DLCV HW4 REPORT

r12942046 薛仁豪

Problem

1. Essence, Key Components and Pros/Cons of NeRF

a. The NeRF idea in your own word

NeRF, which stands for Neural Radiance Fields, is an innovative approach for representing 3D scenes and objects. In detail, NeRF represents a scene or object as a function that maps 3D coordinates to radiance values. This function describes the appearance of the scene from any viewpoint.

To derive this function, NeRF leverages deep neural networks to model the intricacies of the 3D scene. The neural network takes 3D coordinates as input and produces corresponding radiance or color values at each point.

Moreover, NeRF is trained by optimizing its network using a dataset comprising images of the scene. During training, it minimizes the difference between the rendered images from the learned 3D representation and the actual input images. Afterwards, NeRF learns to generate images from arbitrary viewpoints within the scene.

b. Which part of NeRF do you think is the most important

I believe that the novel concept of encoding the information of a scene into a neural network is the most important part of NeRF. The encoding part involves training a neural network to learn a mapping from 3D spatial

coordinates to scene properties such as color and opacity. This network effectively encodes the information about the scene and captures the complex interactions of light with the scene geometry.

c. Compare NeRF's pros/cons w.r.t. other novel view synthesis work

The primary drawback of NeRF stems from its time-consuming processes during both the training and inference stages. In response to this challenge, InstantNGP and DVGO present distinct methods to address and mitigate these issues.

DVGO presents a novel acceleration method that directly models specific modalities of interest (such as density, color, or features) using a dense voxel grid. This voxel-grid representation proves to be efficient for querying any 3D positions through interpolation. Additionally, the model incorporates various supplementary techniques to ensure robust convergence and enhance overall performance. One notable strategy involves post-activation, where non-linear activation is applied after trilinear interpolation. This approach has the capability to generate sharp surfaces, or decision boundaries, using significantly fewer grid cells. Another enhancement involves initializing the dense voxel grid with near-zero opacity and adjusting learning rates for voxels with fewer visible views. This particular measure effectively prevents suboptimal solutions in the model.

In contrast to NeRF, which utilizes trigonometric functions with different frequencies for encoding coordinates and direction, ***InstantNGP*** adopts parametric encodings. These introduce additional trainable parameters into an auxiliary data structure, such as a grid or a tree. Moreover, InstantNGP stores trainable feature vectors in a compact spatial hash table with a tunable size, allowing for a trade-off between parameters and reconstruction quality. Unlike methods that depend on progressive pruning or prior scene knowledge, InstantNGP employs multiple hash tables at different resolutions. Despite having 20 times fewer parameters than dense grid encoding, it maintains comparable reconstruction quality. Additionally, InstantNGP's neural network autonomously resolves hash collisions, diminishing control flow divergence, streamlining implementation, and enhancing overall performance.

2. Implementation Details

The implementation is referenced from the following GitHub repository:

kwea123/nerf_pl: NeRF (Neural Radiance Fields) and NeRF in the Wild using pytorch-lightning (github.com)

Most of the architecture and hyper-parameters follow the settings in this repository, and the dataset is loaded using the code provided by TA. Although there are not many differences, two specific modifications significantly impact the performance. First, an increase in the number of fine samples has been implemented. Second, the standard deviation of noise has been adjusted from

1 to a substantially smaller value.

It is natural that the first modification can enhance the model. The increase of fine samples aids the model in obtaining a higher spatial resolution representation of the scene, facilitating a better understanding of complex structures and alleviating aliasing issues.

Regarding the second modification, the results appear rather weird. Given that our dataset comprises images without noise, introducing noise during volume rendering might be unnecessary to ensure the model's generalization ability. However, experimental results indicate that the optimal outcome is achieved when adding small, non-zero standard deviation noises. Hence, I think that incorporating randomness remains essential to prevent the model from converging to a suboptimal solution and to generate more diverse and accurate representations.

3. Performance

Setting	PSNR	SSIM	LPIPS (vgg)
A	**44.996**	**0.995**	**0.0996**
B	**44.758**	**0.995**	**0.0993**
C	**43.682**	**0.994**	**0.1005**

With all other hyper parameters fixed, settings B and C deviate from setting A in the embedding dimension and the depth of the fully connected network,

respectively. Specifically, in setting B, the embedding dimension is twice that of A. On the other hand, in setting C, the number of layers in the network increases from 4 to 8.

Based on the results above, it is evident that the performance does not show improvement following the adjustment. Consequently, we can conclude that the notion of achieving a better model by employing a deeper network or higher embedding is not substantiated

PSNR (Peak Signal-to-Noise Ratio)

A metric used to evaluate the quality of a reconstructed or compressed signal by comparing it to the original signal.

$$PNSR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$$

MAX is the maximum possible pixel value of the image, while MSE is the Mean Squared Error between the original and reconstructed signals. The higher the PSNR value, the better the quality of the reconstructed signal.

SSIM (Structural Similarity Index)

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

x and y are the two compared images. μ is the average luminance values, σ^2 is the variance and σ_{xy} is the covariance of x and y. SSIM ranges from -1 to 1,

where 1 indicates perfect similarity and -1 indicates complete dissimilarity. Compared to PSNR, it aligns more closely with human perception.

LPIPS (Learned Perceptual Image Patch Similarity)

The LPIPS metric is based on a deep convolutional neural network that has been trained on large datasets of images to predict perceptual similarity. The network is trained to assign similarity scores to image patches, and these scores are then aggregated to compute an overall similarity score for the entire image pair. In our case, VGG is used to compute the similarity.

4. Depth Maps