Deep Learning for Computer Vision

NTU, Fall 2023, homework4
電機所碩一 謝宗翰 r12921a10

- **Problem 1: 3D Novel View Synthesis**

1. Please explain:

   - The NeRF idea in your own words

     NeRF（Neural Radiance Fields）是一種基於深度學習的 3D 視圖合成技術。NeRF 通過 implicit 函數來表示場景。這個函數以空間坐標 x（3D vector）和觀看角度 d（2D vector）作為輸入，輸出該點的顏色 c（3D vector by rgb）和密度 σ（標量）。而此方法使得 NeRF 能夠根據不同的觀看角度，渲染出具有視角依賴的顏色和光線反射效果的場景。

   - Which part of NeRF do you think is the most important

     在 NeRF 中的多層感知機（MLP）應該算是是核心要角。MLP 會先接收 input 坐標 x，並吐出密度 σ 和 intermediate 向量。處理完後，另一個 MLP 會處理剛剛的 intermediate 向量和觀看角度 d，以產生視角依賴的顏色。這種方法顯著提高了訓練過程的效率和效果，也有比較好的結果。

   - Compare NeRF's pros/cons other novel view synthesis work

     - 優點：

       NeRF 訓練 dataset 不需要真實的 3D 模型，他可以從 2D 圖像重建 3D 場景。而輸出結果不僅限於低幾何複雜度的簡單形狀，能夠產生高度詳細的圖像。

     - 缺點：

       NeRF 需要較長的訓練時間，而且通常只適用於單一場景，並沒有明顯的普遍性。

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

架構說明-1

```python
class NeRF(nn.Module):
    def __init__(self, D=8, W=256, in_channels_xyz=63, in_channels_dir=27, skips=[4]):
        """
        D: number of layers for density (sigma) encoder
        W: number of hidden units in each layer
        in_channels_xyz: number of input channels for xyz (3+3*10*2=63 by default)
        in_channels_dir: number of input channels for direction (3+3*4*2=27 by default)
        skips: add skip connection in the Dth layer
        """
        super(NeRF, self).__init__()
        self.D = D
        self.W = W
        self.in_channels_xyz = in_channels_xyz
        self.in_channels_dir = in_channels_dir
        self.skips = skips

        # xyz encoding layers
        for i in range(D):
            if i == 0:
                layer = nn.Linear(in_channels_xyz, W)
            elif i in skips:
                layer = nn.Linear(W + in_channels_xyz, W)
            else:
                layer = nn.Linear(W, W)
            layer = nn.Sequential(layer, nn.ReLU(True))
            setattr(self, f"xyz_encoding_{i+1}", layer)
        self.xyz_encoding_final = nn.Linear(W, W)

        # direction encoding layers
        self.dir_encoding = nn.Sequential(
            nn.Linear(W + in_channels_dir, W // 2), nn.ReLU(True)
        )

        # output layers
        self.sigma = nn.Linear(W, 1)
        self.rgb = nn.Sequential(nn.Linear(W // 2, 3), nn.Sigmoid())

```

模型的結構基本上跟原作者的一模一樣，沒做太多的更動。

架構說明-2

```python
def render_rays(
    models,
    embeddings,
    rays,
    N_samples=64,
    use_disp=False,
    perturb=0,
    noise_std=1,
    N_importance=0,
    chunk=1024 * 32,
    white_back=False,
    test_time=False,
):
```

比較要注意的是原作者的 render_rays 這個函式，其中 input 的參數 noise_std 要設為 0 不然會對於 alphas 產生影響，導致模型有過多的雜訊而 train 不起來，ssim 的數值會卡在 30 左右。

```
noise = torch.randn(sigmas.shape, device=sigmas.device) * noise_std

        # compute alpha by the formula (3)
        alphas = 1 - torch.exp(
            -deltas * torch.relu(sigmas + noise)
        )  # (N_rays, N_samples_)
        alphas_shifted = torch.cat(
            [torch.ones_like(alphas[:, :1]), 1 - alphas + 1e-10], -1
        )  # [1, a1, a2, ...]
        weights = (
            alphas * torch.cumprod(alphas_shifted, -1)[:, :-1]
        )  # (N_rays, N_samples_)
        weights_sum = weights.sum(1)  # (N_rays), the accumulated opacity along the rays
        # equals "1 - (1-a1)(1-a2)...(1-an)" mathematically
```

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.
   - ◆ Please report the PSNR/SSIM/LPIPS on the validation set.
   - ◆ You also need to explain the meaning of these metrics.
   - ◆ Different configuration settings such as MLP and embedding size, etc

Setting	**PSNR**	**SSIM**	**LPIPS (vgg)**
Setting 1	42.93	0.9937	0.0996

Setting 2
`embedding_xyz = Embedding(3, 20)`
`embedding_dir = Embedding(3, 8)`
	43.17	0.9938	0.1007

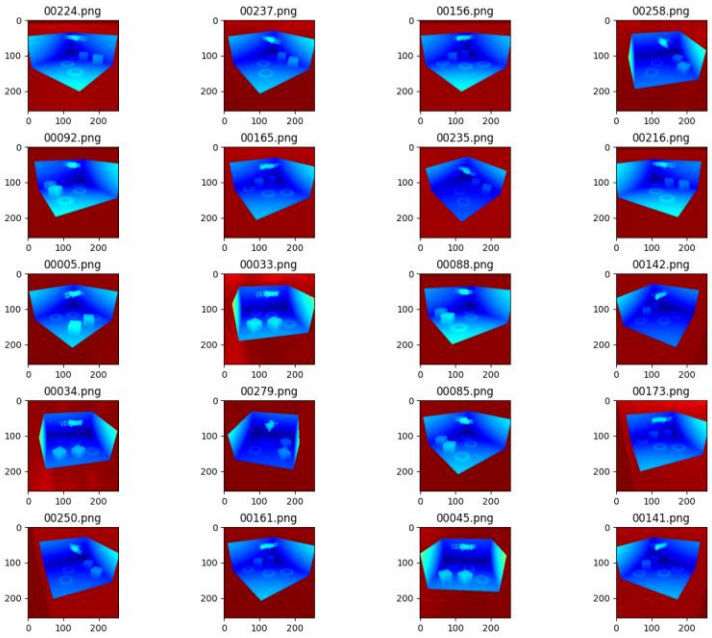
Setting 3
`self.embedding_xyz = Embedding(3, 5)`
`self.embedding_dir = Embedding(3, 2)`
	43.14	0.9938	0.1033

這三種設定數要是針對 embedding 層的不同大小去做設計，就結果來說差異並不大，但 Setting 2 跟 Setting3 都需要比較多的 epoch 才會有比較好的結果。			

分析方法	說明
PSNR (Peak Signal to Noise Ratio)	一種評價圖像品質的度量標準，衡量最大值信號和背景噪音之間的圖像質量參考值。PSNR 高於 40dB 說明圖像質量幾乎與原圖一樣好。
SSIM (structural similarity index)	一種用於量化兩幅圖像間的結構相似性的指標。SSIM 從亮度、對比度以及結構量化圖像的屬性，用均值估計亮度，方差估計對比度，

	協方差估計結構相似程度，SSIM 值的範圍為 0 至 1，越大代表圖像越相似
LPIPS (vgg) Learned Perceptual Image Patch Similarity	用於度量兩張圖像之間的差別，該度量標準學習生成圖像到 Ground Truth 的反向映射強制生成器學習從假圖像中重構真實圖像的反向映射，並優先處理它們之間的感知相似度，LPIPS 的值越低表示兩張圖像越相似。反之，則差異越大。

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



不知道為啥會有奇怪的半透明框框，原作中的範例不會有此現象，但是我問其他同學也會有這種情況發生。

- Reference

  https://zhuanlan.zhihu.com/p/309892873 矩陣說明

  Chatgpt4: help me to code