EE 367 Project Proposal

Irmak Sivgin

I. Motivation and Project Overview

The aim of this project is exploring the use of neural radiance field networks (NeRFs) for 3D object segmentation. The proposed framework is as follows: Multiple images featuring a scene from different angles will be collected and transformed into binary images (object masks) using an open-source state-of-the-art image segmentation model. One candidate would be Meta AI's Segment Anything Model (SAM) [1]. The NeRF model will be trained on these binary images. Since the task does not have any dependance on rendering color, the traditional NeRF pipeline can be simplified. Specifically, mean squared error (MSE) loss can be calculated between the predicted transparency value and the ground truth binary image. Given the activation pattern from the previous layer, the last layer's weights have a closed form (least squares) solution that minimizes the MSE loss. This can be exploited to speed up the gradient descent based optimization procedure. Since the ground-truth image has binary entries, it is more natural to perform a thresholding operation on the predicted image before calculating the MSE loss, however, hard thresholding is not differentiable. Thus, the effect of using hard thresholding on the forward pass and a soft thresholding during error backpropagation will be explored. Also, similarity of this task to that of a classification task suggests cross-entropy loss can be useful. These two loss functions in their effectiveness of model training will be compared. Some details concerning the NeRF architecture are discussed next.

A. Encoding Strategies

Fourier feature encodings play a pivotal role in enhancing the model's ability to learn high-frequency functions. Positional encoding introduced in [2] has been particularly popular for use in NeRFs. Following the ideas in [3], we propose to compare the listed encoding functions:

- **Basic Encoding:** Implemented as $\gamma(v) = [\cos(2\pi v), \sin(2\pi v)]^\top$, this encoding modulates input coordinates around a unit circle.
- **Positional Encoding:** A more complex form is given by $\gamma(v) = [\ldots, \cos(2\pi\sigma^{j/m}v), \sin(2\pi\sigma^{j/m}v), \ldots]^\top$ for $j = 0, \ldots, m-1$, expanding the input space to higher dimensions and allowing the network to learn at multiple scales.
- **Gaussian Encoding:**

  $$\gamma(v) = [\cos(2\pi Bv), \sin(2\pi Bv)]^T$$

  where each entry in $B \in R^{m\times d}$ is sampled from $\mathcal{N}(0, \sigma^2)$. This encoding utilizes random frequency vectors for input coordinates, thus generalizing the positional encoding strategy.

B. Model Architecture

The model will be a 3-layer MLP with ReLU activations, mapping coordinate vectors ($v = (x, y, z)$ or $\gamma(v)$) to a transparency value. Although ReLU is a widely used nonlinearity for neural networks, following the work [4], we will also explore sine activations given by $f(x) = \sin(\omega_0(Wx + b))$, where $W \in R^{N\times M}$ and $b \in R^N$ represent a learnable affine transformation, and ω_0 is a tunable hyperparameter.

C. Analysis of Robustness

For the best performing model selected from the previously discussed encoding, nonlinearity and loss function alternatives will be tested for robustness. There will be two main cases:

- Applying a Gaussian blur to the original image before segmentation.
- Adding different levels of Gaussian noise to the original image before segmentation.

Both are expected to alter the quality of segmentation, making the NeRF task more challenging. The performance change of the model will be assessed.

II. Milestones and Timeline

There are multiple tasks within the proposed project, grouped into 3 main parts. One week will be dedicated to each.

Best Model Selection

- Starting with clean images of an object, extracting binary masks.
- Comparing the performance of 3-layer MLPs with ReLU and sine nonlinearities, paired with encoding strategies discussed above.

Optimization Framework

- Straightforward gradient descent with MSE loss.
- Using least-squares update for the outermost layer weight and bias to speed up optimization.
- Using hard/soft thresholding on the predictions before error computation and comparing image quality when this step is added.
- Changing the loss function to cross-entropy loss and comparing performances.

Robustness

After all steps above have been fixed, corrupting the original clean images before segmentation to observe its effect on the end result by:

- Addition of Gaussian noise with different variance levels.
- Convolution with a Gaussian blur kernel.

REFERENCES

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," 2020.

[4] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *CoRR*, vol. abs/2006.09661, 2020. [Online]. Available: https://arxiv.org/abs/2006.09661