LIDNeRF: Language guided NeRF Editing with InstructDiffusion

Project report submitted in the partial fulfillment

of

BACHELOR OF TECHNOLOGY

in

Artificial Intelligence

by

Khushal Sharma (I083)
Manan Shah (I075)
Aniruddh Kulkarni (I081)

Under the Supervision of

Dr. Vaishali Kulkarni

HOD, Dept. of Artificial Intelligence, MPSTME

MUKESH PATEL SCHOOL OF TECHNOLOGY, MANAGEMENT & ENGINEERING
SVKM's NARSEE MONJEE INSTITUTE OF MANAGEMENT STUDIES
(Declared as Deemed-to-be University Under Sec. 3 of UGC Act, 1956)
V. L. Mehta Road, Vile Parle (West)
MUMBAI - 400056

2023-2024

CERTIFICATE

This is to certify that the project entitled “LIDNeRF: Language guided NeRF Editing with InstructDiffusion”, has been done by Mr. Manan Shah, Mr. Khushal Sharma and Mr. Aniruddh Kulkarni under my guidance and supervision & has been submitted in partial fulfillment of the degree of Bachelor of Technology in Artificial Intelligence of MPSTME, SVKM’s NMIMS (Deemed-to-be University), Mumbai, India

Internal Guide

External Examiner

Date:

Place: Mumbai

HoD
(Dept. of Artificial Intelligence)

ABSTRACT

Our research introduces an innovative approach to instruction-based image editing by leveraging the capabilities of InstructDiffusion and Lang-SAM models. By combining these models, we enable precise and context-aware edits to real-world images based on natural language instructions. The core methodology involves an iterative dataset update process where images are rendered from NeRF scenes, updated using diffusion models, and used to supervise scene reconstruction. This iterative approach allows for targeted and localized edits, enabling tasks such as object addition, removal, and replacement while optimizing the underlying 3D scene. We demonstrate the effectiveness of our method through compelling qualitative results, showcasing its versatility and ability to achieve diverse and complex image edits compared to prior work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to NeRF

1.1.1 What is NeRF?

Neural Radiance Fields (NeRF) represent a significant advancement in computer graphics and computer vision, offering a powerful methodology for modeling complex 3D scenes. Introduced in the paper titled "NeRF: Representing Scenes as Neural Radiance Fields" by Ben Mildenhall et al.[13], NeRF synthesizes high-quality 3D scenes from a sparse set of 2D images using a multi-layer perceptron (MLP) network. This technique involves transforming image pixels into the world coordinate system, generating light rays from the camera position directed toward the scene, and sampling points along these rays. NeRF utilizes volume rendering techniques to determine the color of each ray by integrating over the sampled points. The neural network predicts the volume density and RGB color values of the sampled points after positional encoding. Notably, NeRF represents scenes as continuous functions rather than using traditional geometric primitives, capturing both appearance and geometry through radiance outputs from the neural network.

During training, the network learns to predict color and density values that match observed 2D images given corresponding camera poses. A distinguishing feature of NeRF is its ability to generate novel views of scenes from any viewpoint without explicit 3D geometry reconstruction. By evaluating the neural network along rays cast from the camera through the scene, NeRF facilitates the synthesis of images from new perspectives. This technique has demonstrated remarkable efficacy in producing high-fidelity and realistic 3D scene reconstructions, holding significant promise for applications in computer graphics, virtual reality, and augmented reality.

1.1.2 What is NeRF Editing?

NeRF (Neural Radiance Fields) editing [14] refers to the process of manipulating and modifying scenes represented by NeRF models, which are neural networks capable of synthesizing high-quality 3D scenes from 2D images. NeRF editing allows for precise and detailed modifications to be made to these scenes, including changes to object positions, shapes, colors, lighting, and other scene attributes. This editing process typically involves leveraging additional neural network architectures or algorithms to interactively and intuitively manipulate the NeRF representation based on user input, such as textual descriptions or graphical interfaces.

1.1.3 Significance and Need

1. **Flexibility:** NeRF editing offers a versatile framework for precise modifications, crucial in domains like virtual production, gaming, and virtual reality.

2. **Creative Control:** It empowers users with direct manipulation over scene composition and visual effects, fostering experimentation and rapid iteration.
3. **Streamlined Content Creation:** Integrating editing capabilities directly into the NeRF representation simplifies workflow, eliminating the need for separate tools.
4. **Realism and Immersion:** NeRF editing enhances realism and immersion in virtual environments, improving user engagement across various applications.

1.2 Motivation

1.2.1 Feasibility:

- The feasibility of this project lies in the availability of existing tools and models such as NerfStudio and nerfacto models, which provide a foundation for training NeRF scenes and extracting masks
- The utilization of delta blocks and diffusion models for scene editing has been demonstrated in prior research, indicating the feasibility of implementing these techniques in the project.
- The iterative nature of the extraction and editing process allows for refinement and improvement over multiple iterations, enhancing the feasibility of achieving the project objectives.

1.2.2 Need:

- There is a growing demand for advanced scene editing capabilities in various industries including entertainment, architecture, and virtual reality.
- Existing NeRF-based editing techniques often lack precision and efficiency in modifying specific regions within a scene while maintaining overall coherence and consistency.
- By addressing these limitations, this project fulfills the need for more precise and efficient scene editing tools, empowering artists, designers, and content creators with enhanced capabilities.

1.2.3 Significance:

- The significance of this project lies in its potential to advance the state-of-the-art in NeRF-based scene editing, enabling more intuitive and precise modifications to 3D scenes.
- By integrating delta blocks and diffusion models, the project enhances the selective editing capabilities of NeRFs, facilitating targeted modifications while preserving overall scene consistency.
- The iterative approach ensures that the edited 3D scenes remain coherent and consistent, enhancing their realism and applicability in various domains including virtual production, computer graphics, and immersive experiences.

1.3 Scope of Project:

The scope of the NeRF-Editing project encompasses a comprehensive exploration of techniques and methodologies for enabling efficient and precise editing of NeRF scenes using textual guidance. This project aims to address the limitations of existing NeRF-based scene editing approaches by proposing a novel pipeline that leverages advancements in deep learning and natural language processing. The scope includes developing and implementing algorithms for scene representation, mask extraction, and scene editing, integrating state-of-the-art models such as the nerfacto model, delta block, and diffusion model. The project also involves conducting thorough evaluations to assess the accuracy, consistency, and usability of the proposed methodology. Furthermore, the scope extends to identifying challenges in the implementation process and exploring potential solutions to improve the efficiency and effectiveness of NeRF scene editing. Overall, the NeRF-Editing project seeks to advance the state-of-the-art in 3D scene editing and contribute to the broader fields of computer graphics, artificial intelligence, and human-computer interaction.

Chapter 2

Literature Survey

2.1 Brief Explanation on all recent works

Neural Radiance Fields (NeRFs), introduced by Mildenhall et al. [1] in 2020, leverage deep neural networks to represent scenes as continuous volumetric functions, enabling photo-realistic view synthesis with intricate geometry and appearance details. This technique involves tracing camera rays through scenes, capturing radiance and density information at sampled points along these rays, and predicting color and volume density values essential for rendering scenes from different viewpoints.

NeRF models Following the initial NeRF framework, subsequent advancements have addressed critical challenges such as aliasing, rendering quality, and computational efficiency. For instance, Mip-NeRF [2] integrates conical frustums to model scenes at multiple scales, reducing aliasing while exhibiting non-smooth mappings with the iNGP backend. Zip-NeRF [3] enhances rendering quality by integrating iNGP's grid pyramid into Mip-NeRF's framework, combining scale-aware anti-aliasing and fast grid-based training.

NVIDIA InstantNGP [4] utilizes multiresolution hash encoding to efficiently store trainable feature vectors in spatial hash tables, enhancing inference speed through a coarse-to-fine strategy. PyNeRF [5] introduces a hierarchical grid-based NeRF model optimizing rendering by training NeRFs at different voxel resolutions. By leveraging a voxel hierarchy and interpolating between levels, PyNeRF achieves efficient rendering with improved multiscale sampling, enabling enhanced representation of large-scale scenes.

For real-world scene captures, the nerfacto model [6] by NeRFStudio integrates camera pose refinement, proposal sampling, and hash encoding techniques to optimize NeRF-based scene reconstruction. It addresses camera pose errors, efficient sample distribution, and focused sampling guided by density functions, resulting in enhanced reconstruction quality.

Texts and NeRFs Text-based methods like Text2NeRF [7] and CLIP-NeRF [8] utilize pre-trained text-to-image models to guide NeRF scene reconstruction based on text prompts. These approaches generate high-quality, multi-view consistent results solely from natural language descriptions.

Additional methods such as DreamFusion [9], NeRF-Art [10], and Blended-NeRF [11] focus on stylization and localized editing of NeRF scenes driven by text prompts. These techniques introduce innovative strategies for altering appearance and geometry based on textual input, demonstrating flexibility and effectiveness for various 3D editing tasks.

Recent advancements in text-driven editing of Neural Radiance Fields (NeRFs) have expanded the capabilities of scene modeling and editing. InstructNeRF2NeRF [15] the current SOTA employs an image-conditioned diffusion model (InstructPix2Pix) for text-based NeRF scene editing. By iteratively optimizing input images while preserving scene structure, this method achieves refined 3D scene modifications aligned with specific edit instructions, demonstrating enhanced effectiveness in editing large-scale real-world scenes.

Blending-NeRF [12] introduces a novel approach for localized 3D object editing driven by text prompts. This method utilizes two NeRF networks—a pretrained NeRF and an editable NeRF—along with new blending operations guided by text descriptions. Leveraging the CLIP model, Blending-NeRF enables natural and localized modifications to object features based on text input, showcasing versatile 3D editing capabilities.

LatentEditor [13] presents a framework for precise and locally controlled editing of neural fields using text prompts. By embedding scenes into latent space with denoising diffusion models, LatentEditor offers a more adaptable NeRF backbone for editing. This method calculates a 2D mask in latent space to guide local modifications, achieving faster editing speeds and superior output quality. Free-Editor [14] introduces a training-free technique for 3D scene editing that overcomes multi-view inconsistency issues. By leveraging a ”single-view editing” approach and an Edit Transformer for intra-view consistency and inter-view style transfer, Free-Editor achieves significant speed improvements without compromising performance.

ReplaceAnything3D (RAM3D) [16] enables specific object replacement within scenes using text prompts and multi-view images, showcasing versatility and integration without compromising scene integrity. InseRF [17] addresses the challenge of generative object insertion into NeRF reconstructions by leveraging text-to-image diffusion model priors and single-view object reconstruction techniques. This method successfully maintains 3D consistency while inserting new objects into scenes based on textual descriptions and 2D bounding boxes. TeSTNeRF [18] focuses on text-driven 3D style transfer, using advanced text encoders and style supervision to control 3D style transfer and generate consistent views according to text prompts.

Each of these advancements contributes significantly to the progress of NeRF-based scene synthesis techniques, addressing various challenges and expanding the capabilities of NeRFs for diverse applications in scene reconstruction, editing, and stylization driven by natural language instructions. Table 2.2 Summarizes the findings, advantages and shortcomings of some of the major publications that we referred to.

2D Image diffusion models In our literature survey, we conducted a thorough review of current state-of-the-art (SOTA) models in the domain of 2D image editing, particularly focusing on textual diffusion-based approaches that could complement our methodology effectively. Due to the foundational structure of our approach, which incorporates a 2D image editing model to enhance the NeRF scene, this exploration was integral to our research.

During this survey, we specifically examined several text-based diffusion models for 2D image editing. Our goal was to identify a model capable of addressing certain limitations associated with the InstructPix2Pix model used in InstructNeRF2NeRF for dataset image editing. We aimed to evaluate recent advancements in this field and select a highly capable model with accessible checkpoints.

The InstructPix2Pix model utilizes pre-trained language (GPT-3) and text-to-image (Stable Diffusion) models to generate a dataset tailored for image editing guided by human instructions. This conditional diffusion model enables rapid and effective image edits based on user-written instructions, showcasing compelling results across a diverse range of input images and instructions.

In a similar vein, the InstructDiffusion framework integrates human instructions into computer vision tasks, offering a unified approach for applications such as image editing and segmentation. Leveraging the diffusion process, this framework predicts pixel-level changes based on detailed user prompts, enhancing the interpretability of computer vision tasks driven by natural language input.

SmartEdit introduces an innovative instruction-based image editing approach that leverages Multi-modal Large Language Models (MLLMs) and a Bidirectional Interaction Module to enhance reasoning capabilities. Excelling in complex instruction-based editing tasks, SmartEdit surpasses previous methods on the Reason-Edit dataset through comprehensive bidirectional information interactions between input images and model outputs.

Imagic presents a novel approach to text-conditioned image editing, enabling complex semantic edits on single real images using a pre-trained text-to-image diffusion model. This method facilitates diverse and high-quality edits within a unified framework without the need for multiple input images or synthetic data, demonstrating superior versatility and effectiveness in comparison to previous methods.

Furthermore, the InstructCV model leverages text-to-image generation to execute various computer vision tasks based on natural language instructions. By training on a diverse dataset covering tasks like segmentation, object detection, depth estimation, and classification, InstructCV exhibits competitive performance and strong generalization capabilities to new data and instructions, highlighting its effectiveness in unified language-guided vision tasks within the research landscape.

2.2 NeRF Papers

2.2.1 Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions[3]

The paper introduces a method for intuitive and accessible 3D scene editing using natural text instructions. The key findings include the ability to perform a wide range of edits on people, objects, and large-scale scenes while maintaining 3D consistency. The method is shown to be effective at instruction-driven, contextual, large-scale edits such as editing textures, replacing objects, and changing global properties of a scene. However, the paper also discusses limitations, such as challenges in adding entirely new objects to the scene and removing objects without replacing them with similarly salient content. The method's benefits lie in democratizing 3D scene editing for everyday users and enabling 3D-consistent edits using natural language instructions.

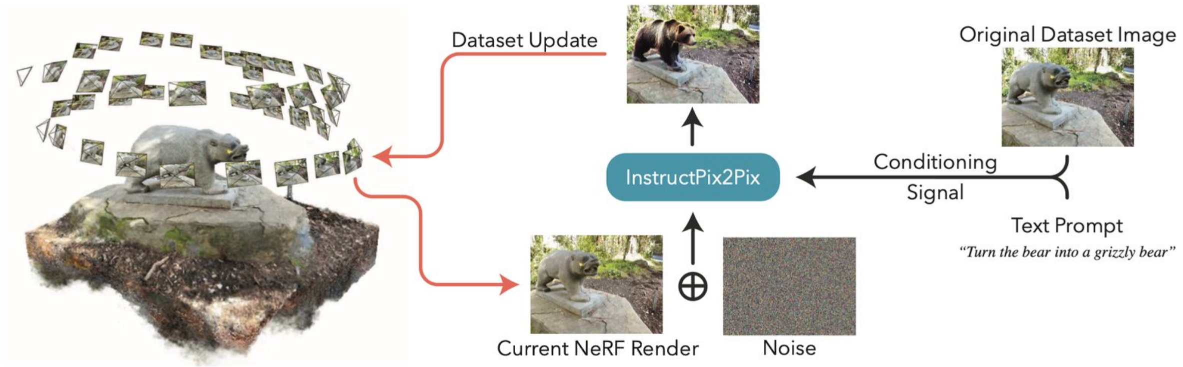


Figure 2.1: IN2N - Approach Proposed

The method uses text-based instructions to edit 3D scenes by iteratively updating the dataset images while training the NeRF. It operates on a pre-captured 3D scene and ensures that the resulting edits are reflected in a 3D-consistent manner. The process involves the following steps:

1. **Image Rendering:** An image is rendered from the 3D scene at a training viewpoint.

2. **Text-Guided Editing:** The rendered image is edited by a diffusion model called InstructPix2Pix, which is conditioned on a global text instruction.

3. **Dataset Update:** The training dataset image is replaced with the edited image.

4. **NeRF Training:** The NeRF continues training as usual, incorporating the edited images.

This iterative approach allows for the incorporation of text-based instructions into the 3D scene editing process, making it accessible and intuitive for users.

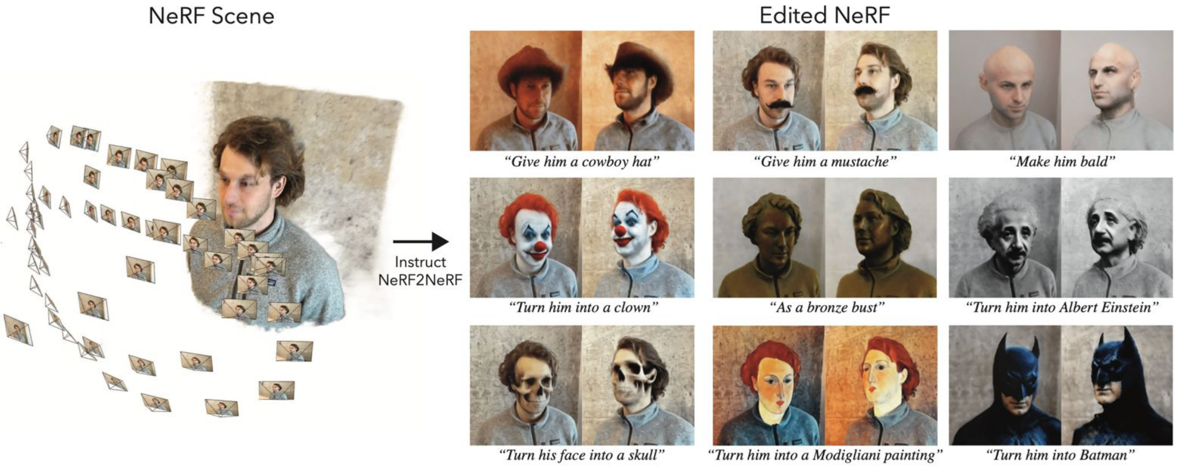


Figure 2.2: IN2N - Results

2.2.2 Free-Editor: Zero-shot Text-driven 3D Scene Editing[6]

The paper proposes a zero-shot text-driven 3D scene editing technique that allows for editing 3D scenes without the need for re-training the model during test time. It addresses the challenge of multi-view style inconsistency in state-of-the-art methods by enabling editing of a particular 3D scene by modifying only a single view. This reduces the overall editing time significantly and avoids the need for scene-specific re-training or adaptation across various editing styles. The approach leverages a generalized NeRF model and an Edit Transformer to enforce intra-view consistency and inter-view style transfer using self- and cross-attention, respectively. By utilizing pixel-aligned features and an Edit Transformer, style information is effectively transferred from the starting view to the target view, allowing for zero-shot 3D scene editing. The method also incorporates a view-filtering system to reduce 3D inconsistency among the edited images, and it relies on 2D image pre-editing for editing a single view. Overall, FREE-EDITOR offers a novel training-free 3D scene editing technique with the ability to generate novel views based on a text description while maintaining high 3D consistency.

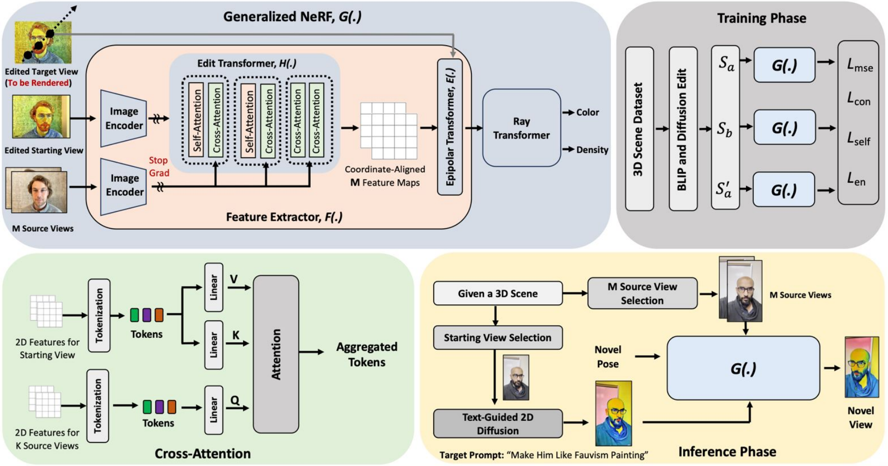


Figure 2.3: FREE-EDITOR - Approach Proposed

The approach addresses the following problems:

1. Avoiding the need for re-training the model during test time for editing 3D scenes.
2. Reducing training costs and editing time significantly.
3. Overcoming the challenge of training a diffusion model specifically for 3D scene editing due to the lack of large-scale datasets.

The limitations of the approach utilized in the paper include the potential inconsistency in multi-view edited images from the same scene, especially when rendering is performed without re-training. This inconsistency can be particularly problematic and may require trial and error to achieve the desired editing effects. The process may involve hundreds of iterations before reaching a reasonably good performance.

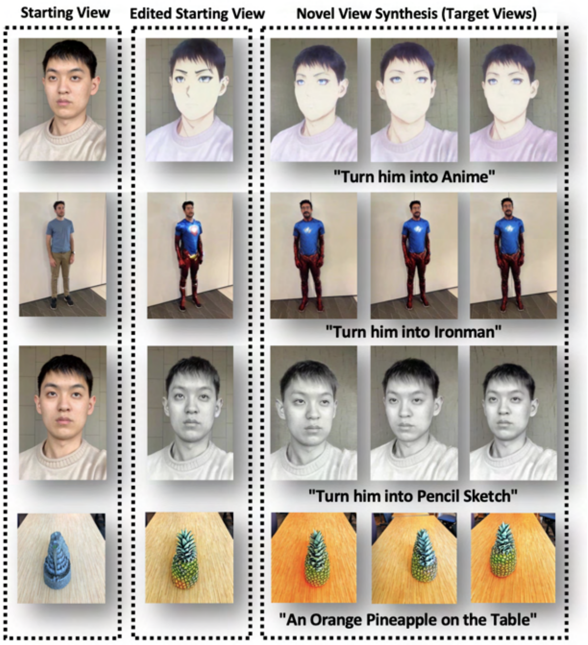

Starting View
Edited Starting View
Novel View Synthesis (Target Views)
"Turn him into Anime"
"Turn him into Ironman"
"Turn him into Pencil Sketch"
"An Orange Pineapple on the Table"

Figure 2.4: FREE-EDITOR - Results

2.2.3 LatentEditor: Text Driven Local Editing of 3D Scenes[8]

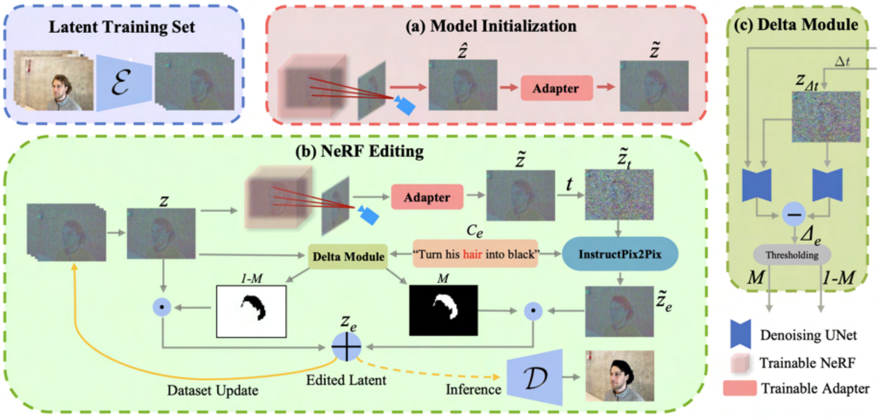


Figure 2.5: LatentEditor - Approach Proposed

The authors framework proposes a method aimed at enabling localized editing of Neural Radiance Fields (NeRF), with a focus on enhancing precision in scene alterations. It introduces a delta module to assign delta scores within the latent space, allowing NeRF training on real-world 3D scenes within this space. By leveraging latent masks, the framework facilitates more accurate editing. Additionally, it includes a refining module featuring a residual adapter and self-attention mechanisms to maintain consistency between rendered latent features and the scene's original latent representations. This approach involves training NeRF directly within the latent space, addressing challenges related to the alignment of latent and image pixels and improving the effectiveness of NeRF training. The framework also generates masks for local modifications while preserving irrelevant regions using a pixel-level scoring approach. It utilizes InstructPix2Pix (IP2P) to discern the disparity between IP2P conditional and unconditional noise predictions in the latent space. The edited latents conditioned on the 2D masks are then iteratively updated in the training process, enabling precise and locally controlled editing of neural fields using text prompts.

This framework aims to address several specific problems, including:

1. Facilitating local editing of Neural Radiance Fields (NeRF) to enable precise and locally controlled editing of neural fields using text prompts.

2. Empowering users with the ability to perform precise and locally controlled editing of neural fields using text prompts.

3. Introducing a delta module that assigns delta scores within the latent space to facilitate precise editing and preserve irrelevant regions.

4. Training NeRF directly within the latent space to enable local latent editing, ensuring consistency between rendered latent features and the scene's original latent.

The limitations of this approach include challenges such as prolonged training duration and limitations to specific scene types. Additionally, the segmentation models often fail to accurately interpret the editing prompts, requiring object specification in a singular format for segmentation. This can hinder the accuracy and efficiency of the local editing process.

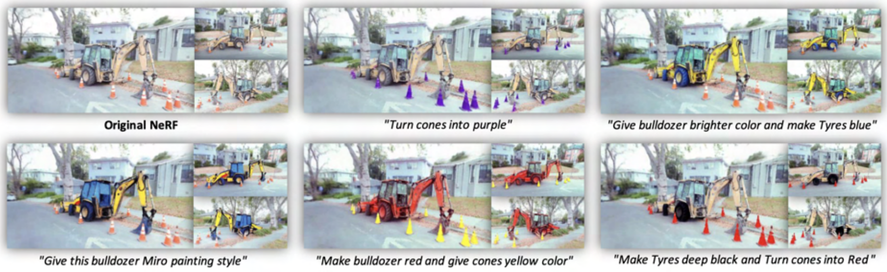

Original NeRF
"Turn cones into purple"
"Give bulldozer brighter color and make Tyres blue"
"Give this bulldozer Miro painting style"
"Make bulldozer red and give cones yellow color"
"Make Tyres deep black and Turn cones into Red"

Figure 2.6: LatentEditor - Results

2.2.4 Blending-NeRF: Text-Driven Localized Editing in Neural Radiance Fields[7]

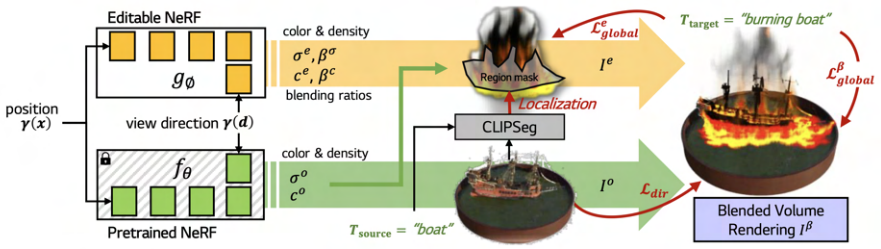


Figure 2.7: Blending-NeRF - Approach Proposed

The proposed method for editing NeRF is based on a novel architecture called Blending-NeRF. This architecture consists of two components: a pretrained NeRF and an editable NeRF. The pretrained NeRF provides the original 3D object representation, while the editable NeRF is trained to render a blended image of the two NeRFs, allowing for localized editing of the original object. The target region for editing is specified by the source text and the original object in the pretrained NeRF.

The method involves parameterizing specific regions in the implicit 3D volumetric representations and blending the original 3D object representation with the editable NeRF architecture specifically trained to render the blended image naturally. This allows for various editing operations, including color changes, density addition, and density removal. By blending density and color from the two NeRFs, fine-grained localized editing of 3D objects can be achieved.

The proposed architecture, enables the natural editing of specific regions of the 3D object while maintaining the overall appearance. This method allows for precise and localized editing of object shapes and colors based on textual guidance.

The proposed approach has limitations in the following areas:

1. Computational Cost: Multiple NeRF networks and blending operations may lead to a high computational cost, affecting real-time applications.

2. Training Data Dependency: Effective learning of blending operations and achieving accurate localized editing may require a substantial amount of training data, posing challenges in scenarios with limited or specific datasets.

3. Boundary Ambiguity: Ambiguous object boundaries may cause difficulties, potentially resulting in noise in edited results.

4. Editing Precision: While offering localized editing, precision may be limited, particularly with complex or intricate 3D objects.

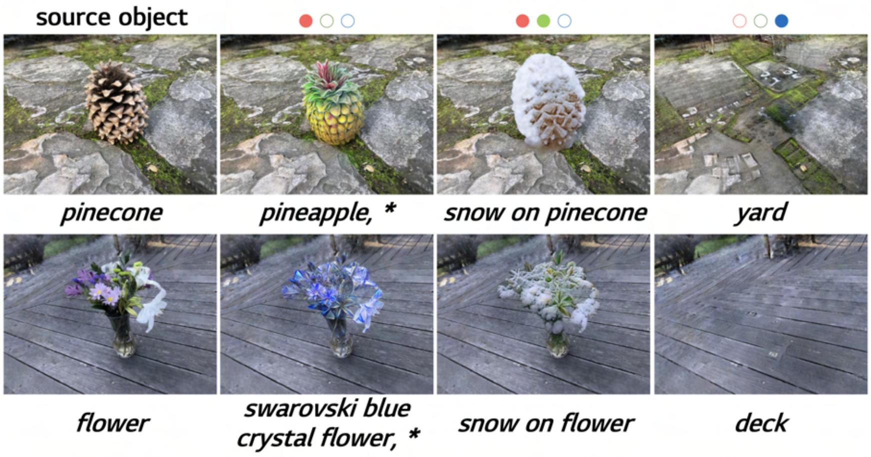
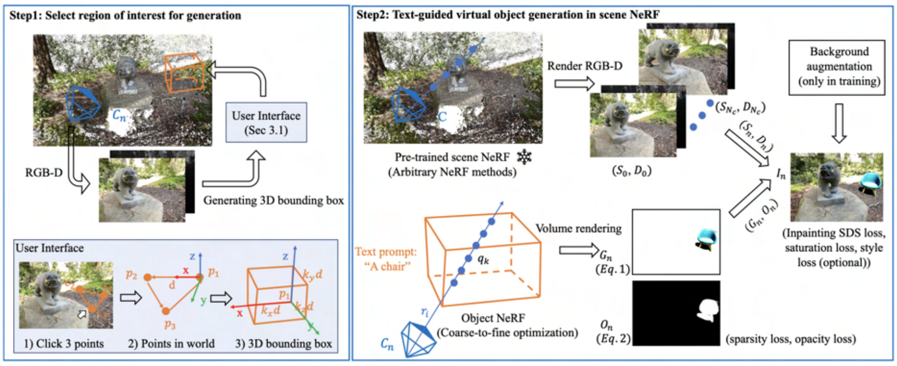


Figure 2.8: Blending-NeRF - Results

2.2.5 GO-NeRF: Generating Virtual Objects in Neural Radiance Fields[11]



Step1: Select region of interest for generation
C_n
User Interface (Sec 3.1)
RGB-D
Generating 3D bounding box
User Interface
p_2
p_1
p_3
x
y
z
d
$k_y d$
$k_x d$
$k_z d$
1) Click 3 points
2) Points in world
3) 3D bounding box
Step2: Text-guided virtual object generation in scene NeRF
C
Pre-trained scene NeRF (Arbitrary NeRF methods)
Render RGB-D
(S_0, D_0)
(S_{N_c}, D_{N_c})
(S_n, D_n)
Background augmentation (only in training)
I_n
(Inpainting SDS loss, saturation loss, style loss (optional))
Text prompt: "A chair"
q_k
r_i
C_n
Object NeRF (Coarse-to-fine optimization)
Volume rendering
G_n $(Eq.1)$
O_n $(Eq.2)$
(G_n, O_n)
(sparsity loss, opacity loss)

Figure 2.9: GO-NeRF - Approach Proposed

The proposed methodology of GO-NeRF involves utilizing scene context for high-quality and harmonious 3D object generation within an existing NeRF. The method employs a compositional rendering formulation that allows the generated 3D objects to be seamlessly composited into the scene using learned 3D-aware opacity maps without introducing unintended scene modification. Additionally, tailored optimization objectives and training strategies are developed to enhance the model's ability to exploit scene context and mitigate artifacts originating from 3D object generation within a scene.

This paper addresses the problem of generating context-compatible 3D virtual objects from text prompts and seamlessly compositing them into a pre-trained neural radiance field while preserving unchanged scene content beyond the desired editing space. It aims to enable the generation of high-quality and seamlessly integrated 3D objects within a given scene context, while also addressing issues such as over-saturation in synthesized objects, floater artifacts caused by compositing, and the need for reference-guided feature space loss for style control.

The proposed GO-NeRF methodology has a few limitations:

1. Reflection Optimization: Optimizing reflections is constrained by the 3D bounding box range, impacting reflection completeness due to uncertain reflection regions.

2. Uncertain Reflection Region: Reflections may extend beyond the 3D box range, reducing the accuracy of generated reflections.

3. Scene Context Utilization: The model's generative capacity and the utilization of scene contextual information can be further improved for better optimization and seamless composition.

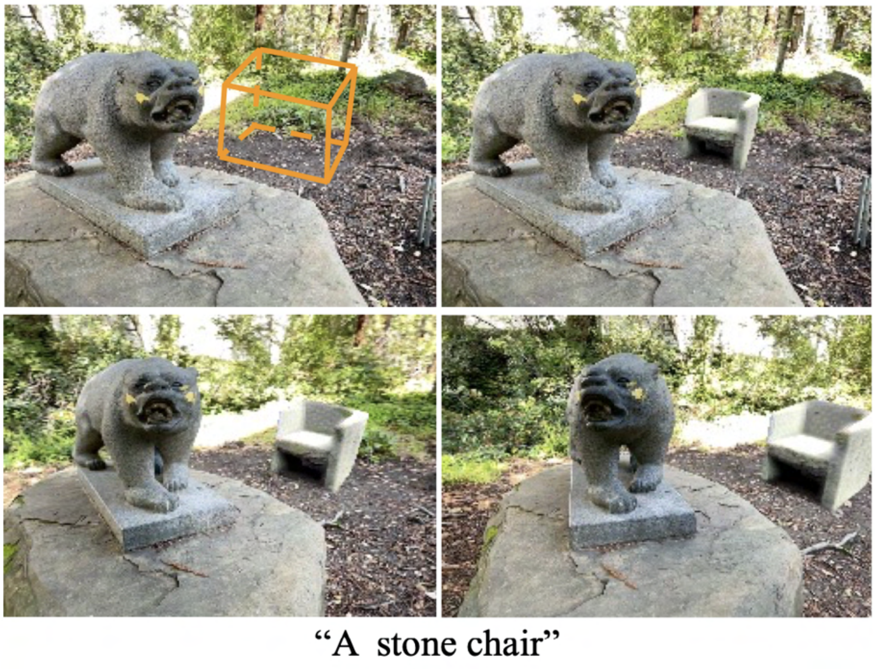


Figure 2.10: GO-NeRF - Results

Based on the literature review of current textual based NeRF editing research approaches, we have found out that most of the methodologies incorporate the InstructPix2Pix model for 2d image editing and this model itself have some limitations. Hence, we headed out to find a 2d image editing model which overcame the limitations of the InstructPix2Pix model and gave better editing results along with having better hold over language inputs. We have referred and shortlisted 2 other models which could be used instead of the current SOTA model.

2.3 2D Image editing Papers

2.3.1 InstructDiffusion: A Generalist Modeling Interface for Vision Tasks[1]

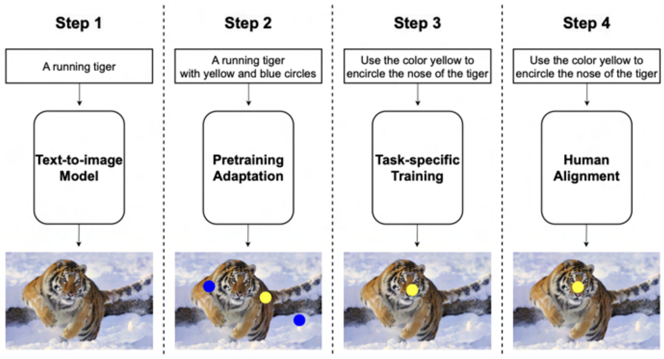
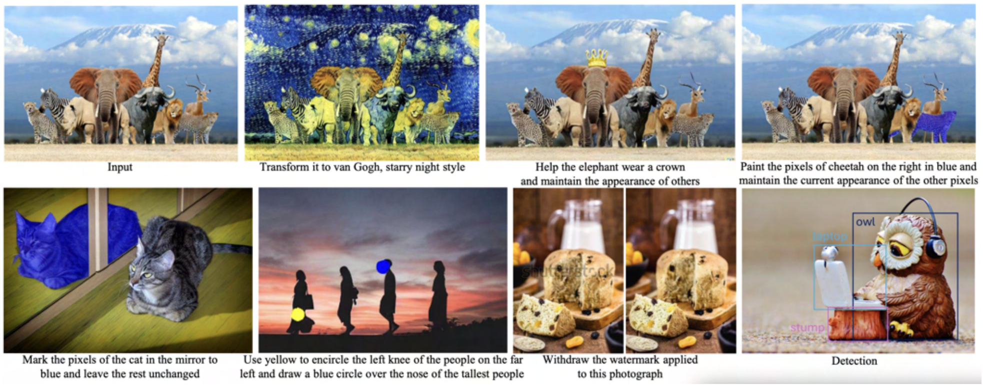

Step 1
A running tiger
Text-to-image Model
Step 2
A running tiger with yellow and blue circles
Pretraining Adaptation
Step 3
Use the color yellow to encircle the nose of the tiger
Task-specific Training
Step 4
Use the color yellow to encircle the nose of the tiger
Human Alignment

Figure 2.11: InstructDiffusion - Approach Proposed

The InstructDiffusion framework represents a notable advancement in aligning computer vision tasks with human instructions, offering a unified approach across various vision tasks such as image editing, segmentation, keypoint detection, object detection, and low-level vision. Built upon the diffusion process, this framework trains the model to predict pixels based on detailed user instructions, allowing for tasks like encircling specific areas, applying masks, or adding/removing elements from images. Notably, the framework showcases its ability to handle unseen tasks not encountered during training, outperforming previous methods on unseen datasets. Multi-task learning is highlighted for its role in improving the model's generalization ability, while human alignment further fine-tunes the model for enhanced performance.
Despite these strengths, potential limitations include the need for further exploration in generalization and adaptability to entirely new tasks, potential constraints in model performance in certain scenarios, and the complexity of the model and training process, which may require extensive computational resources and impact widespread deployment. Additionally, while the framework exhibits versatility, there may be challenges related to information loss during discretization and potential biases in the training data quality stemming from web-crawled data and user requests.


Input

Transform it to van Gogh, starry night style

Help the elephant wear a crown and maintain the appearance of others

Paint the pixels of cheetah on the right in blue and maintain the current appearance of the other pixels

Mark the pixels of the cat in the mirror to blue and leave the rest unchanged

Use yellow to encircle the left knee of the people on the far left and draw a blue circle over the nose of the tallest people

Withdraw the watermark applied to this photograph

Detection

Figure 2.12: InstructDiffusion - Result

2.3.2 SmartEdit: Exploring Complex Instruction-based Image Editing with Multimodal Large Language Models[2]

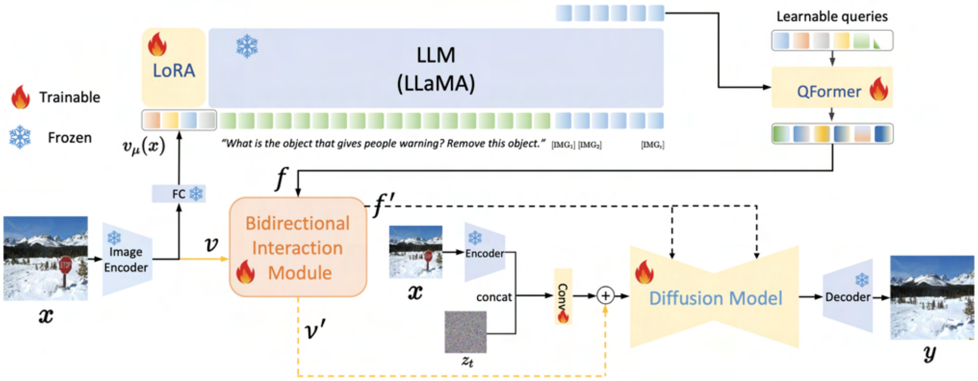


Figure 2.13: SmartEdit - Approach Proposed

SmartEdit is a novel approach to instruction-based image editing that leverages Multimodal Large Language Models (MLLMs) to enhance understanding and reasoning capabilities .The Bidirectional Interaction Module enables comprehensive bidirectional information interactions between the input image and the output of LLaVA, facilitating complex reasoning. A small amount of complex instruction editing data effectively stimulates SmartEdit's editing capabilities for more complex instructions. The Reason-Edit dataset is constructed specifically for evaluating complex instruction-based image editing methods. SmartEdit surpasses previous methods in both quantitative and qualitative results on the Reason-Edit dataset, demonstrating its effectiveness in complex instruction-based image editing.

Existing instruction-based editing methods, such as InstructPix2Pix[5], often fail to produce satisfactory results in complex scenarios due to their dependence on the simple CLIP text encoder in diffusion models. The paper addresses the challenge of complex reasoning in instruction-based image editing by proposing a Bidirectional Interaction Module that enables comprehensive bidirectional information interactions between the input image and the output of LLaVA.

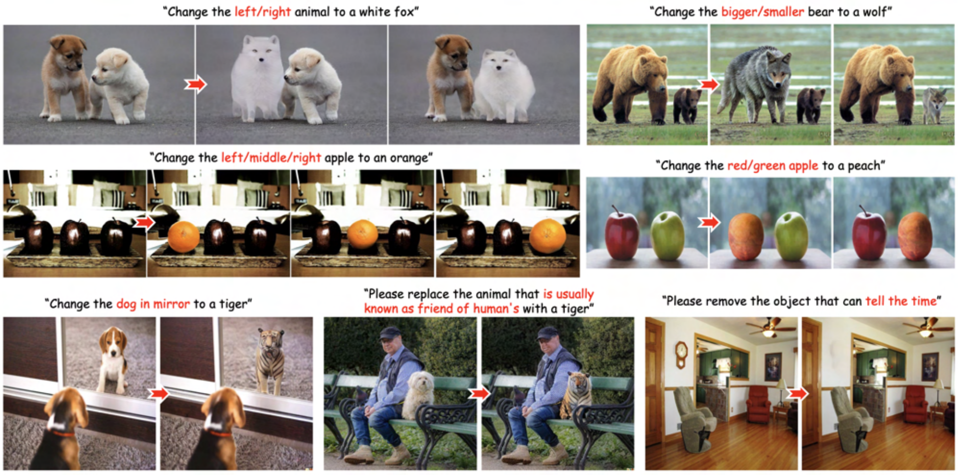

"Change the left/right animal to a white fox"
"Change the bigger/smaller bear to a wolf"
"Change the left/middle/right apple to an orange"
"Change the red/green apple to a peach"
"Change the dog in mirror to a tiger"
"Please replace the animal that is usually known as friend of human's with a tiger"
"Please remove the object that can tell the time"

Figure 2.14: SmartEdit - Result

	Summary	**Results**	**Advantages**	**Limitations**
Instruct-NeRF2NeRF [3]	Utilizes an image-conditioned diffusion model. Iteratively updates input images while optimizing the underlying NeRF. Enables a diverse range of local and global scene edits. Democratize 3D scene editing Make it accessible and intuitive for everyday users.	Demonstrates ability to edit a variety of real scenes. Includes environmental and localized changes. Qualitative and quantitative evaluations showcase effectiveness.	Allows for intuitive, instruction-driven edits on 3D scenes. Enables contextual and large-scale edits. Democratizes 3D scene editing.	Unable to perform large spatial manipulations. Occasional difficulty in performing binding between objects referred to in the instruction and the corresponding scene objects. May suffer from artifacts in the edited NeRF scenes. Textures in the edited scenes may appear slightly blurrier.
Free-Editor[6]	Allows for editing 3D content without re-training models during testing. Addresses the challenge of multi-view style consistency in 3D scenes Leverages a generalized NeRF model and a set of loss functions to ensure spatial smoothness and color rendering consistency across views.	Outperforms state-of-the-art methods in terms of runtime efficiency, achieving nearly 20 times faster editing times. Demonstrates better space efficiency with a constant space complexity.	Significant reduction in editing time and memory requirements due to avoidance of re-training. Ability to perform diverse editing capabilities without compromising the integrity of the original scene. Greater flexibility and ease of use across various editing styles and scenes.	Requires several iterations of trial and error to achieve desired editing effects. Accurate 2D editing of the starting view can impact the overall 3D scene's editing outcome. May require a view-filtering system or repeated editing to address.
LatentEditor [8]	Enable precise and locally controlled editing of neural fields using text prompts. Faster and more adaptable NeRF editing compared to traditional methods. Achieves superior performance in text fidelity, content preservation, and scene consistency.	Demonstrates superior performance in text fidelity, content preservation, and scene consistency. Achieves high scores in user studies.	Exhibits enhanced speed and resource efficiency in NeRF editing. Enables local editing while maintaining background integrity. Achieves multi-attribute editing capabilities.	Efficacy of LatentEditor is contingent on capabilities of pre-trained IP2P model. Model's effectiveness in executing specific editing instructions may be limited.
Blending-NeRF[7]	Combines pretrained and editable NeRF networks. Achieves precise and localized object editing. Outperforms existing baselines in qualitative and quantitative evaluations. Demonstrates effectiveness in achieving natural and localized 3D object editing from various text prompts.	Outperforms all baselines in a user study. Achieves the highest mean score for text-driven editing precision. Demonstrates superior qualitative and quantitative performance.	Enables precise and localized editing of 3D objects by text prompts. Achieves natural and localized editing from various text prompts. Enables complete stylization, including density-based localized editing.	Affected by off-the-shelf models like CLIPSeg. Limitations observed in certain editing tasks, such as editing the shape of a single NeRF by a text prompt.
GO-NeRF[11]	Generate text-controlled 3D virtual objects within existing NeRF-based scenes. Leverages a compositional rendering formulation. Tailored optimization objectives. Utilizes image priors from pre-trained text-guided image inpainting networks. Ensures high-quality and scene-harmonious results.	Successfully generates vibrant virtual objects within scenes, such as Pikachu and a pumpkin. Produces objects with well-defined shapes and reasonable poses. Leverages image priors from pre-trained text-guided image inpainting networks.	Leverages scene context for virtual object generation. Outperforms previous methods on both feed-forward and 360-degree datasets. Offers a user-friendly interface for defining object positions in 3D scenes.	May not be capable of modifying areas not encompassed by the specified 3D box, such as reflections. Subject to limitations of the SDS loss, such as the Janus problem.

Table 2.2: NeRF Literature Review

Chapter 3

Preliminaries

3.1 Diffusion Models

A diffusion model is a probabilistic generative model used in various fields, including machine learning, image processing, and data generation. This model is designed to capture and model complex data distributions by iteratively diffusing or spreading information across data points in a sequential manner. What we particularly want to implement, is a specific set of models called Denoising Diffusion Probabilistic Models (DDPMs).

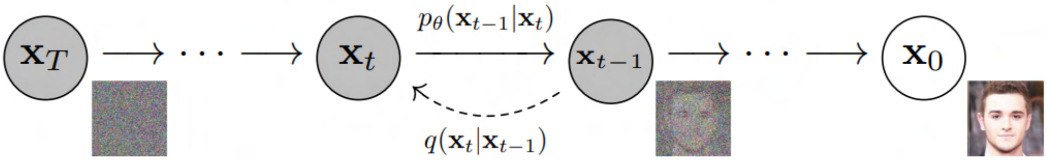

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Figure 3.1: Forward and Reverse Diffusion[3].

The model is made up of two phases:

- Forward diffusion q that gradually adds random noise in each iteration
- Reverse diffusion p_θ, where a neural network works to remove the noise to return back to the original image

3.2 Software/Hardware Requirements for Project

3.2.1 Software Requirements

1. **Python:** Python is the primary programming language for implementing machine learning algorithms, data processing, and visualization.
2. **Deep Learning Frameworks:** PyTorch or TensorFlow. These deep learning frameworks provide a wide range of tools and modules for building and training neural network models, including implementations of NeRF and diffusion models.
3. **NerfStudio:** Utilize NerfStudio for training the NeRF scene using the nerfacto model. NerfStudio offers a convenient interface for NeRF training with various optimizations.
4. **Text Processing Libraries:** Libraries like NLTK (Natural Language Toolkit) or SpaCy for text processing and natural language understanding to handle textual inputs for editing instructions.

5. **Image Editing Libraries:** Libraries like OpenCV or PIL (Python Imaging Library) for image processing and editing, particularly for generating masks and applying edits to NeRF scenes.

3.2.2 Hardware Requirements

1. **GPU (Graphics Processing Unit):** A powerful GPU is essential for training deep learning models efficiently. Nvidia GPUs such as GeForce RTX or A100 would be required.

2. **RAM (Random Access Memory):** Adequate RAM is necessary to handle large datasets and model parameters during training and inference. A minimum of 16GB RAM is recommended, although higher capacity (32GB or more) may be beneficial for handling larger scenes and datasets.

Chapter 4

Methodology

4.1 Proposed Methodology

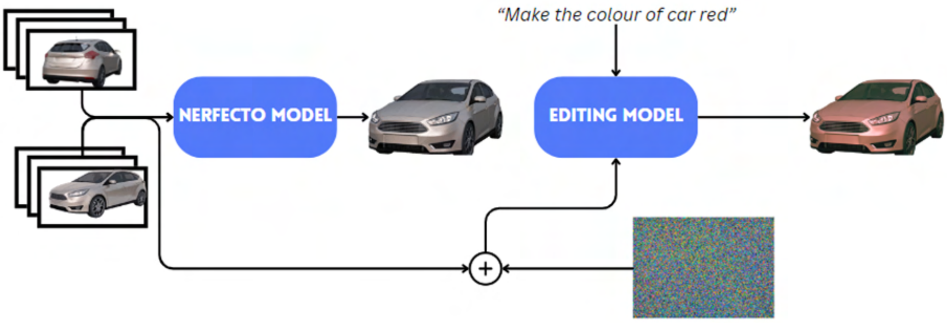


Figure 4.1: Proposed architecture

Our method takes textual prompts and a base NeRF model as input and returns another NeRF model which is the modification of the original model based on the prompt. Our method takes inspiration from recent implementations and research papers, such as the LatentEditor [13] and IN2N [15], to propose a novel approach for editing NeRF scenes with improved efficiency and consistency. Fig. 5.1 shows some results from our approach.

Initially a number of images along with their camera parameters are used to generate a base NeRF model. We use the nerfacto model by Nerfstudio [19] for the base model. This base model and the text prompt is provided to our model. We have extended the IN2N implementation to change the 2D diffusion model, InstructPix2Pix [23], with InstructDiffusion [24]. We also add masks to gain ability of localized edits in the scene. We achieve this by employing the Lang-SAM model [25].

4.1.1 Training a NeRF Scene with nerfacto Model

- Our methodology involves leveraging the Nerfacto model to synthesize a 3D NeRF (Neural Radiance Fields) scene using input images and corresponding camera locations. The Nerfacto model, developed by the creators of Nerfstudio, represents a significant advancement in NeRF-based scene generation frameworks. Nerfstudio itself provides a streamlined end-to-end pipeline for creating, training, and evaluating NeRF models, enhancing interpretability by modularizing key components.

- The Nerfacto model was selected for its unique combination of techniques, including camera pose refinement, per-image appearance conditioning, proposal sampling, scene contraction, and hash encoding. This amalgamation of approaches contributes to the model's efficiency and speed, distinguishing it from other NeRF implementations while delivering commendable results in scene reconstruction.

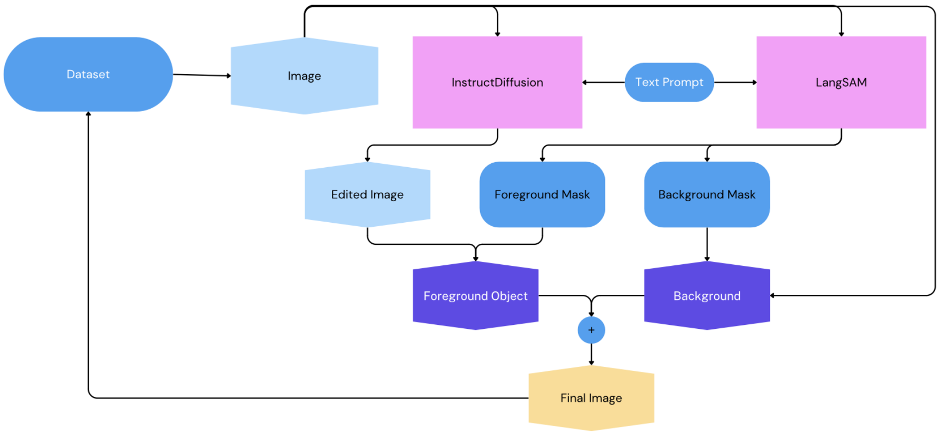


Figure 4.2: **Detailed Architecture** The process that happens during the editing part. We employ the combination of InstructDiffusion and Lang-SAM models to edit the NeRF scene efficiently

- Specifically, the incorporation of camera pose refinement ensures accurate alignment of rendered viewpoints with the captured images. Per-image appearance conditioning allows for fine-grained adjustments to the appearance of individual scenes or objects within the NeRF context. Proposal sampling enhances the model's ability to explore diverse scene configurations efficiently. Scene contraction aids in managing computational complexity by focusing on key regions of interest, while hash encoding optimizes data representation and retrieval.

- The Nerfacto model's efficiency and performance make it a preferred choice for our research, enabling effective synthesis of high-quality 3D scenes from input images and camera poses. This selection aligns with our goal of achieving accurate and scalable 3D reconstruction within a NeRF framework.

4.1.2 Editing Using Diffusion Model

- We employ a diffusion model for localized editing of the masked regions in the NeRF scene. Diffusion models are powerful probabilistic generative models capable of generating realistic images conditioned on textual descriptions.

- By integrating the diffusion model into the pipeline, we enable seamless editing of the NeRF scene while preserving its consistency and realism. The diffusion model operates in 2D space, allowing for efficient addition, removal or editing of objects without the need for retraining the entire NeRF model or changing the dataset.

InstructDiffusion: A Generalist Modeling Interface for Vision Tasks

In our approach, we leveraged the InstructDiffusion framework to achieve precise and intuitive image manipulation guided by natural language instructions. The InstructDiffusion model, rooted in diffusion techniques, is designed to predict pixel values based on human-specified commands, such as ”circle the left shoulder of the man with red and place a blue mask on the left car." This method enables fine-grained control over image modifications, aligning closely with user intentions expressed through text prompts.

Compared to traditional methods like InstructPix2Pix, InstructDiffusion offers several key advantages. First, InstructDiffusion adopts a more user-friendly and interactive approach by directly translating natural language instructions into pixel-level predictions, eliminating the need for predefined output spaces or complex mappings. Additionally, InstructDiffusion demonstrates superior generalization capabilities across diverse vision tasks, including segmentation, keypoint detection, editing, and restoration. Its adaptability to novel tasks and datasets outpaces existing methods, showcasing its versatility as a generalist modeling interface for various image-related applications. Furthermore, InstructDiffusion's human-alignment design ensures that its outputs closely match user expectations, enhancing the interpretability and usability of the model in practical image editing scenarios. By focusing on flexible and interactive pixel space manipulation guided by natural language, InstructDiffusion represents a significant advancement in bridging the gap between human intent and computer vision outputs.

4.1.3 Extractor for building mask

In our experimentation with the InstructDiffusion model, we encountered a challenge illustrated in Figure 4.3. The issue stemmed from the model's tendency to alter the entire image rather than focusing on specific regions as desired. To address this, we introduced the Language Segment Anything [25] model into our workflow.

Initially, we applied the instructdiffusion model to the entire image for editing purposes. Subsequently, we identified and isolated the specific area within the image that required modification. This step was crucial in minimizing the risk of inadvertently altering unrelated portions of the image. To achieve this, we adopted a semantic breakdown of the input prompt to isolate the object-related component.

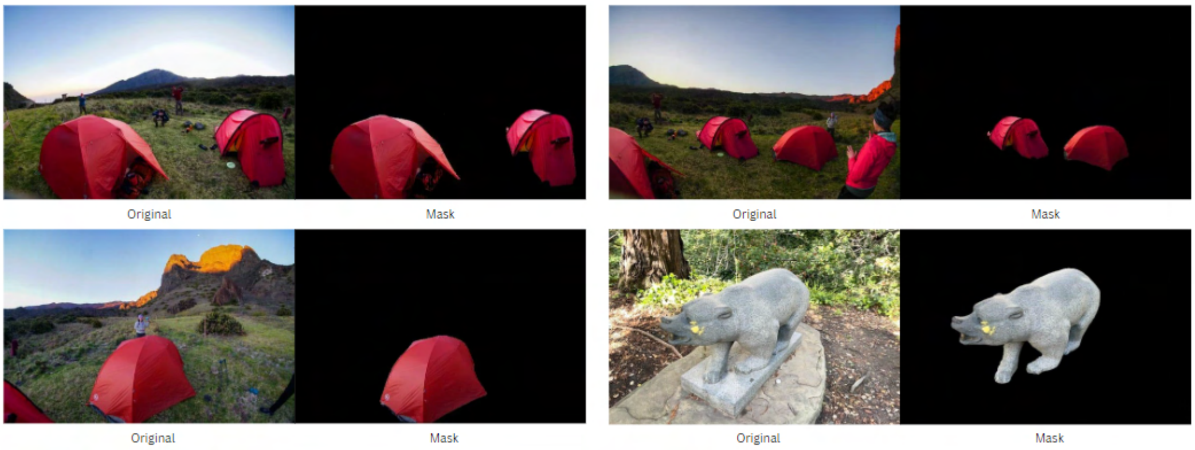


Figure 4.3: Some of the outputs extracted from Lang-SAM during the editing process

The extracted object segment was then utilized as input for the Lang-SAM model. The Lang-SAM model was specifically employed to extract and manipulate only the identified portion of the image that corresponded to the specified object from the prompt. This targeted approach facilitated precise and controlled image editing, effectively addressing the challenge posed by the InstructDiffusion model's tendency to globally modify images.

The Lang-SAM, implements an innovative method that combines instance segmentation with text prompts to generate masks for specific objects in images. This open-source initiative leverages the Meta model, segment-anything [29], and the GroundingDINO [31] detection model to achieve accurate object detection and image segmentation. The project integrates a zero-shot text-to-bbox approach for object detection, enabling users to generate masks for desired objects based on text inputs.

The methodology involves a modified transformer encoder serving as the mask decoder to translate encoded prompts and image embeddings into object masks. The model, named SAM, generates three relevant masks for a given prompt, providing users with multiple segmentation options. SAM [29] is trained using a combination of focal, dice, and IoU loss functions to optimize performance.

During training, SAM utilizes CLIP-based prompts [30] followed by iterative point prompts for refinement. During inference, the unmodified CLIP text encoder is used to create prompts for SAM, enabling precise and context-aware segmentation results. This methodology effectively harnesses natural language prompts for accurate segmentation and object detection.

4.1.4 Iterative Refinement for Consistency

The Iterative Dataset Update (Iterative DU) process described here is adapted from the InstructNerf2Nerf research paper. This approach involves a sequential alternation between rendering images from the NeRF, updating them with the InstructDiffusion model, and using these updated images to supervise NeRF reconstruction.

In this method, the training dataset starts with original captured images I_{0v} from various viewpoints v. During each iteration, a series of image updates d and NeRF updates n are performed. Image updates modify the dataset images based on instructions provided by the InstructDiffusion model, while NeRF updates sample random rays from the entire dataset, incorporating a mixture of old and recently updated information. Over iterations, edited images replace their original counterparts, gradually converging towards a globally consistent depiction of the edited scene.

This method is a crucial component of our research, adapted from the InstructNerf2Nerf approach, and leverages the InstructDiffusion model for image editing within the NeRF framework.

Chapter 5

Experimental Analysis

In our research, we conducted experiments on real-world scenes that were enhanced using Nerfstudio [19]. These scenes encompassed a diverse range of complexities, including 360-degree views of environments and objects, portraits of faces, and full-body compositions. We used datasets from IN2N [15] and NeRFStudio [19]. Some scenes from our labroratory were captured using our smartphone camera. Camera poses were determined using either Kiri Engine [22] or through the PolyCam [20] apps, resulting in datasets consisting of 50 to 300 images each. Our evaluation process involved comprehensive qualitative assessments and comparisons with state-of-the-art (SOTA) models to validate our methodological decisions.

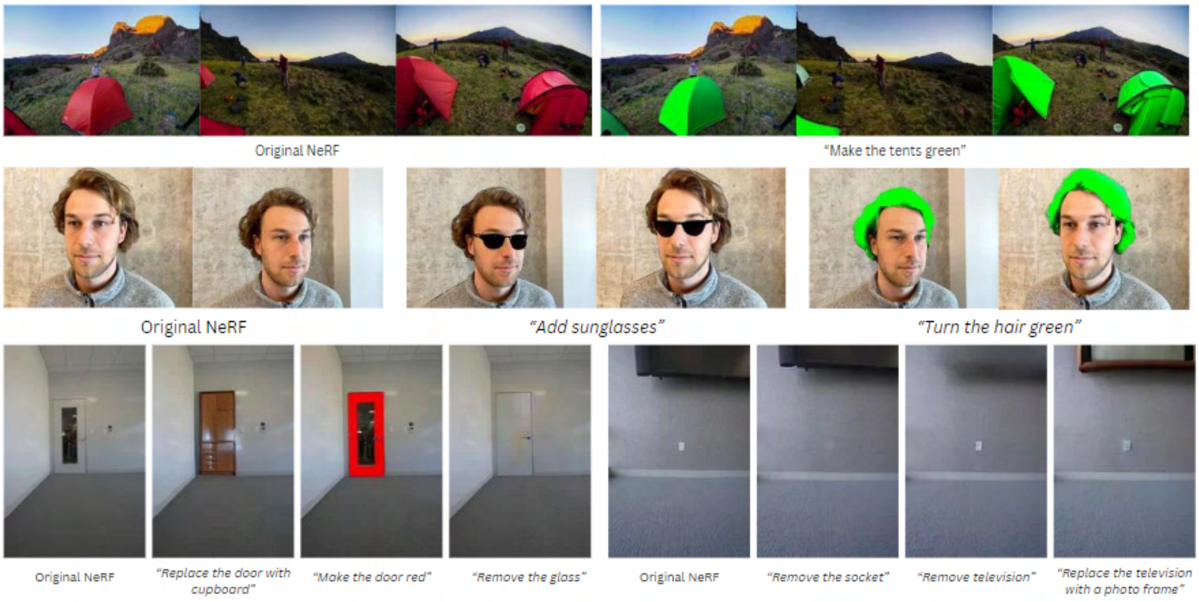


Figure 5.1: **Results** We showcase the outcomes of our approach across a range of scenes. Each scene is presented with its corresponding input views displayed on the left, followed by the results achieved from various text prompts.

5.1 Qualitative evaluation

Our research presents compelling qualitative outcomes exemplified in Figures 5.2 and 5.1 showcasing the versatility and effectiveness of our approach in diverse image editing tasks. Each edit is depicted from multiple viewpoints to emphasize the consistency and quality of our methodology.

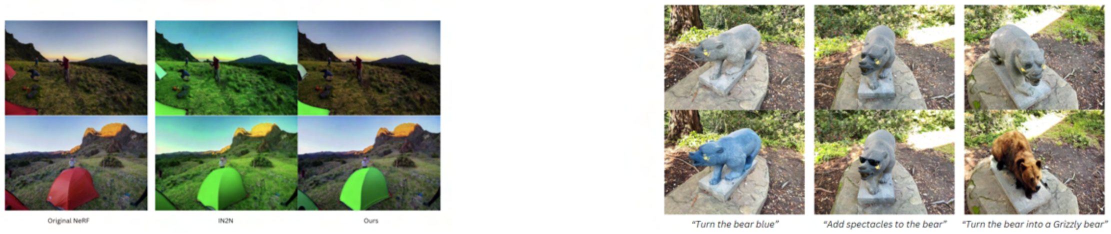
In Figure 5.1, we demonstrate precise localized edits such as altering hair color, underscoring our capability to efficiently modify specific elements within objects and scenes.

Furthermore, our method adeptly handles tasks like adding small objects such as sunglasses, removing unwanted elements like sockets, and refining object details such as removing glass from a door. These capabilities address limitations observed in prior methods like InstructNerf2Nerf.

Moreover, our approach excels in object replacement tasks, exemplified by seamlessly replacing a door with a wooden cupboard or transforming a television into a frame. Even nuanced object parts within the frame are effectively handled, as demonstrated by the precise replacement of television components with frames.

Additionally, we highlight the impactful outcomes achieved using Lang-SAM (Language Segment-Anything Model), which plays a pivotal role in ensuring precise and consistent image edits. Lang-SAM demonstrates exceptional efficiency in accurately identifying and segmenting objects within images, enabling targeted and high-quality edits while preserving scene coherence.

For instance, in the showcased results Figure 4.3, Lang-SAM efficiently isolates specific objects like tents for color adjustments. Notably, this includes precise segmentation of both primary objects and subtle components like tent edges, ensuring comprehensive and detailed edits. The ability of Lang-SAM to accurately segment objects and their parts significantly contributes to achieving refined and visually appealing image modifications.


Original NeRF
IN2N
Ours
"Turn the bear blue"
"Add spectacles to the bear"
"Turn the bear into a Grizzly bear"

(a) Comparisons with IN2N

(b) Results on bear dataset

Figure 5.2: Results

Our qualitative findings underscore the effectiveness of our approach across diverse scenarios and instructions, including modifications to environmental conditions like time of day, seasons, and weather settings such as snow and desert. This versatility and robustness highlight the broad applicability of our method in diverse image editing contexts.

Furthermore, a notable distinction between our approach and InstructNerf2Nerf is evident in the tent scene, particularly regarding tent color adjustments Figure **??**. Unlike InstructNerf2Nerf, which applies color changes globally to the entire scene, our mask-based approach precisely targets and colors only the tents while preserving the fidelity of the surrounding environment. This difference underscores the advantage of our method, leveraging segmentation masks from Lang-SAM to achieve controlled and contextually accurate edits tailored to specific objects within complex scenes.

5.2 Quantitative Results

In our quantitative evaluation, we compared our proposed approach utilizing InstructDiffusion with state-of-the-art methods such as NeRF-Art, Control4D, DreamEditor, InstructNerf2Nerf (IN2N), LatentEditor, and our method. We measured the alignment of

Method	CLIP Text-Image Direction Similarity (↑)	CLIP Direction Consistency (↑)
NeRF-Art	0.2755	0.9672
Control4D	0.2503	0.9751
DreamEditor	0.2604	0.9802
IN2N	0.2788	0.9850
LatentEditor	0.2801	0.9881
Ours	0.2785	0.9747

Table 5.1: Comparison of CLIP-based Metrics

the performed 3D edits with text instructions using the CLIP Text-Image Direction Similarity metric. The results, summarized in Table **??**, reveal notable improvements achieved by our method. We observed that our approach exhibited higher CLIP Text-Image Direction Similarity compared to IN2N and LatentEditor, indicating improved alignment between textual instructions and 3D edits. Additionally, our method demonstrated competitive CLIP Direction Consistency scores, showcasing strong temporal consistency across different views, comparable to or surpassing existing state-of-the-art approaches. These quantitative metrics provide valuable insights into the efficacy and performance of our proposed method in comparison to current state-of-the-art techniques in instruction-based image editing.

5.2.1 Limitations

Despite the promising results demonstrated by our approach, several limitations must be acknowledged. The performance of our method heavily relies on the capabilities and effectiveness of the InstructDiffusion model. Therefore, any inherent limitations or shortcomings of InstructDiffusion could potentially impact the overall performance and quality of our image editing results. Additionally, our reliance on Lang-SAM for object detection and segmentation introduces specific challenges. For instance, Lang-SAM may struggle to accurately detect certain objects or parts of objects, particularly when prompted with unclear instructions or when dealing with very small or obscured objects within the frame.

Moreover, in some edits where the position or structure of an object is altered, noticeable noise or artifacts may be observed in the edited image. This issue could be mitigated by integrating an inpainting model into our pipeline. By employing an inpainting technique, we could effectively remove and inpaint the original object in the image, subsequently incorporating the edited part from the modified image. This strategy would enhance the overall visual quality and cohesiveness of our edits, addressing challenges related to structural changes and noise introduced during object replacement.

Chapter 6

Conclusion

Our research presents a novel approach to instruction-based image editing leveraging the synergistic capabilities of InstructDiffusion and Lang-SAM models. Through our method, we have demonstrated compelling qualitative results showcasing the versatility and effectiveness of our approach in achieving diverse and complex image edits. Our method excels in localized and precise edits, enabling tasks such as object addition, removal, and replacement with high fidelity and context-awareness.

While our approach exhibits promising results, it is important to acknowledge certain limitations inherent to the underlying models. The reliance on InstructDiffusion introduces dependencies on its capabilities, and challenges with object detection and segmentation may arise with Lang-SAM in certain scenarios.

Looking ahead, future work could focus on addressing these limitations by integrating advanced inpainting techniques to refine structural changes and mitigate noise in edited images. Additionally, exploring ways to enhance the robustness and adaptability of our approach to a broader range of tasks and scenarios would be a valuable direction for further research in this field. Overall, our research contributes to the evolving landscape of instruction-based image editing, demonstrating the potential of language-guided models to empower users in creating rich and contextually accurate visual content.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, 'NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis', arXiv [cs.CV]. 2020.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, 'Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields', arXiv [cs.CV]. 2021.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, 'Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields', arXiv [cs.CV]. 2023.

[4] T. Müller, A. Evans, C. Schied, and A. Keller, 'Instant Neural Graphics Primitives with a Multiresolution Hash Encoding', ACM Trans. Graph., vol. 41, no. 4, p. 102:1-102:15, Jul. 2022.

[5] H. Turki, M. Zollhöfer, C. Richardt, and D. Ramanan, 'PyNeRF: Pyramidal Neural Radiance Fields', arXiv [cs.CV]. 2023.

[6] "Nerfacto," docs.nerf.studio. https://docs.nerf.studio/nerfology/methods/nerfacto.html

[7] J. Zhang, X. Li, Z. Wan, C. Wang, and J. Liao, 'Text2NeRF: Text-Driven 3D Scene Generation with Neural Radiance Fields', arXiv [cs.CV]. 2024.

[8] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, 'CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields', arXiv [cs.CV]. 2022.

[9] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, 'DreamFusion: Text-to-3D using 2D Diffusion', arXiv [cs.CV]. 2022.

[10] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao, 'NeRF-Art: Text-Driven Neural Radiance Fields Stylization', arXiv [cs.CV]. 2022.

[11] O. Gordon, O. Avrahami, and D. Lischinski, 'Blended-NeRF: Zero-Shot Object Generation and Blending in Existing Neural Radiance Fields', arXiv [cs.CV]. 2023.

[12] H. Song, S. Choi, H. Do, C. Lee, and T. Kim, 'Blending-NeRF: Text-Driven Localized Editing in Neural Radiance Fields', arXiv [cs.CV]. 2023.

[13] U. Khalid, H. Iqbal, N. Karim, J. Hua, and C. Chen, 'LatentEditor: Text Driven Local Editing of 3D Scenes', arXiv [cs.CV]. 2024.

[14] N. Karim, U. Khalid, H. Iqbal, J. Hua, and C. Chen, 'Free-Editor: Zero-shot Text-driven 3D Scene Editing', arXiv [cs.CV]. 2023.

[15] A. Haque, M. Tancik, A. A. Efros, A. Holynski, and A. Kanazawa, 'Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions', arXiv [cs.CV]. 2023.

[16] E. Bartrum, T. Nguyen-Phuoc, C. Xie, Z. Li, N. Khan, A. Avetisyan, D. Lanman and L. Xiao, 'ReplaceAnything3D:Text-Guided 3D Scene Editing with Compositional Neural Radiance Fields', arXiv [cs.CV]. 2024.

[17] M. Shahbazi, L. Claessens, M. Niemeyer, E. Collins, A. Tonioni, L. V. Gool and F. Tombari, 'InseRF: Text-Driven Generative Object Insertion in Neural 3D Scenes', arXiv [cs.CV]. 2024.

[18] J. Chen, B. Ji, Z. Zhang, T. Chu, Z. Zuo, L. Zhao, W. Xing and D. Lu, 'TeSTNeRF: text-driven 3D style transfer via cross-modal learning', in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023.

[19] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister and A. Kanazawa, 'Nerfstudio: A Modular Framework for Neural Radiance Field Development', in Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings, 2023.

[20] Polycam. Polycam - lidar and 3d scanner for iphone and android.

[21] J. L. Schönberger and J.-M. Frahm, 'Structure-from-Motion Revisited', in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4104–4113.

[22] K. Engine, "KIRI Engine: 3D Scanner app for iPhone, Android and Web," KIRI Engine: 3D Scanner App. https://www.kiriengine.app/

[23] T. Brooks, A. Holynski, and A. A. Efros, 'InstructPix2Pix: Learning to Follow Image Editing Instructions', arXiv preprint arXiv:2211. 09800, 2022.

[24] Z. Geng, B. Yang, T. Hang, C. Li, S. Gu, T. Zhang, J. Bao, Z. Zhang, H. Hu, D. Chen and B. Guo, 'InstructDiffusion: A Generalist Modeling Interface for Vision Tasks', arXiv [cs.CV]. 2023.

[25] Luca-Medeiros, "GitHub - luca-medeiros/lang-segment-anything: SAM with text prompt," GitHub. https://github.com/luca-medeiros/lang-segment-anything

[26] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu and L. Zhang, 'Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection', arXiv [cs.CV]. 2023.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár and R. Girshick, 'Segment Anything', arXiv [cs.CV]. 2023.

[28] P. Dai, F. Tan, X. Yu, Y. Zhang, and X. Qi, 'GO-NeRF: Generating Virtual Objects in Neural Radiance Fields', arXiv [cs.CV]. 2024.

[29] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár and R. Girshick, 'Segment Anything', arXiv [cs.CV]. 2023.

[30] W. Chen, Q. Ke, and Z. Li, ‘CLIP Guided Image-perceptive Prompt Learning for Image Enhancement’, arXiv [cs.CV]. 2023.

[31] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu and L. Zhang, ‘Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection’, arXiv [cs.CV]. 2023.