source	Train Loss	Val Loss	Dice	mIoU	Precision	Recall	F1	Accuracy
Apartment 1	ScanNet	5.960	6.288	0.028	0.053	0.066	0.049	0.056	0.061
	MASt3R	7.045	8.094	0.025	0.044	0.043	0.055	0.048	0.042
Apartment 2	ScanNet	7.996	8.466	0.033	0.008	0.035	0.039	0.037	0.041
	MASt3R	8.312	8.682	0.030	0.009	0.037	0.029	0.033	0.039
Apartment 3	ScanNet	7.443	7.832	0.038	0.059	0.103	0.189	0.133	0.086
	MASt3R	6.720	7.762	0.037	0.047	0.081	0.180	0.112	0.091

Table 3. 3-D segmentation metrics with ScanNet vs. MASt3R depth on three apartment scenes.

5.4. Analysis and Discussion

2D Segmentation. From the results, we were able to interpret that Zero-Shot DeepLabV3 + and Segformer both struggled when deployed directly on the ScanNet frames, indicating that there was a gap between the COCO [26] and AED20K [27] pre-trained models and the indoor Scannet Scans. Both suffer from the overlap i.e the mIoU is less than 0.04, this can be intepreted as a domain mismatch between the pre-training datasets. In Particular, DeepLab V3+ often produces the large, black regions-evidence of widespread miclassifications. After the fine-tuning the Segformer-B0, We noticed that the segformer performed extremely well on the data, which i think initally worked as per this reuslts in the Figure7. From the metrics, we did notice that the metrics were extremely idealistic, which was because of a data leak from the bug in the code, but it was too late to fix it.

3D Segmentation. From the results of the 3D Segmentation Figure 3, Fusing the true sensor depth into a TSDF volume yields 20% better mIoU and lower validation loss showing the depth noise from the learned stereo directly affects the segmantic accuracy. MASt3R depth sometimes smooths the TSDF and yields marginally lower loss but the noise in the MASt3R data costs real semantic accuracy.

6. Conclusion & Future Work

We were able to learn the process of acquiring a licensed open source dataset from the owners, working on the preprocessing of the ScanNet dataset helped us understand the need for rgb, depth , pose and semantics to be able to generate a 3D mesh reconstruction. Running the inference on the pre-trained models helped us gain the intuition of how to make use of the off the shelf models with the Dataset we have by making the dictionary classes to map the class information.We started with the DeepLabv3+ and then we pivoted to the Segformer based on the results we have seen on the Metrics. The performance of the Segformer fine-tuning yields an improvement, but the results have an issue with the training and validation data mixing up, and it was too late to identify the bug after running the experiments. But we did learn the effects of the validation and training data mixup. Volumetric Segmentation using the predicted depth of MASt3R sometimes eases optimization, but the depth of the ground truth sensor consistently produces the higher semantic overlap. In Future, we would like to scale to more scenes and various indoor layouts and test the generalization, explore hybrid 2D-3D architectures with the cross-modal transformers. Integrating this to a real-time indoor humanoid or legged robot and running experiments is the ultimate goal.

7. Contributions and Acknowledgments

All authors contributed to the manuscript, discussed the results extensively, and provided feedback on the experimental design. We would like to extend a special thank you to our mentor, Kyle Sargent, for his invaluable guidance throughout this project. We had spent several hours over office hours discussing and getting feedback on our progress. We would like to thank Cristobal Eyzaguirre for helping us explore the ScanNet dataset initially, We also thank the course instructors and teaching assistants for their guidance throughout this project and helping us get the approvals for the dataset; we also thank the ScanNet team for providing us with the dataset; and additional appreciation goes to the open-source community for providing tools such as Hugging Face Transformers.We note that this project has not been submitted to a peer-reviewed conference or journal.

Karthik helped getting the approval and documentation work for Scannet Dataset Usage. He worked on the ScanNet Data Pre-Processing, converting the dataset to a suitable format to be able to use it for training/inference on the models. Karthik did the metrics calculation for Pose and Depth to compare the results between ScanNet and the MASt3R. Karthik also worked on the 2D Segmentation baseline by running experiments on the pretrained models DeepLabV3+ and Segformer, fine-tuning of the Segformer on the Scannet Data to improve the segmentation masks.

Arya helped on processing the ScanNet Dataset and getting MASt3R to work in the Google colab. Arya also augmented the dataset with pose and depth predictions from MASt3R using the ScanNet dataset. Arya also worked on setting up and training the 3D U-net in python scripts, investigating the 3D reconstruction pipeline (Open3D, MONAI).

References

[1] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018.

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[4] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760, 2012.

[5] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *ICLR*, 2020.

[6] Naver Labs. DUSt3R: Deep uncertainty-based stereo and 3D reconstruction. `https://github.com/naver/dust3r`, 2022.

[7] Ryosuke Murai, Ravi Garg, and Seokju Lee. MASt3R: Boosting 3D reconstruction accuracy with matching and stereo. `https://github.com/rmurai0610/MASt3R-SLAM`, 2023.

[8] John McCormac, Andrew Handa, Stefan Leutenegger, and Andrew J. Davison. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *ICRA*, 2017.

[9] Seiichiro Narita, Yusuke Nakashima, Takayuki Okatani, and Toru Nagano. PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. In *ICRA*, 2019.

[10] Christopher B. Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio–Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019.

[11] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[12] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. ElasticFusion: Dense SLAM without a pose graph. In *RSS*, 2015.

[13] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. In *ACM TOG (SIGGRAPH)*, 2017.

[14] Helen Oleynikova, Zachary Taylor, Marius Fehr, Juan Nieto, and Roland Siegwart. Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning. In *IROS*, 2017.

[15] Angela Dai and Matthias Nießner. 3DMV: Joint 3D–multi-view prediction for 3D semantic scene segmentation. In *ECCV*, 2018.

[16] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *CVPR*, 2021.

[17] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. Semantic-NeRF: In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.

[18] Richard A. Newcombe, Andrew Fitzgibbon, and Andrew Davison. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.

[19] MONAI Consortium. MONAI: A framework for deep learning in healthcare. `https://monai.io`, 2020.

[20] Marius Cordts et al. The Cityscapes Dataset for semantic urban scene understanding. In *CVPR*, 2016.

[21] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[22] Fabian Isensee et al. nnU-Net: Self-adapting framework for U-Net-based medical image segmentation. *Nature Methods*, 18(2):203–211, 2021.

[23] Adam Paszke et al. PyTorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[25] Liang-Chieh Chen et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[26] Tsung-Yi Lin et al. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[27] Bolei Zhou et al. Scene parsing through ADE20K dataset. In *IJCV*, 2017.