Dynamic Object Removal and Inpainting for improving Visual SLAM (Simultaneous Localization and Mapping)

Ankur Mahesh Chavan achavan1@umd.edu
Neha Nitin Madhekar nehanm97@umd.edu
Rashmi Kapu rashmik@umd.edu
Vinay Krishna Bukka vinay06@umd.edu

Abstract

Visual Simultaneous Localization and Mapping (SLAM) find applications in autonomous navigation, robotics, and virtual reality. However, existing research faces challenges, particularly in dynamic scenes where the presence of moving objects poses obstacles for long term use. This project addresses the need for enhancing Visual SLAM in dynamic environments by introducing an innovative approach to identify and eliminate dynamic objects from video frames, followed by seamless inpainting. Utilizing transformer architectures for dynamic object segmentation and employing a dual-domain propagation along with an efficient mask-guided sparse video Transformer for video inpainting, our project achieved overall improved accuracy. The proposed system holds substantial potential to advance Visual SLAM capabilities in real-world scenarios, with practical applications in robotics, autonomous vehicles, and augmented reality. The source code used for this project is available at https://github.com/Dynamic-Object-Removal-and-Inpainting

1. Introduction and Motivation

Simultaneous Localization and Mapping (SLAM) is a pivotal technology in the field of robotics, enabling robots to construct maps of their surroundings while precisely determining their positions within those maps. Visual SLAM, based on camera sensors, has garnered significant attention due to its advantages in terms of size, power efficiency, and cost-effectiveness. However, it faces unique challenges, particularly in dynamic environments, where conventional approaches assume static surroundings. This limits the application of visual SLAM in real-world scenarios with common dynamic elements, such as pedestrians and moving vehicles, which are prevalent in applications like home robotics, autonomous vehicles, and augmented reality. Identifying and managing dynamic objects is crucial for the accurate estimation of stable maps, particularly in the context of long-term applications. Failure to detect dynamic content results in its inclusion in the 3D map, posing challenges for effective tracking and relocation purposes.

Dynaslam [1] addresses the need to enhance Visual SLAM by tackling the challenge of dynamic objects. The primary objective of this project is to improve dynamic object segmentation within video frames,remove them from the scene,and subsequently inpaint the removed regions to ensure smooth video frames which then can be used for 3D map generation.

The innovative approach proposed here leverages transformer architecture, Segmenter[10] for dynamic object segmentation, aiming for improved results. The study utilizes Transfer Learning on Pascal VOC dataset [5] to finetune the segmentation module. Furthermore, the project uses state-of-the-art pre-trained architecture ProPainter[15] for improved background restoration, which is based on dual-domain propagation along with an efficient mask-guided sparse video Transformer.To evaluate the proposed system, various metrics are employed, encompassing segmentation accuracy and the quality of inpainting results. The overall result of our system after inpainting is better than baseline, DyanSLAM[1].

2. Problem Statement

This project aims to segment dynamic objects in the video, remove them from the frames, and further inpaint these frames with background as we don't want dynamic objects to be used for camera localization and to be included in the final reconstructed map of the scene.This project focuses on two major aspects, Dynamic Object Segmentation and Video Inpainting.

3. Related Work

ORB-SLAM2 [9] introduces a robust simultaneous localization and mapping (SLAM) system for various camera configurations, achieving state-of-the-art accuracy in real-time across diverse environments. It includes features such as map reuse, loop closing, and relocalization, offering a comprehensive SLAM solution. In contrast, DynaSLAM [1] extends ORB-SLAM2 by addressing the limitations of scene rigidity assumptions in traditional SLAM algorithms.

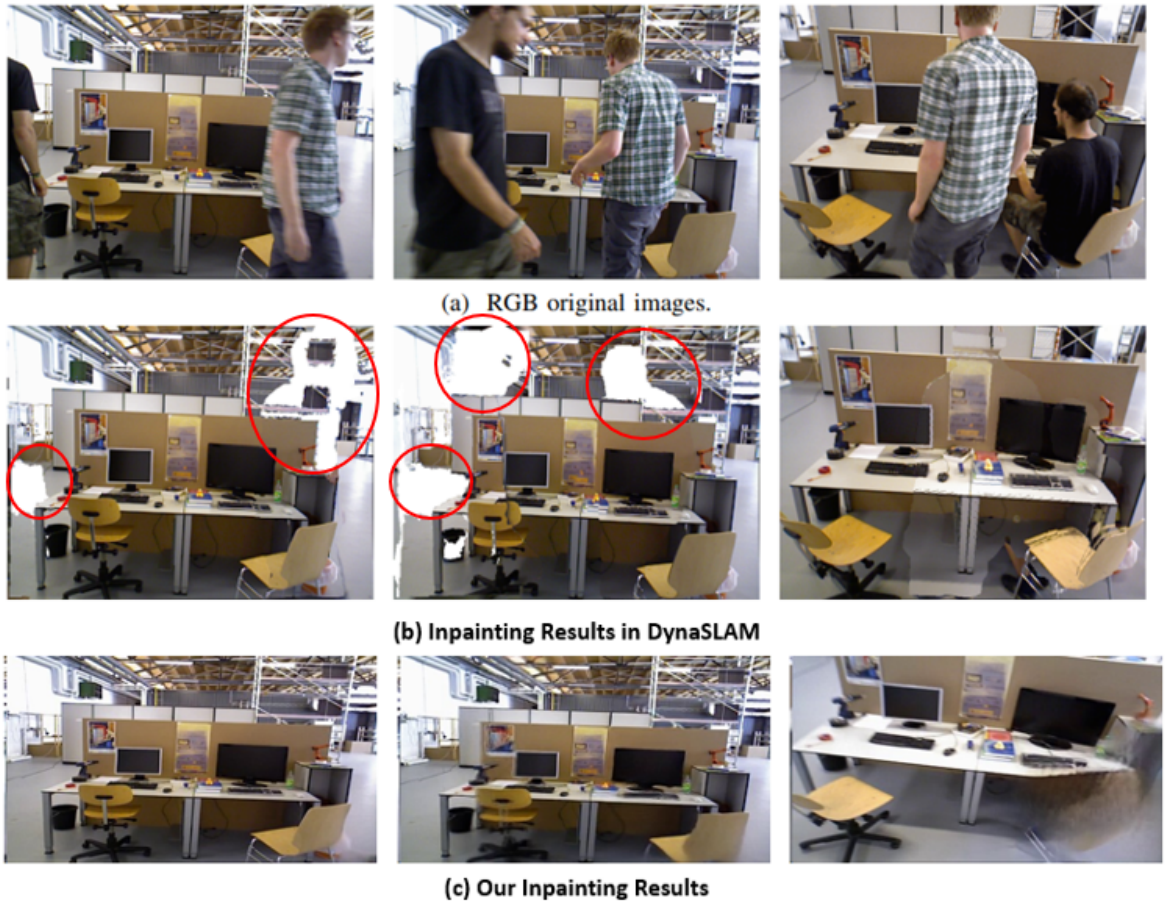

Figure 1. Comparison of Our Inpainting Results with DynaSLAM

DynaSLAM introduces dynamic object segmentation and background inpainting, enhancing its robustness in dynamic scenarios. It outperforms standard visual SLAM baselines in accuracy, providing a solution for applications in real-world environments with moving objects. Both approaches contribute significantly to advancing SLAM capabilities, catering to different aspects of dynamic scene understanding and mapping.

The transformative impact of transformers in Natural Language Processing (NLP) has extended to computer vision, influencing various tasks such as object detection, semantic segmentation, panoptic segmentation, video processing, and few-shot classification. Vision Transformers (ViT) [4] marked a significant shift by introducing a convolution-free transformer architecture for image classification, treating input images as sequences of patch tokens. While ViT requires training on large datasets, methods like Data-efficient Image Transformer (DeiT) [12] propose token-based distillation strategies to achieve competitive vision transformer performance using a CNN as a teacher. Recent extensions explore applications in video classification and semantic segmentation, with approaches like SETR [13] utilizing a ViT backbone and a standard CNN decoder, and Swin Transformer [8] employing a variant of ViT composed of local windows and UpperNet as a pyramid Fully Convolutional Network (FCN) decoder. These advancements showcase the versatility and adaptability of transformer-based architectures across diverse computer vision tasks.

In the domain of video inpainting (VI), two dominant mechanisms, flow-based propagation and spatiotemporal Transformers, have demonstrated effectiveness but are not without limitations. Previous approaches employing propagation-based methods often operate separately in either the image or feature domain, leading to spatial misalignment due to inaccurate optical flow. Additionally, memory or computational constraints in existing temporal feature propagation and video Transformers limit their exploration of correspondence information from distant frames. To overcome these challenges, ProPainter [15], introduces dual-domain propagation that seamlessly combines image and feature warping, leveraging global corre-

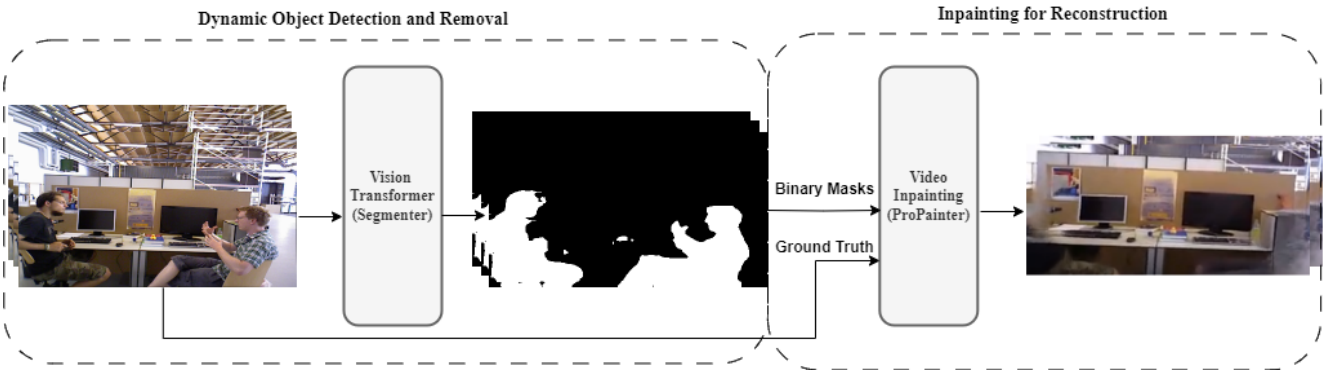

Figure 2. Pipeline Diagram of the Method Implemented

spondences more reliably. ProPainter[15] presents a mask-guided sparse video Transformer for enhanced efficiency by discarding unnecessary tokens. It surpasses prior methods by a substantial margin, demonstrating superior performance while maintaining efficiency.

DynaSLAM[1] uses Mask RCNN[7] along with multi-view geometry for dynamic object detection. For inpainting, it utilizes a naive geometric approach using the information from previous 20 frames.However, the inpainting quality using this method is not good. Some gaps have no correspondences and are left blank, some areas cannot be inpainted because their correspondent part of the scene has not appeared so far in the keyframes.To tackle this problem, this work proposes to use other methods of inpainting.

4. Methodology

The main goal of the project is to formulate an end-to-end pipeline that will perform dynamic object detection and removal from a scene and reconstruct the scene without dynamic objects. This will help to create an efficient SLAM map of an environment. The overall pipeline is depicted from Figure 2 Section 4.1 explains how dynamic objects from a scene are removed. Section 4.2 explains how efficiently a scene is reconstructed without dynamic objects.

4.1. Segmentation of Dynamic objects

The baseline for our project, DynaSLAM[1], utilizes Mask R-CNN[7] for pixel-wise semantic segmentation. However, recent advancements in neural network architectures have yielded models with superior performance compared to Mask RCNN. Notably, transformers have demonstrated enhanced capabilities in semantic segmentation when compared to traditional convolutional neural networks. Due to the aforementioned benefits, we experimented with Segmenter[10],the Transformer for Semantic Segmentation, using baseline code [11]. The Segmenter[10] utilizes a fully transformer-based encoder-decoder architecture that translates a sequence of patch embeddings into pixel-level class annotations. The model overview is illustrated in Figure 3. The sequence of patches undergoes encoding by a transformer encoder, and decoding through either a point-wise linear mapping or a mask transformer. For our application, we fine-tuned the Segmenter model on Pascal[5] dataset to segment dynamic object classes.

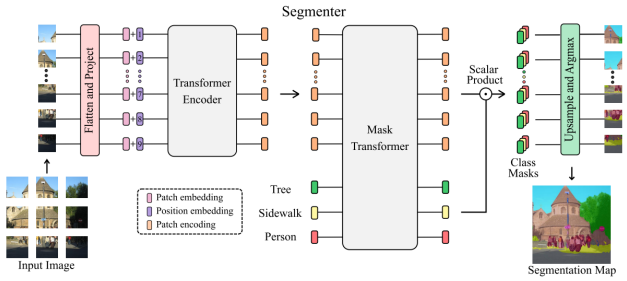

Figure 3. Segmenter Architecture [10]

4.2. Background Inpainting

Once the binary masks of the frames are generated, Propainter[15] is used to reconstruct the background containing the static structure of the environment, and for relocation and camera tracking after the map is created. ProPainter[15] was selected for inpainting after a thorough literature review of all state-of-the-art inpainting architectures.It is a dual-domain propagation efficient mask-guided sparse video Transformer. ProPainter incorporates three pivotal components as depicted in Figure 4: a highly efficient recurrent flow completion network, dual-domain propagation, and mask-guided sparse Transformer. Initially, they utilized the recurrent flow completion network to efficiently restore corrupted flow fields. Subsequently, they implement propagation in both the image and feature domains, with joint training. This strategy allows the exploration of correspondences from both global and local temporal frames, leading to more reliable and effective propagation. The subsequent mask-guided sparse Transformer

blocks refine the propagated features using spatiotemporal attention, employing a sparse strategy that selectively considers a subset of tokens.

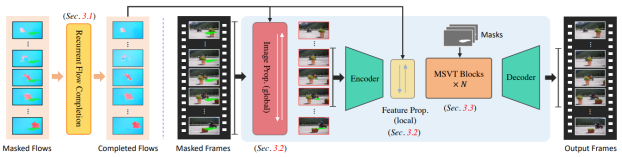

Figure 4. ProPainter Architecture [15]

Parameters	7M
Dataset	Pretrained ADE20K
Transfer Learning	Pascal VOC Dataset
Backbone	ViT
Number of Epochs	256
Weight Decay	0.00354
mIOU	70-74.91
Training Loss	0.30065
Validation Accuracy on PASCAL VOC	74.91

Table 1. Fine Tuned Seg-T mask model on PASCAL VOC

5. Data Preprocessing

To conduct inference and facilitate comparison with the baseline DynaSLAM[1], we utilize the TUM dataset[6]. A data pipeline had to be created to make the sequences in TUM RGBD dataset to be usable by Propainter. Additionally, we standardized the frames to a fixed resolution of 432x240 to mitigate computational demands and prevent potential out-of-memory errors.

6. Dataset

Train/Test Dataset	No. of Images	No. of Classes
ADE20K	20,000	150
Cityscapes	25,000	30
Pascal VOC 2010	10,103	20

Figure 5. Description of Datasets used for Training

To accomplish the dynamic object segmentation task, we intend to utilize transformer models pre-trained on the ADE20K[14] and Cityscapes[3] dataset. The ADE20K dataset for semantic segmentation comprises over 20,000 scene-centric images with comprehensive annotations at the pixel level, encompassing objects and object parts. The dataset encompasses a total of 150 semantic categories, including scenes such as sky, road, grass, and discrete objects like person, car, and bed. Cityscapes is an extensive database designed for the semantic comprehension of urban street scenes. It delivers detailed annotations, including semantic, instance-wise, and dense pixel annotations, covering 30 classes organized into 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). The dataset encompasses approximately 5,000 finely annotated images and 20,000 coarsely annotated ones. Additionally, we employed the Pascal[5] dataset for transfer learning the segmentation model specifically for dynamic object detection. For evaluating the segmentation and inpainting framework, we utilized the TUM RGB-D [6] dataset, which includes both RGB and depth images, along with corresponding ground-truth trajectories. The Pascal dataset is known for its challenging and high-quality visuals for segmentation tasks.

7. Experiments

We performed transfer learning where we used the Seg-T-Mask/16 model pre-trained on ADE20k [14] and fine-tuned it with Pascal [5] building upon baseline segmenter code [11] using cross-entropy loss function for 256 epochs to enhance performance and create a dynamic-object-specific segmentation model. The loss curves for a range of epochs are shown in Figure 6 and Figure 7. Our experiments involved Seg-T-Mask/16 models with 7 million parameters trained on ADE20K and Seg-L-Mask/16 models with 322 million parameters trained on the Cityscapes dataset. The Table 1 explains the fine tuned model characteristics. Subsequently, we utilized the binary mask frames obtained from the transformers for inpainting purposes, employing ProPainter. Inpainting results were acquired for each transformer architecture. For testing, various scenes from the TUM dataset were chosen, where we segmented moving humans in the video, extracted them, and seamlessly inpainted the scenes. Additionally, we conducted experiments by recording a video ourselves, featuring a moving human within the scene exhibiting the real-world application of our system.

Applying concepts acquired through the coursework of ENPM809K at the University of Maryland, College Park, in areas such as data preprocessing, loss functions, transfer learning, hyper-parameter tuning, evaluation metrics, and transformers, the project successfully explored and implemented ideas, leading to improved results.

8. Evaluation

8.1. Input & Output

The input to the model is a video of a dynamic scene. The final result of the proposed pipeline is a video with extracted

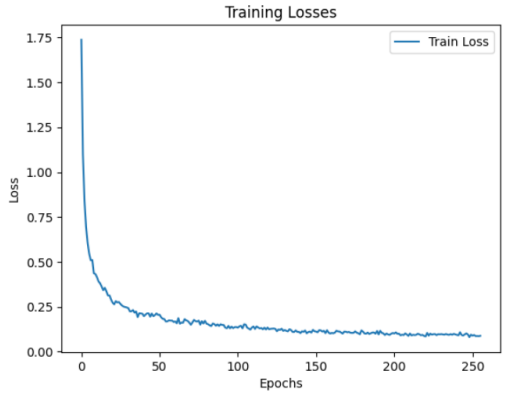

Figure 6. Training Loss

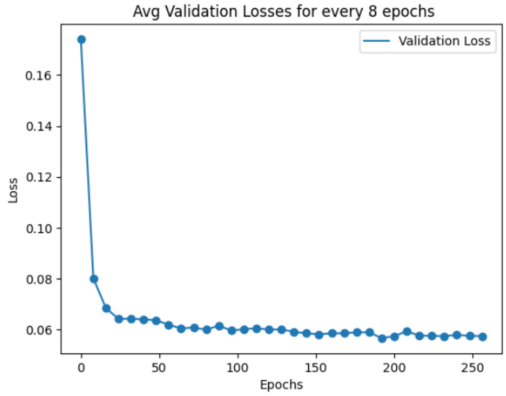

Figure 7. Validation Loss

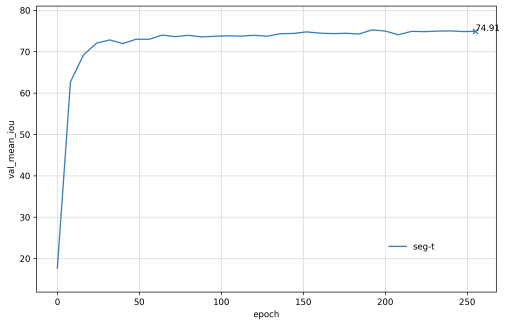

Figure 8. IoU

dynamic objects and a background inpainted. Our pipeline consists of two subsystems with intermediary results. For the Segmentation part, the result will be the masked images with pixel wise class IDs. The class IDs belonging to the dynamic classes are listed and binary mask frames are obtained for only those classes during inference. For the background inpainting part, the result will be the video frames with removed dynamic objects and an inpainted background. Using our custom data preprocessing pipeline the input to the system are frames of 432x240 resolution and the resulting final video is of the same resolution.

8.2. Evaluation Metrics

The trained models are tested and compared both qualitatively and quantitatively with different metrics.

1. **IOU:** IoU is used for evaluating the segmentation architecture it is a metric commonly used to evaluate the performance of segmentation models. Higher IoU values signify better accuracy in tasks such as object detection and segmentation. The formula for IoU is:

$$IOU = \frac{AreaofOverlap}{AreaofUnion}$$

2. **PSNR:** Peak Signal-to-Noise Ratio (PSNR) is a metric used to evaluate the quality of image or video reconstruction, including video inpainting. It measures the ratio of the maximum possible power of a signal to the power of corrupting noise, expressed in decibels (dB). For video inpainting, PSNR is calculated by comparing the original video frames with the inpainted frames. Higher PSNR values indicate better reconstruction quality, as they suggest a smaller difference between the original and inpainted frames.
   The formula for PSNR is given by:

$$PSNR = 10 \cdot \log_{10}(\frac{MAX^2}{MSE})$$

3. **SSIM:** Structural Similarity Index (SSIM) is a metric commonly used to assess the similarity between two images, including video frames in the context of video inpainting. SSIM evaluates the structural information and luminance of images to provide a perceptual similarity measure. Higher SSIM values suggest better perceptual quality

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

9. Results

9.1. Segmentation

1. Seg-T-Mask/16: The segmentation results for the pretrained model are presented in Figure 9. Evidently, the outcomes were unsatisfactory, with instances such as a chair being misclassified as a person, and a portion of a person on the left side missing from the segmentation.
2. Fine-tuned Seg-T-Mask/16:The segmentation results following transfer learning on the Pascal dataset, specifically tailored for dynamic obstacle segmentation, are illustrated in Figure 10. Post transfer learning, the model

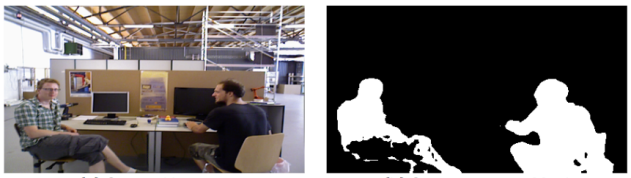

Figure 9. Segmentation results before transfer learning

demonstrates enhanced accuracy 8 in segmenting images. The corresponding evaluation metric obtained after transfer learning is depicted in the figure 1. Since the Pascal dataset has much less number of classes, the train image representation for each class is higher and hence this would be a good dataset to fine-tune on, for this project's usecase.

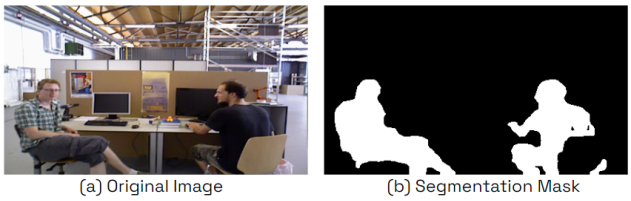

Figure 10. Segmentation results after transfer learning

3. Segmentation on Custom Data: The segmentation results on real-world recorded video are depicted in Figure 11. The discernible difference in outcomes before and after fine-tuning is readily apparent.

Figure 11. Segmentation results on Real-world recorded video

9.2. Video Inpainting:

Qualitative comparison with baseline DynaSLAM[1] is performed which can be seen in Figure 1. Our proposed system achieves better results than the baseline. Inpainting results are also obtained for a real-world recorded video, where we achieved PSNR of 21.75 and SSIM of 89.99.

10. Analyzing the Results

After qualitative analysis of poor segmentation results 9 11 obtained from the basic model with 7 million parameters it can be seen 9 that the model segments non-dynamic objects like a chair as well which is not desirable. To tackle this issue we performed transfer learning to segment only dynamic classes like person. This also resulted in the overall improvement of segmentation accuracy. Binary frames obtained after transfer learning resulted in much better inpainting results. Drastic improvements in inpainting results are observed after transfer learning.

11. Limitations

The limitations of our approach stem from the need to individually train the model for each dynamic class, leading to a time-consuming and expensive process due to the creation of an exhaustive dataset for dynamic objects. This requirement for class-specific training impedes the scalability and flexibility of the model. Furthermore, the inference time of transformer architectures, which are crucial components of our system, tends to be longer, restricting their practical use to scenarios requiring 3D maps for long-term applications such as Virtual Reality and autonomous navigation in familiar environments. The real-time application, specifically for exploration robots that demand live 3D map generation, remains challenging due to the current approaches' inability to achieve real-time processing. These limitations highlight areas for future improvement in terms of model efficiency, scalability, and real-time performance.

12. Challenges During Implementation

1. **Baseline Code Issues:** To improve the SLAM and build accurate maps, the proposed segmentation and inpainting architectures were planned to be integrated with baseline DynaSLAM [2]. But, due to a large number of dependency issues with the versions of code present, the baseline code could not be executed in the short span of the project.
2. **Choosing models to fine-tune:** Several models employing ViT and DeiT backbones were investigated, with notable observations favoring the performance of Seg-L-Mask/16 trained on Cityscapes over Seg-T-Mask/16 trained on ADE20K. This disparity in performance is attributed to the abundance of classes in ADE20K leading to fewer training image representations of classes, in contrast to Cityscapes with a more limited class set. A significant contributing factor lies in the notable difference in the number of parameters between Seg-L-Mask/16 (322M) and the comparatively less parame-

Figure 12. Inpainting Result on Custom Data (PSNR = 21.75, SSIM=89.99)

terized Seg-T-Mask/16 (7M). Despite these challenges, the decision to proceed with transfer learning on Seg-T-Mask/16 was motivated by constraints in GPU capacity and time, resulting in commendable outcomes. Future endeavors aim to enhance the capabilities of Seg-L-Mask/16 through further exploration and incorporation of DeiT, demonstrating a commitment to advancing the model's performance.

13. Innovation

This project represents a notable innovation within the realm of Visual SLAM systems, focusing on the critical aspects of dynamic obstacle extraction and video inpainting. The innovation lies in the deliberate architectural choices made during the experimentation phase. In contrast to prior approaches heavily reliant on CNN-based architectures for dynamic object extraction, our project takes a pioneering step by exploring the application of visual transformer-based architectures. This avenue, relatively unexplored in this context, seeks to address the shortcomings of existing methods, where segmentation masks often lacked the desired quality for effective inpainting. This innovation not only contributes to the refinement of Visual SLAM systems but also sheds light on the transformative potential of visual transformers in these applications.

14. Conclusion

In conclusion, our project successfully addressed the limitations of the baseline DynaSLAM[1], which employed Mask RCNN for dynamic obstacle segmentation and a naive geometric approach for inpainting. By exploring the application of transformer architectures, particularly for dynamic obstacle segmentation and video inpainting, we achieved substantial improvements. Transfer learning played a pivotal role, enhancing the dynamic obstacle segmentation results and, consequently, the quality of inpainted videos. This project not only fills the gap in exploring transformers within the Visual SLAM system but also demonstrates superior performance compared to our baseline. Real-world applicability was showcased through successful experiments on recorded videos. The resulting high-quality 3D maps that can be generated by using the inpainted videos devoid of dynamic obstacles obtained from our system have valuable applications in Virtual Reality and autonomous navigation, where precise scene replication is crucial. Furthermore, researchers focusing on localization, tracking, trajectory estimation, and 3D scene regeneration within Visual SLAM can leverage our pipeline, benefiting from its ability to deliver high-quality dynamic obstacle extraction and video inpainting results.

15. Future Improvements

For future improvements, the development of context-based learning models presents an exciting avenue. These models could autonomously discern between moving and movable objects, potentially mitigating the need for extensive and specific model training for each dynamic class. Additionally, strategies to minimize inference time are crucial for achieving real-time operation, especially in dynamic environments where prompt responses are essential. Exploring techniques such as model optimization and hardware acceleration could contribute to reducing latency. Furthermore, a promising direction involves the creation of a unified, end-to-end trainable network. Such a network would have the capability to seamlessly integrate dynamic object segmentation and video inpainting into a single, efficient process.

16. New Applications

1. **Video Editing and Post-Production:** Filmmakers can use the pipeline to enhance the visual aesthetics of

scenes by selectively removing or modifying dynamic elements that might detract from the intended mood or atmosphere.

2. **Augmented Reality (AR):** In AR applications, where virtual objects are overlaid on the real-world environment, removing dynamic obstacles from the video feed can enhance the AR experience by providing a clearer view of the real scene.
3. **Live Broadcasting:** For live broadcasts, especially in crowded environments, removing or anonymizing people or objects in real-time can be useful for privacy or content moderation purposes.

17. Contributions

Summary of contributions is listed in table below

Literature Review	Ankur, Neha, Rashmi, Vinay
Segmentation Fine Tuning	Rashmi, Vinay
Video Inpainting	Ankur, Neha
Report Making	Ankur, Neha, Rashmi, Vinay
Presentation Preparation	Ankur
Result, Graphs Generation	Neha, Rashmi

Table 2. Team Contributions

18. Acknowledgement

We extend our sincere appreciation to Professor George Zaki for imparting in-depth knowledge of Deep Learning concepts through the ENPM809K course and for providing continued support throughout the project's lifecycle. Additionally, our gratitude extends to Davit Soselia, whose guidance was instrumental in overcoming implementation challenges associated with our baseline, and for introducing alternative approaches that proved invaluable in successfully achieving our project milestones.

References

[1] Berta Bescos, Jose M. Facil, Javier Civera, and Jose Neira. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 1, 3, 4, 6, 7

[2] Fácil JM. Civera Javier Bescos, Berta and José Neira. DynaSLAM: Tracking, mapping and inpainting in dynamic environments. https://github.com/BertaBescos/DynaSLAM, 2018. 6

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. 4

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, 2010. 1, 3, 4

[6] Caner Hazirbas, Andreas Wiedemann, Robert Maier, Laura Leal-Taixé, and Daniel Cremers. Tum rgb-d scribble-based segmentation benchmark. https://github.com/tum-vision/rgbd_scribble_benchmark, 2018. 4

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 3

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 2

[9] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[10] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021. 1, 3

[11] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. https://github.com/rstrudel/segmenter, 2021. 3, 4

[12] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2

[13] Li Zhang, Jiachen Lu, Sixia Zheng, Xinxuan Zhao, Xiatian Zhu, Yanwei Fu, Xiang Tao, and Jianfeng Feng. Vision transformers: From semantic segmentation to dense prediction. *arXiv*, 2023. 2

[14] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 4

[15] Shangchen Zhou, Chongyi Li, Kelvin C. K. Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting, 2023. 1, 2, 3, 4