RBE 549 Computer Vision
P2 - Phase 2 - NeRF

Taruneswar Ramuu
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, MA, USA
Email: tramuu@wpi.edu

Soumik Saswat Patnaik
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, MA, USA
Email: sspatnaik@wpi.edu

***Abstract*—This study replicates and extends Neural Radiance Fields (NeRF) for synthesizing novel scene views from multiple photographs. By leveraging NeRF's neural network architecture, we learn a continuous, volumetric representation of scenes from sets of images and their camera poses. This methodology is tested on three datasets, including two pre-existing and one self-compiled, demonstrating our implementation's adaptability. A testing pipeline evaluates model performance on new viewpoints, using metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for quality assessment. Results underline NeRF's capability in photorealistic rendering from limited observational data, contributing to computer vision research.**

***Index Terms*—Image-based Rendering, Volume Rendering, Neural Radiance Fields, Scene Reconstruction, Volumetric Rendering, Positional Encoding**

Introduction

The ability to synthesize novel views of scenes has significant implications across various fields, including computer graphics, virtual reality, and augmented reality. Traditional methods often rely on explicit geometric representations or depth information, limiting their applicability to scenes with simple geometry and appearance. In contrast, recent advancements in neural rendering have shown promising results in synthesizing realistic views of complex scenes without explicit geometric priors.

This paper builds upon the concept of neural radiance fields, leveraging a fully-connected deep network, whose input is a single continuous 5D coordinate (spatial position: (x, y, z) and viewing direction: (θ, ψ)) and output is the volume density and the RGB pixel values at that viewing direction, to represent scene properties continuously. By optimizing this representation using a sparse set of input views, the proposed method can synthesize photorealistic novel views of scenes with intricate geometry and appearance.

I. Neural Radiance Fields - NeRF

A. Dataset

The datasets provided are retrieved from the original datasets used by the original author of the NeRF implementation. For our project, we have used the Lego and ship datasets from the above. Along with these, we have also implemented a NeRF implementation on a dataset created on our own.

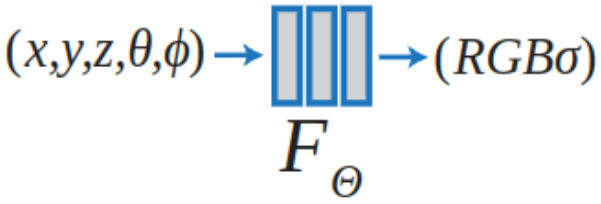


Fig. 1: NeRF Network Architecture

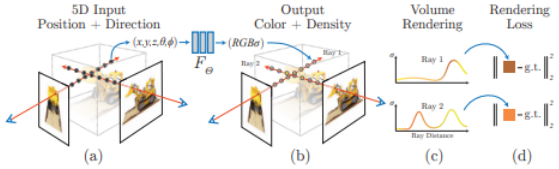


Fig. 2: An overview of NeRF representation and rendering procedure.

B. NeRF Architecture

Neural Radiance Fields (NeRF) [Figure 2] introduce a groundbreaking method for synthesizing photorealistic 3D scenes from a sparse set of images. This approach relies on a fully connected deep neural network to model a scene as a continuous, high-dimensional function that maps 5D coordinates (comprising 3D spatial positions and 2D viewing directions) to color and volume density. The model is trained using a combination of photometric loss and structural similarity metrics to ensure accurate reconstruction of the scene geometry and appearance. The original NeRF architecture meticulously renders complex scenes with intricate lighting and material properties by integrating the contributions of light along camera rays, a process enabled by differentiable volume rendering. This technique allows for the creation of detailed 3D reconstructions from limited photographic data, capturing nuances of light and shadow with high fidelity.

C. Our Architecture

Building upon the original NeRF framework, we implemented modifications to adapt the architecture to our specific

project needs, focusing on simplification and computational efficiency. Our version includes:

1) A reduced network width of 64 channels, optimizing processing speed while retaining the ability to capture essential scene details.
2) Positional encoding limited to 16 frequencies, balancing the model's spatial awareness with the need for a streamlined input representation.
3) An implementation focused solely on the coarse network, foregoing the fine-resolution network to enhance training and rendering speed.
4) Adherence to the original 8-layer network structure, ensuring depth consistency while simplifying other aspects of the model.
5) Inspiration drawn from tinynerf, a variant proposed by the original authors, guiding our efforts to achieve a more compact and efficient implementation.

These modifications enable our architecture to efficiently produce high-quality 3D renders from sparse image sets, balancing detail and computational demand to meet the unique challenges of our project.

II. Results Analysis

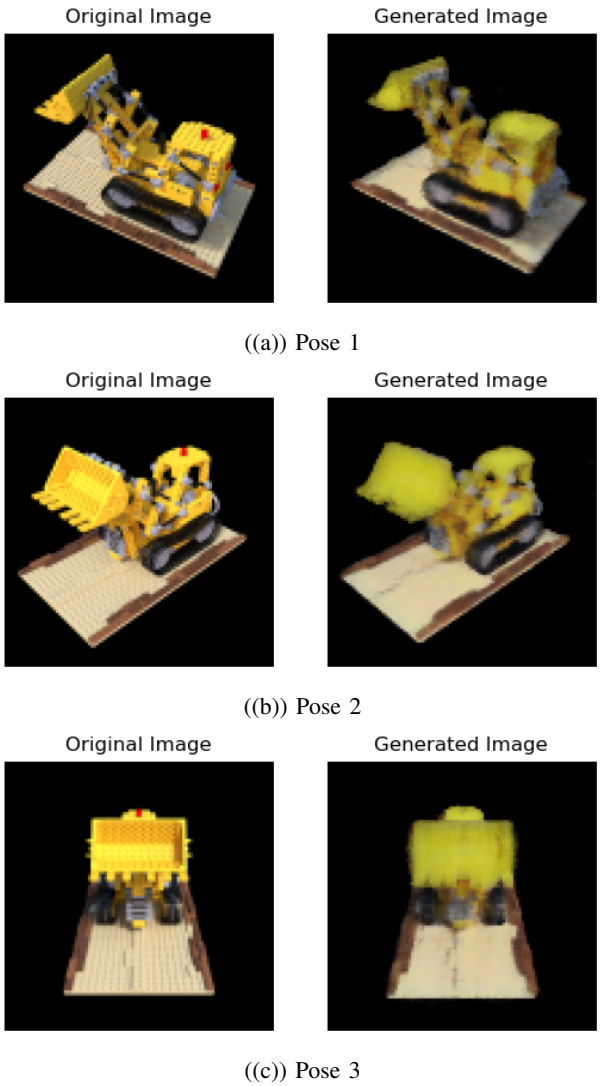
1) Efficiency and Quality of 3D Models: Our modified NeRF model demonstrated remarkable efficiency, producing high-quality 3D models within just 40 iterations for both datasets. This performance is a testament to the effectiveness of our implementation. While accuracy and model quality improved up to approximately 500 iterations, further training led to overfitting, resulting in quality degradation due to distortions. Please refer to Figure 6.

2) Impact of Positional Encoding: A critical comparison between rendered outputs with and without positional encoding highlighted its significance. Positional encoding not only enhanced the quality of the renders but also expedited the rendering process, confirming its pivotal role in achieving superior NeRF outcomes.

3) Challenges in Model Training: We encountered a challenging issue where the neural network would occasionally enter an indefinite execution state without producing viable outputs. This problem was traced back to the initial training seed. Implementing a `random.seed()` function allowed us to substantially mitigate this issue, highlighting the sensitivity of model training to initial conditions.

4) Invaluable Insights into Image Handling: Throughout the debugging process, we gained important insights into image handling in deep learning contexts. We learned the importance of image normalization, which simplifies mathematical operations, and discovered techniques for manipulating a single image element through tensors, varying shapes, and types within a loop for efficient and effective processing. This knowledge was crucial for enhancing our model's performance and reliability.

5) Model Evaluation and Dataset Limitations: Our initial attempts at reconstructing the desired 3D models using the Neural Radiance Fields (NeRF) approach have not yet achieved the desired level of accuracy. This suggests a potential for significant improvement through extended training periods. However, upon a thorough evaluation of our experimental outcomes, we have identified that the primary impediments to achieving optimal results lie within the dataset itself. Please refer to Figure 11 and Figure 12.



((a)) Pose 1

((b)) Pose 2

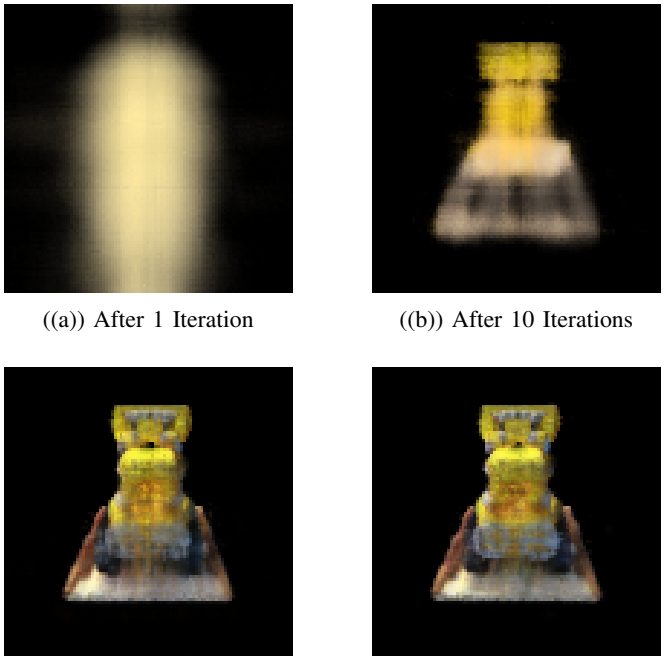
((c)) Pose 3

Fig. 3: Lego Testing - Original vs Generated

Various dataset-related factors contribute to the observed reconstruction inaccuracies. In particular, inconsistencies in lighting conditions, variations in background colors, and the presence of blurriness within the images collectively introduce artifacts that adversely affect the model's performance. These issues highlight the critical importance of dataset quality in successfully applying NeRF techniques.

Given these insights, we believe that refining the dataset to ensure uniform lighting, consistent background colors, and sharp imagery would substantially enhance the fidelity of the 3D reconstructions. Prioritizing the creation of such an ideal dataset in future research endeavors will be essential to fully

leverage the capabilities of NeRF for high-quality 3D model synthesis.



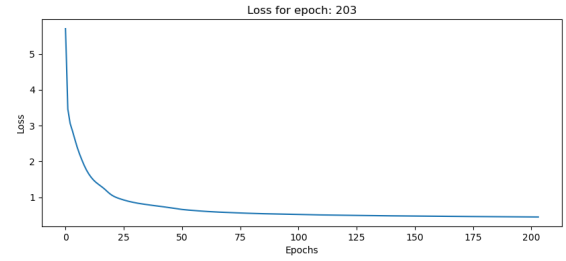
((a)) After 1 Iteration

((b)) After 10 Iterations

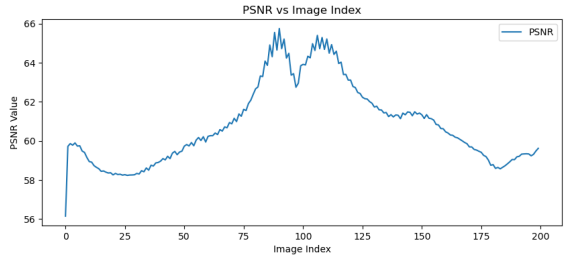
((c)) After 100 Iterations

((d)) After 200 Iterations

Fig. 4: Lego Dataset Training Trends



((a)) Epochs vs Loss



((b)) Original vs Rendered - PSNR

((c)) Original vs Rendered - SSIM

Fig. 5: Lego Training Results

III. Conclusion

1) Successful Implementation and Refinement of NeRF: The successful implementation and refinement of NeRF has illuminated its potential for crafting photorealistic scenes from a sparse collection of images. This accomplishment stands as a testament to the technique's utility in synthesizing high-fidelity views.

2) Extensive Analysis Across Varied Datasets: By extending our analysis across three varied datasets, our project not only showcased the model's robustness but also its adaptability, reinforcing the versatility of our approach.

3) Establishment of a Testing Pipeline: The establishment of a testing pipeline marked a significant stride towards quantitatively assessing the model's ability to generate images from novel viewpoints. Metrics such as PSNR and SSIM provided a lens through which the quality of these images was measured, offering empirical evidence of the model's performance.

4) Critical Roles of Proper Initialization and Advanced Image Processing: Throughout this project, the critical roles of proper initialization and advanced image processing techniques were underscored. These elements were pivotal in enhancing the model's output, contributing to our understanding of efficient and effective 3D scene reconstruction.

5) Conclusion: In weaving together these threads of innovation and discovery, our project not only affirms the capabilities of NeRF in achieving photorealistic renderings from limited data but also enriches the tapestry of computer vision research, laying groundwork for future explorations in 3D scene synthesis.

References

[1] M. Tancik, "NeRF From Google Colab," 2020. [Online]. Available: https://colab.research.google.com/github/bmild/nerf/blob/master/tiny_nerf.ipynb. [Accessed: Date].

[2] M. Tancik, "NeRF: Representing Scenes as Neural Radiance Fields," 2020. [Online]. Available: https://www.matthewtancik.com/nerf. [Accessed: Date].

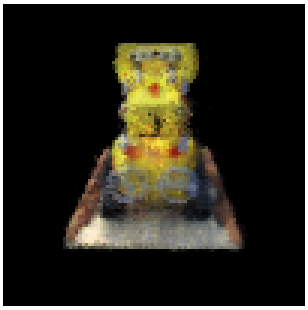
[3] "RBE549 - Final Project: NeRF," Worcester Polytechnic Institute, Spring 2023. [Online]. Available: https://rbe549.github.io/spring2023/proj/p2/. [Accessed: Date].

[4] B. Mildenhall, P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields," *arXiv preprint arXiv:2003.08934*, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934. [Accessed: Date].

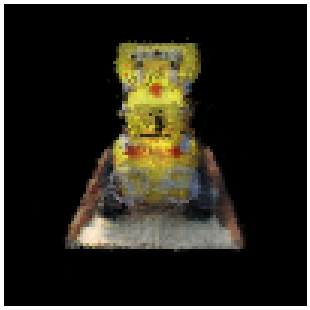
[5] A. Prasad, "It's NeRF from Nothing: Build a Vanilla NeRF with PyTorch," Towards Data Science, 2020. [Online]. Available: https://towardsdatascience.com/its-nerf-from-nothing-build-a-vanilla-nerf-with-pytorch-7846e4c45666. [Accessed: Date].

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields," in *European Conference on Computer Vision (ECCV)*, 2020. [Online]. Available: https://cseweb.ucsd.edu/~viscomp/projects/LF/papers/ECCV20/nerf/. [Accessed: Date].

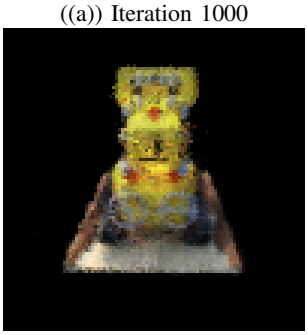
[7] "NVIDIA Research. Instant NeRF: Rapid Neural Rendering with In-Situ Scene Representation," 2021. [Online]. Available: https://github.com/NVlabs/instant-ngp/. [Accessed: Date].



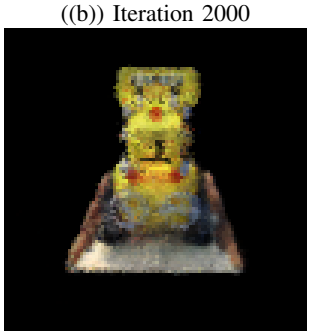
((a)) Iteration 1000



((b)) Iteration 2000

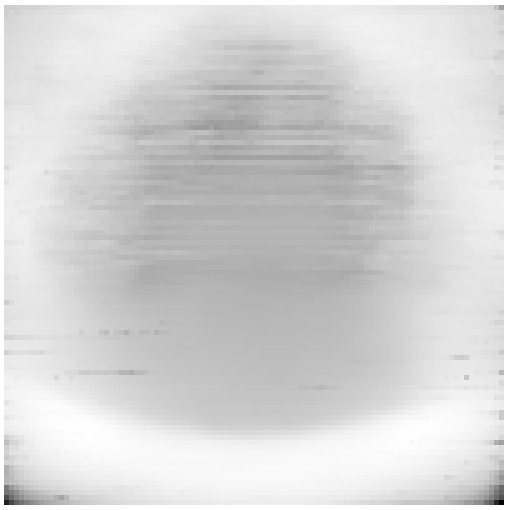


((c)) Iteration 3000

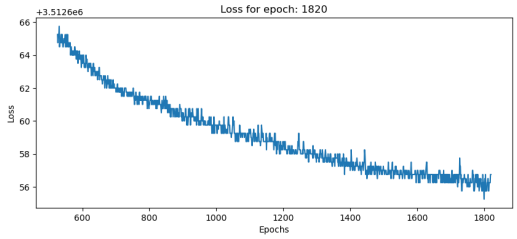


((d)) Iteration 4000

Fig. 6: After achieving a satisfactory level of accuracy, further iterations did not result in improved performance. Instead, we observed the distortion of rendered rays, indicative of overfitting in the model.

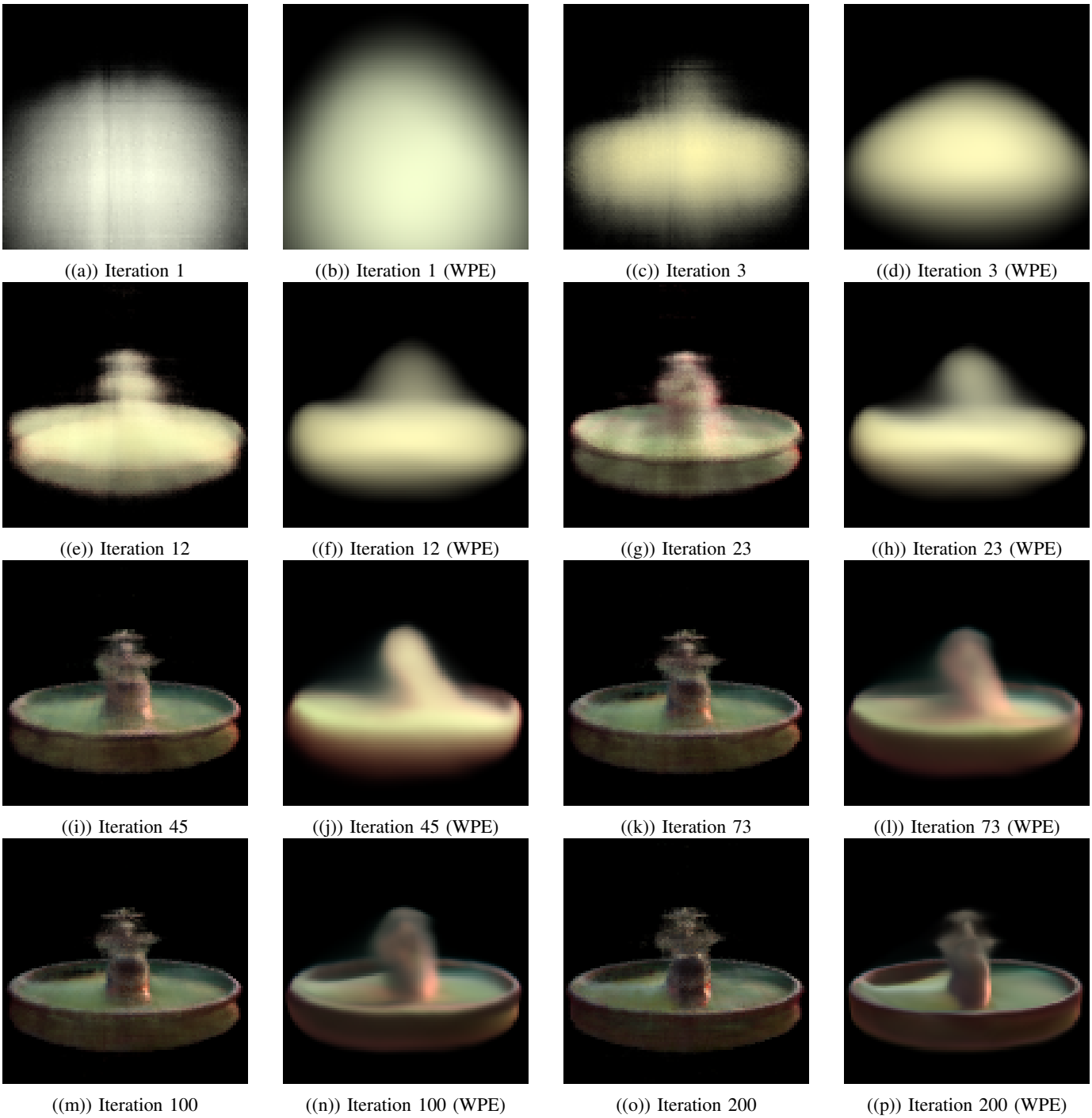


((a)) Training output at Iteration 1800



((b)) Epoch vs Loss

Fig. 7: Failed Training Model of Ship Dataset: We encountered a failed model during the training of the Ship dataset. Despite running nearly 2000 iterations, the results were minimal, if not negligible. Upon investigation, we identified the issue with the input image's data type (float64) and color range (0-255). Once these issues were acknowledged and addressed, we were able to achieve proper outcomes as observed above.



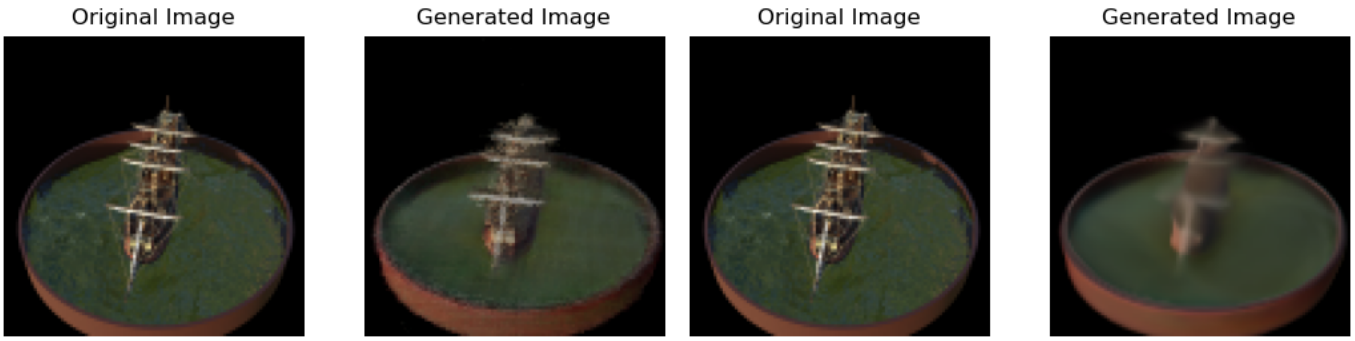
((a)) Iteration 1 ((b)) Iteration 1 (WPE) ((c)) Iteration 3 ((d)) Iteration 3 (WPE)

((e)) Iteration 12 ((f)) Iteration 12 (WPE) ((g)) Iteration 23 ((h)) Iteration 23 (WPE)

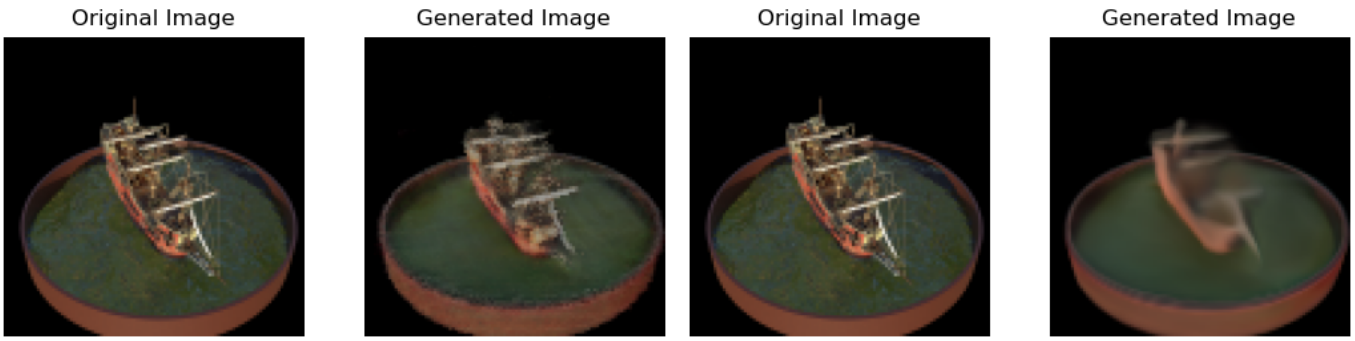
((i)) Iteration 45 ((j)) Iteration 45 (WPE) ((k)) Iteration 73 ((l)) Iteration 73 (WPE)

((m)) Iteration 100 ((n)) Iteration 100 (WPE) ((o)) Iteration 200 ((p)) Iteration 200 (WPE)

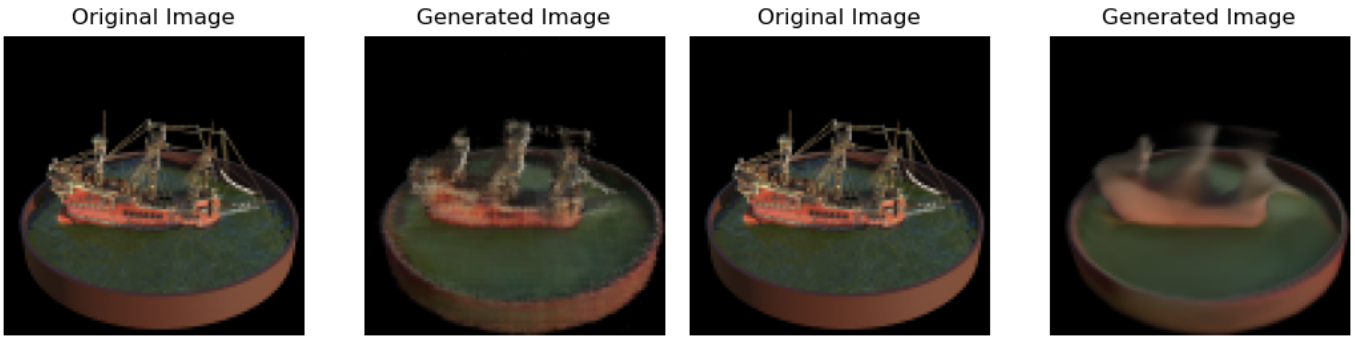
Fig. 8: Ship Dataset Training Trends - Positional Encoding vs Without Positional Encoding (WPE)



((a)) Pose 1 ((b)) Pose 1 (WPE)

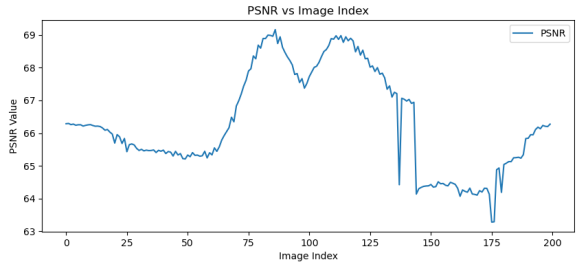


((c)) Pose 2 ((d)) Pose 2 (WPE)

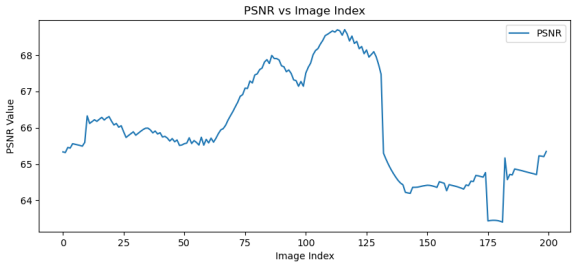


((e)) Pose 3 ((f)) Pose 3 (WPE)

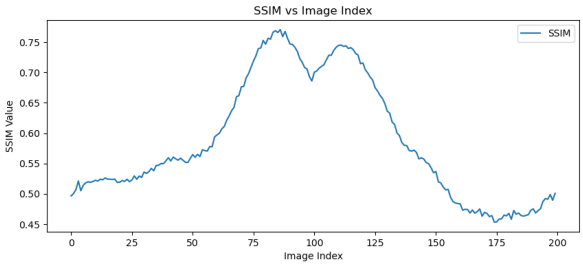
Fig. 9: Ship Testing Results - Original vs Generated - Positional Encoding vs Without Positional Encoding (WPE)



((a)) Ship - PSNR

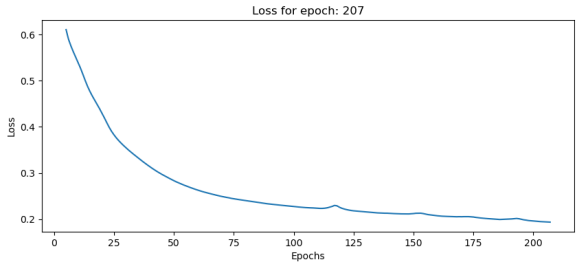


((b)) Ship (WPE) - PSNR

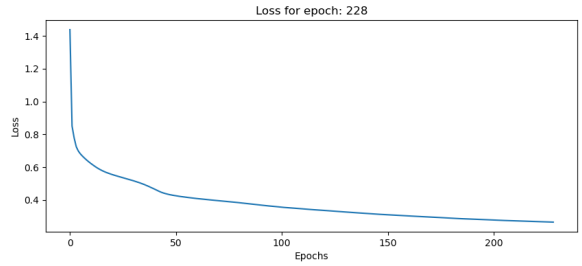


((c)) Ship - SSIM

((d)) Ship (WPE) - SSIM

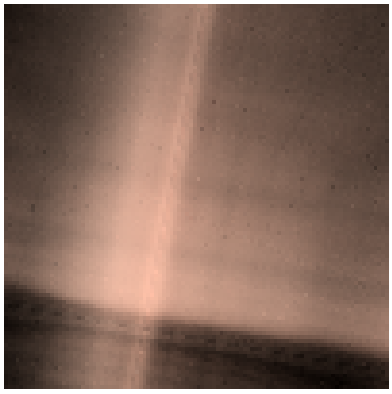


((e)) Ship - Epoch vs Loss

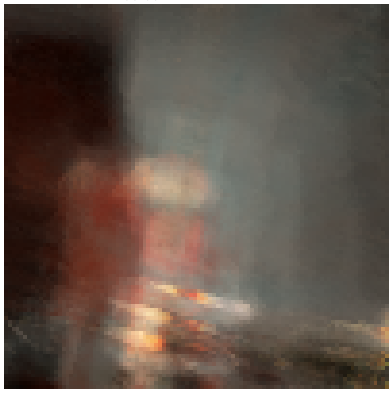


((f)) Ship (WPE) - Epoch vs Loss

Fig. 10: Ship Training Results - Postional Encoding vs Without Positional Encoding



((a)) Iteration 1



((b)) Iteration 200

((c)) Iteration 600

((d)) Iteration 1000

Fig. 11: The training trend of our own dataset seemed very promising with satisfying levels of reconstruction.

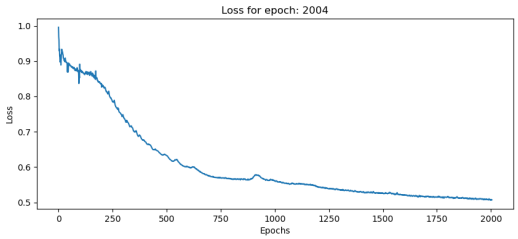
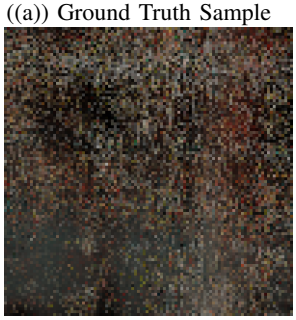


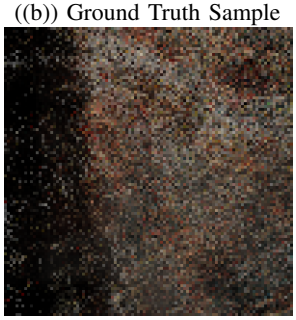
Fig. 12: Own Dataset - Training Results - Epochs vs Loss

((a)) Ground Truth Sample

((b)) Ground Truth Sample



((c)) Rendered Image



((d)) Rendered Image

Fig. 13: Own Dataset Test Results: While the model's reconstruction is currently faintly visible, further training may enhance its accuracy. However, the primary issue lies within the artifacts present in our dataset, such as lighting inconsistencies, background color variations, and blurriness. Improving the dataset quality is crucial for achieving better results.

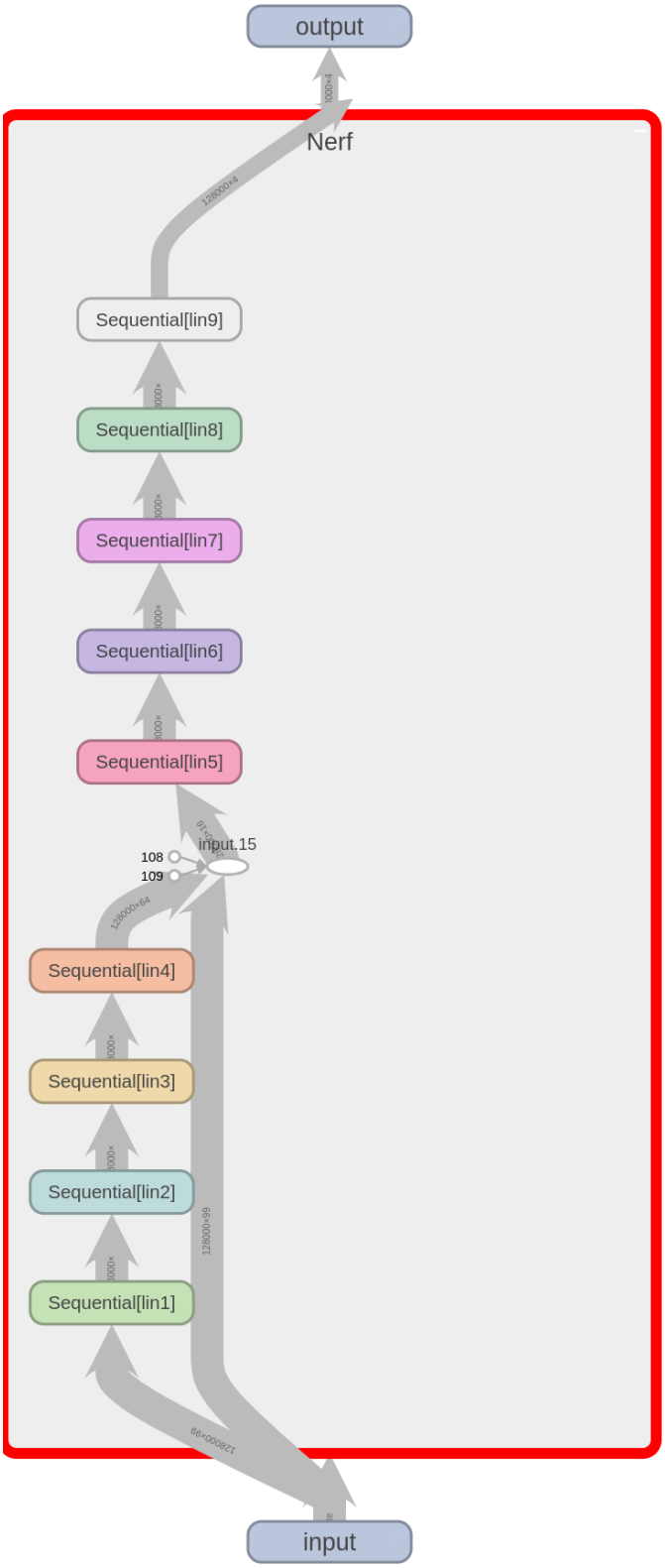


Fig. 14: Neural Network Architecture