DLCV

Homework 2

R12922147 魏子翔

Problem 1.

1. Please explain:
   a. the NeRF idea in your own words

      NeRF (Neural Radiance Fields) is a technique that enables the generation of highly realistic and detailed 3D scenes from a collection of 2D images. It takes 5-d data as input (3-d for Spatial location, 2-d for Viewing direction) and gives 4-d data (3-d for Output color and 1-d for Output density) as output. NeRF can generate novel views of the scene from previously unseen viewpoints by casting rays through the scene and using the learned function to predict the appearance of objects along those rays.

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

      Since the model mainly rely on the MLP layers to generate the different views, I think the MLP layers are the most important part in NeRF. The concatenated inputs go through an MLP that comprises several fully connected layers. This MLP learns to approximate the underlying mapping from the input coordinates and viewing directions to the corresponding radiance (color) and density values.

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

      Pros:

      Capable of capturing fine details, resulting in highly realistic renderings.

      Represents scenes as a continuous function, enabling smooth and accurate rendering from novel viewpoints.

      Does not rely on predefined meshes or geometric structures, making it adaptable to various scene complexities.

      Cons:

      Training and rendering can be computationally intensive.

      Challenging to handle large-scale scenes or dynamic objects efficiently.

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

   I implemented NeRF following the github repo, which first maps the input coordinate and camera pose into their corresponding embedding space. Rather than using 2-dimension to represent the camera pose, this implementation used 3-dimension as the direction input. After passing the positional encoding process, the model sent the data into the 8-layered mlp, which is the main part of the model, to generate the output rgb and density data.

3. Given novel view camera pose from metadata.json, your model should render novel view images. Please evaluate your generated images and ground truth images with the following three metrics (mentioned in the NeRF paper). Try to use at least three different hyperparameter settings and discuss/analyze the results.

   (1) PSNR (Peak Signal to Noise Ratio):

   PSNR measures the quality of a reconstructed or compressed image by comparing it to the original, uncompressed image. It quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The formula of PSNR is $10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$, where MAX is the maximum possible pixel value of an image, and MSE represents the average squared difference between the original and reconstructed image pixels.

   (2) SSIM (Structural Similarity Index Measure):

   SSIM quantifies the similarity between two images by considering luminance, contrast, and structure. It measures the perceived change in structural information, capturing the visual similarity more accurately than PSNR. SSIM produces values between -1 and 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect anti-correlation.

   (3) LPIPS (Learned Perceptual Image Patch Similarity)

   LPIPS measures image similarity based on deep neural network embeddings. Unlike PSNR and SSIM, LPIPS uses a learned model that has been trained on a dataset of images to predict perceptual similarity. LPIPS tries to emulate human judgment on image similarity, considering factors that affect human perception such as color, texture, and structure. It computes distances between deep features extracted from images using a neural network, and lower LPIPS scores indicate higher perceptual similarity.

   All of my hyperparameters follow the default setting of training LLFF in the nerf_pl github, only num_epoch=20.

   Also, setting A is with 8 layers of MLP, setting B is with 6 layers, and setting C is with 10 layers of MLP.

	PSNR	SSIM	LPIPS (vgg)
A	44.58	0.9947	0.1002
B	44.56	0.9946	0.0979
C	42.69	0.9932	0.1049

   The LPIPS(vgg) score is much higher than the score in NeRF paper. In my opnion, it's because the model will output the container box of the object in the picture(see p4). Although it's not obvious for human eyes, but the vgg model could be sensitive to the difference.

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