Report - DLCV HW4

Problem 1

1. Please explain:
    - a. the NeRF idea in your own words
        - NeRF is a method for rendering 3D scenes using a neural network, which is designed based on traditional rendering approaches in computer graphics. It represents a 3D scene as a function that maps positions in 3D space to the RGB colors and alpha values of the scene at those positions. This function is learned from training data and can be used to generate images of the scene from any viewpoint using volume rendering.
        - The neural network is trained to predict the radiance at each point in 3D space by reproducing input views of the scene and optimizing with gradient descent on the rendering loss. Once trained, the network can be used to generate images of the scene from any viewpoint by tracing rays through 3D space and evaluating the network at each point along the ray. This enables NeRF to produce high-quality images of the scene with realistic lighting and occlusion effects.
    - b. which part of NeRF do you think is the most important
        - NeRF is designed based on traditional rendering approaches in computer graphics, rather than 3D models in computer vision. The main focus is on producing realistic images, rather than accurately representing the 3D scene.
    - c. compare NeRF's pros/cons w.r.t. other novel view synthesis work
        - Neural Radiance Fields (NeRF) v.s. Generative Query Networks (GQN)
            - They are both methods for generating realistic images using neural networks. NRF generates images by modeling the distribution of images in a dataset using a neural network, while GQN generates images by using a neural network to generate a scene representation based on a set of images of that scene.
            - Pros of NeRF:
                - generate high-quality images that are realistic and diverse.
                - able to model complex image distributions, such as those found in natural images.
                - suitable for generating a wide range of images from different types of data
                - high-quality than GQN.
            - Cons of NeRF:
                - computationally expensive.
                - be challenging to implement.
                - needs more data than GQN.
            - Pros of GQN:
                - generate images of novel scenes that are not present in the training data.

        - generate images that are highly realistic and detailed, even when given only a few images of a scene.
        - excels at generating realistic images of novel scenes based on a small number of images.
        - needs less data than NeRF.
    - Cons of GQN:
        - limited to generating images of scenes that are similar to those present in the training data.
        - may not be able to generate images of complex or unusual scenes.
        - not high-quality than NeRF

   Please read through these two reference paper to get their ideas.

2. Describe the implementation details of Direct Voxel Grid Optimization(DVGO) for the given dataset. You need to explain DVGO's method in your own ways.
    - code:
        - The program first looks for the hotdog config file to determine how to process the data. It then reads the transform files and retrieves the RGB, rotation, and translation for each frame. After processing the images and poses, it follows a coarse-to-fine approach. For each point on a ray emitted from a pixel, trilinear interpolation is first performed on the two voxel grids to obtain the density and color features of the point, and then the point is trained using an MSE loss. Finally, we obtain the RGB for each pixel.
    - concept:
        - DVGO is a method for representing and rendering 3D scenes using voxel grids. It utilizes a coarse-to-fine approach, first identifying coarse 3D areas of interest before reconstructing fine details and view-dependent effects. During training, two coarse-grained voxels are trained using prior information and multi-view images, and the density field within them is used to determine free space in the scene. A tighter bounding box can then be obtained in the fine stage, and the grid can be defined within it to reduce the training of irrelevant variables. During volume rendering, it is also possible to skip over certain points on the ray by using the thick density field. The information of the entire scene is mainly stored in two voxel grids, and the information of a spatial point can be directly obtained using trilinear interpolation without an MLP.
        - DVGO represents the entire scene using two voxel grids and also utilizes volume rendering like NeRF. For each point on a ray emitted from a pixel, trilinear interpolation is first performed on the two voxel grids to obtain the density and color features of the point. The color and density of the point are then obtained through the corresponding decoding process. This represents the entire scene in a voxel grid and its corresponding decoder.
        - Coarse-to-fine allows for the skipping of sampling a large number of irrelevant points. The information of the entire scene is mainly stored in two voxel grids. Compared to NeRF, the information of a spatial point can be directly obtained using trilinear interpolation without an MLP.

3. Given novel view camera pose from transforms_val.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 two 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.
    - The Structural Similarity Index (SSIM)
        - a measure of the similarity between two images. It takes into account the perception of the human visual system, making it sensitive to local structural changes in the images. SSIM compares the brightness, contrast, and structure of the images, using the mean to estimate brightness, variance to estimate contrast, and covariance to estimate structural similarity. The SSIM value ranges from 0 to 1, with a larger value indicating greater similarity. If the two images are exactly the same, the SSIM value will be 1. In contrast to the L2 loss function, SSIM incorporates the theory of structural similarity to quantify the similarity between images.
    - The Peak Signal to Noise Ratio (PSNR)
        - a metric used to evaluate the quality of an image. However, it has certain limitations and is only used as a reference value for comparing the maximum signal to background noise in an image. The unit of PSNR is decibels (dB), and a larger value indicates less image distortion. In general, a PSNR higher than 40 dB indicates that the image quality is close to the original image; a value between 30-40 dB usually signifies that the loss of image quality due to distortion is within an acceptable range; a value between 20-30 dB indicates relatively poor image quality; and a PSNR less than 20 dB signifies significant image distortion.
    - The Learned Perceptual Image Patch Similarity (LPIPS)
        - is used to measure the difference between two images by learning the reverse mapping of generated images to the ground truth. It forces the generator to learn the reverse mapping of reconstructing real images from fake images, prioritizing the perceptual similarity between them. LPIPS is better suited to human perception than traditional methods. The lower the LPIPS value, the more similar the two images are, and vice versa, the greater the difference.
- Different configuration settings such as iteration number/number of voxel/stepsize ... lead to different performance.

Setting	PSNR	SSIM	LPIPS
DEFAULT: coarse_train: {N_iters=5000, N_rand=8192, lrate_density=1e-1, lrate_k0=1e-1, lrate_rgbnet=1e-3}, fine_train: {N_iters=20000, lrate_density=1e-1, lrate_k0=1e-1, lrate_rgbnet=1e-3}, coarse_model_and_render: {num_voxels=1024000, num_voxels_base=1024000}, fine_model_and_render: {num_voxels=160^3, num_voxels_base=160^3}	35.2001238822937	0.9742709962363135	0.041543695740401745

Setting	PSNR	SSIM	LPIPS
coarse_train: {N_iters=5000, N_rand=8192, lrate_density=1e-1, lrate_k0=1e-1, lrate_rgbnet=1e-3}, fine_train: {N_iters=40000, lrate_density=3e-1, lrate_k0=3e-1, lrate_rgbnet=3e-3}, coarse_model_and_render: {num_voxels=1024000, num_voxels_base=1024000}, fine_model_and_render: {num_voxels=160^3, num_voxels_base=160^3}	35.32416768074036	0.9746224998212297	0.04122317023575306
coarse_train: {N_iters=10000, N_rand=16384, lrate_density=2e-1, lrate_k0=2e-1, lrate_rgbnet=2e-3}, fine_train: {N_iters=40000, lrate_density=2e-1, lrate_k0=2e-1, lrate_rgbnet=2e-3}, coarse_model_and_render: {num_voxels=2048000, num_voxels_base=2048000}, fine_model_and_render: {num_voxels=160^3x2, num_voxels_base=160^3x2}	35.33480324745178	0.9751114421158853	0.03941033978015184
coarse_train: {N_iters=40000, N_rand=16384, lrate_density=2e-1, lrate_k0=2e-1, lrate_rgbnet=2e-3}, fine_train: {N_iters=200000, lrate_density=2e-1, lrate_k0=2e-1, lrate_rgbnet=2e-3}, coarse_model_and_render: {num_voxels=4096000, num_voxels_base=4096000, stepsize=0.3}, fine_model_and_render: {num_voxels=160^3x2, num_voxels_base=160^3x2}	35.25471591949463	0.9752839798683746	0.03891087893396616

We can easily find that more iterations benefit the performance before overfitting. Similarly, in common, the more the voxels become, the better the model performs. However, since the data isn't more enough, training too complicated model with many iterations will cause over-fitting.

Problem 2

1. Describe the implementation details of your SSL method for pre-training the ResNet50 backbone. (Include but not limited to the name of the SSL method you used, data augmentation for SSL, learning rate schedule, optimizer, and batch size setting for this pre-training phase)
   - You have to pre-train the backbone on the Mini-ImageNet without loading default pretrained weights.

- We recommend you to reference the following Github. (It's easy to understand, use, and train with relatively small batch size.)
  - BYOL - LINK (Recommended), Barlow Twins - LINK, etc.
- Since the pre-training phase may take weeks/months to finish under the general SSL setting, we fix the following training setting to reduce the training time. (You MUST follow this setting in the pre-training and fine-tuning phase for fair comparison.)
  - Image size: 128*128
  - Backbone: ResNet50 (You can choose whether to use the FC layer of ResNet50)
- implementation details
  - SSL method: BYOL
  - learning rate: 3e-4
  - weight decay: 1e-9
  - batch size: 320
  - optimizer: AdamW
  - scheduler: reduce on loss
  - epoch: 1000
  - early stopping if loss doesn't descend in a while

2. Please conduct the Image classification on Office-Home dataset as the downstream task. Also, please complete the following Table, which contains different image classification setting, and discuss/analyze the results.
   - Please report the mean classification accuracy on validation set.
   - TAs will run your code to verify the performance of the setting C in the Table below.
   - The architecture of the classifier needs to be consistent across all settings.

Setting	**Pre-training (Mini-ImageNet)**	**Fine-tuning (Office-Home dataset)**	**Validation accuracy (Office-Home dataset)**
A	-	Train full model (backbone + classifier)	0.26354679802955666
B	w/ label (TAs have provided this backbone)	Train full model (backbone + classifier)	0.4064039408866995
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone + classifier)	0.47783251231527096
D	w/ label (TAs have provided this backbone)	Fix the backbone. Train classifier only	0.21674876847290642
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	0.2684729064039409

C > B > E > A > D

According to the performance in the table, it implies that mine pre-trained backbone is precise enough and is more accurate than TA's.

We can easily find that it's better to train full model than fix the backbone and train classifier only. I think the reason is that the classifier I design isn't complicated enough. Furthermore, each dataset has its own specific way and focus points to extract the features. Thus, training the backbone will perform better.

Also, it's clear that better outcome comes with pre-training with the method of self-supervised learning, and it indicates that SSL is helpful.