DLCV HW4 REPORT

R11521701 程懷恩

Problem 1：3D Novel View Synthesis

1. Model comprehension

A. The NeRF idea in your own words

The core idea is to represent a scene as a continuous 5D function, mapping every 3D point and viewing direction to a color and density. When you train a neural network with this function, using multiple 2D images of a scene from different angles, the network learns to predict the color and opacity of light passing through every point in space from any direction.

NeRF can render photorealistic 2D images of the scene from new, unobserved viewpoints while integrating predicted camera rays and images.

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

NeRF treats a scene like a continuous mathematical function with flowing space where every single point can be described in terms of color and transparency. This is used to break down a scene into lots of tiny cubes (voxels) or dots (point clouds).

The two-stage process of NeRF enables model to build a detailed 3D representation of the scene by first broadly identifying areas of interest and then honing in on those areas for finer detail.

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

Pros:

1. High Quality and Detail: NeRF can create highly detailed and photorealistic renderings, capturing intricate details like lighting, shadows, and textures.
2. Continuous Representation: Unlike discrete representations, NeRF's continuous function enables smooth transitions and renderings from any viewpoint.

Cons:

1. High Computational Cost: NeRF requires significant computational resources and time for training and rendering, making it less practical

for real-time applications. Therefore, without any optimization, Nerf can not be implemented in real-time.

2. Limited to Static Scenes: NeRF mainly works with static scenes but fails with dynamic elements or changing lighting conditions.

2. **Describe the implementation details of your NeRF model for the given dataset. You need to explain your ideas completely.**

According to the official settings of NeRF, the sampling size was set to 64 per ray in the first stage and 192 in the second stage, totaling 256. In the first stage, equidistant sampling is performed, and RGB and opacity are predicted simultaneously. In the second stage, more sampling points are placed in regions with higher density to predict RGB and opacity. After obtaining predictions for RGB and opacity in each of the two stages, they are plugged into the formula to calculate the MSE loss and return it.

In the MLP part of the model, I followed the official configuration, setting the MLP to have eight layers with 256 channels each. Then, an additional layer outputs the volume density σ (which is rectified using a ReLU to ensure non-negativity) and a 256-dimensional feature vector. This feature vector is concatenated with the positional encoding of the input viewing direction ($\gamma(d)$). Finally, it passes through a 128-dimensional fully connected layer to output the RGB predictions.

3. **Scenario comparison**

Firstly, I tried to modify the number of mlp layers and dimensions. However, after several attempts, I found that my model was prone to overfitting, causing the PSNR to drop drastically to 2. Therefore, I decided to make adjustments to the positional encoding part. I experimented with changing the channel numbers to see if the differences matched the results mentioned in the paper.

1. PSNR (Peak Signal-to-Noise Ratio) is used to assess the similarity between an image and its original counterpart, particularly in the areas of image compression and reconstruction. A higher PSNR value indicates a greater similarity between the two images, reflecting higher quality.

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

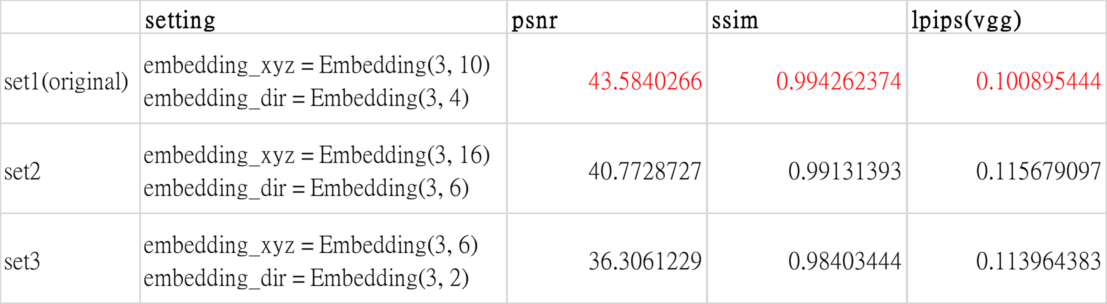
2. SSIM (Structural Similarity Index) is a metric used to measure the similarity between two images. It takes not only brightness and contrast but also structural information into account. SSIM is commonly used in image quality assessment, image compression, and image enhancement, among other fields. The values of SSIM typically range from -1 to 1, with a value closer to 1 indicating a higher degree of similarity between the two images and thus better image quality.

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

μ : brightness mean
Sigma: brightness variance

3. LPIPS (Learned Perceptual Image Patch Similarity) with VGG as the feature network: A lower value indicates smaller perceived differences between the reconstructed and reference images, hence better perceptual quality.

Experiments (on validation set):



	setting	psnr	ssim	lpips(vgg)

set1(original)	embedding_xyz = Embedding(3, 10)
	embedding_dir = Embedding(3, 4)
		43.5840266	0.994262374	0.100895444

set2	embedding_xyz = Embedding(3, 16)
	embedding_dir = Embedding(3, 6)
		40.7728727	0.99131393	0.115679097

set3	embedding_xyz = Embedding(3, 6)
	embedding_dir = Embedding(3, 2)
		36.3061229	0.98403444	0.113964383

Note: The red line means the best performance.

According to the NeRF paper, the number of frequencies controls the magnitude of learning variations. If the number of frequencies is set too high, it can easily lead to overfitting and result in noise (Fig 1.). Conversely, if it's set too low, high-frequency components may not be reproduced, leading to a loss of fine image details (Fig 2.).

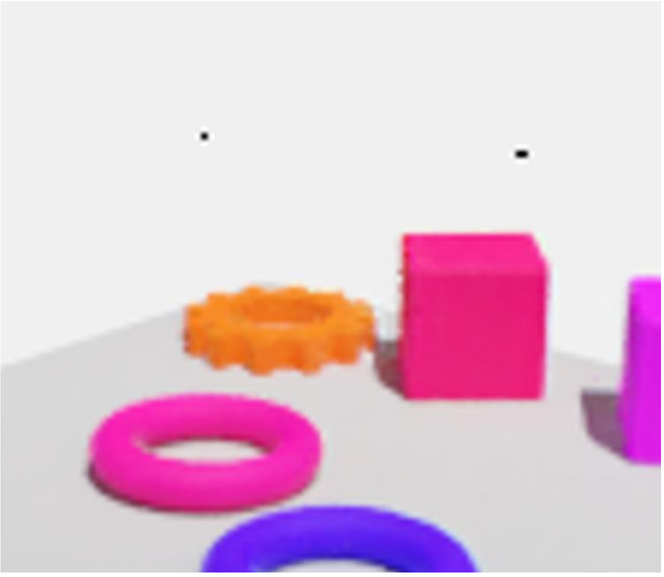


Fig 1. Picture with noises in setting 2

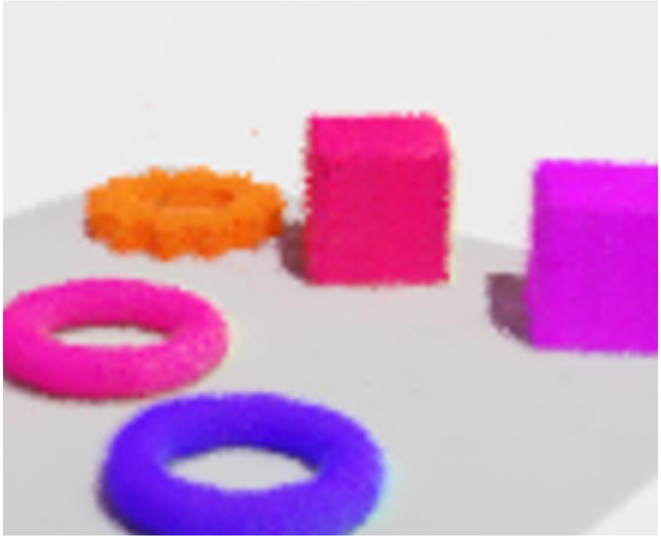
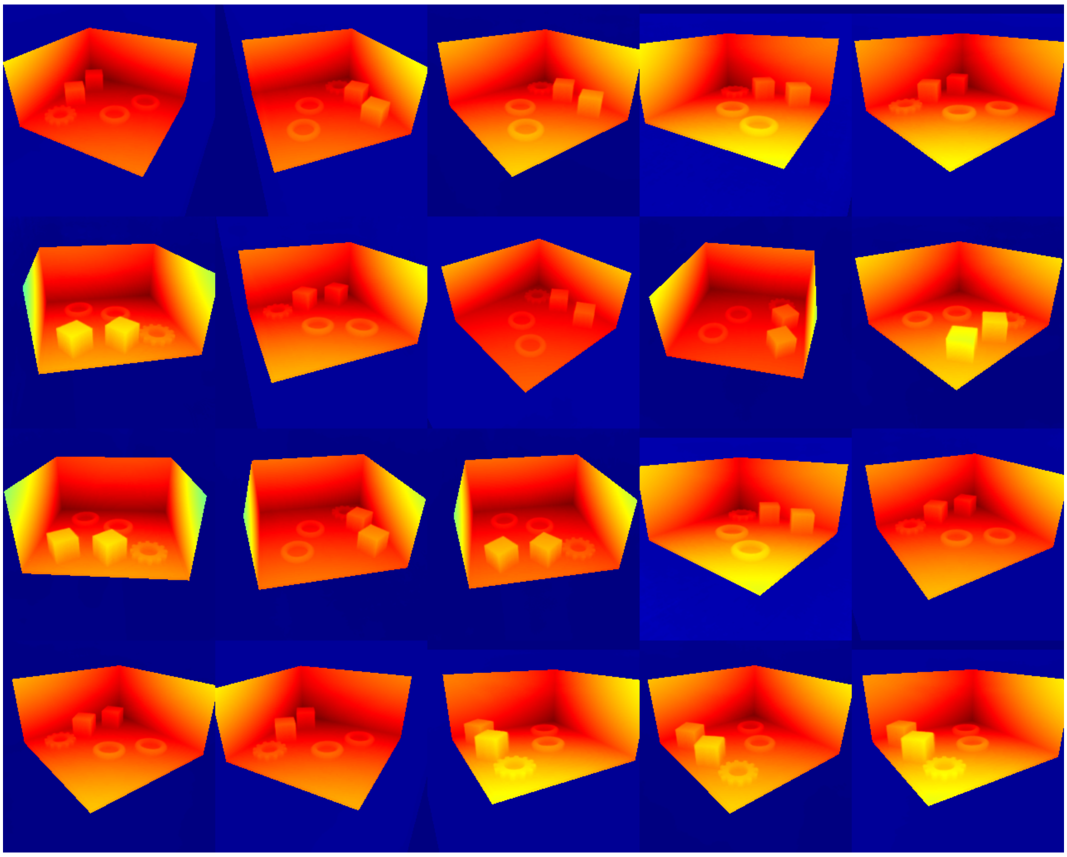


Fig 2. Loss of fine image details in setting 3

4. **With your trained NeRF, please implement depth rendering in your own way and visualize your results.**



Reference:

1. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric
   https://arxiv.org/abs/1801.03924
2. 【影像評估指標】PSNR LPIPS LMD SSIM FID
   https://zhuanlan.zhihu.com/p/658827245
3. 【AI 講壇】程式碼導讀 - NeRF [Neural radiance fields]
   https://www.youtube.com/live/SoEehTR2MiM?si=p_HkVkY70NSgwIBt
4. NeRF_pl: https://github.com/kwea123/nerf_pl
5. LPIPS: https://github.com/richzhang/PerceptualSimilarity
6. arXiv:1801.03924 [cs.CV] https://doi.org/10.48550/arXiv.1801.03924

Collaborator:

Thanks to R11521613, R11942180, Lin, and Huang for the discussion, which helped complete the homework.