# DLCV HW4

## Problem 1

### Please explain:

#### a. the NeRF idea in your own words

They use MLP which input are spacial location and view direction to represent a 3D view. The MLP output are volumn density and view-dependent color. Then they use classical volumn rendering to synthesis the 3D view by accumulating color considering density. Also, they use position encoding to improve the model performance.

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

I think the most important part is to use MLP as whole model. They consider the physic feature of light so we only need spatial location and view direction as our input. This not only reduce the model overhead but also improve model performance.

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

**pros :** NeRF can synthesize more complex shape.
**cons :** NeRF is hard to do real-time synthesis.

### 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.

DVGO is an optimized version of NeRF. It reduce the training time of NeRF. DVGO use density voxel to represent density and feat voxel to represent color and use these two information to synthsis 3D image. They also use post-activation to get sharper surface. In training, DVGO use course to fine training method. In course step, they use prior and multi-view image to train two voxel. After that, we can get shape information and use this in fine step to avoid too many meaningless voxel. In fine step, they also train two voxel but with high total numbers of voxel. They also adopt several trick to accelerate training process such as progressive scaling, known free space and unknown space, fine voxels allocation, free space skipping, fine-stage points sampling and training objective for fine representation. In the end, experiment show that DVGO can accelerate training process and also maintain acceptable performance.

### 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. Try to use at least two different hyperparameter settings and discuss/analyze the results.

**SETTING A :** number of training iteration multiply by 2 in fine and course step
**SETTING B :** number of voxel and voxel base multiply by 2 in fine step
**PNSR :** signal to noise ratio between two image, which can calculate by MSE
**SSIM :** structural similarity between two image, which is calculate by lightness, constrast, structure
**PNSR :** perceptual similarity between two image, which is calculate by feed image to model

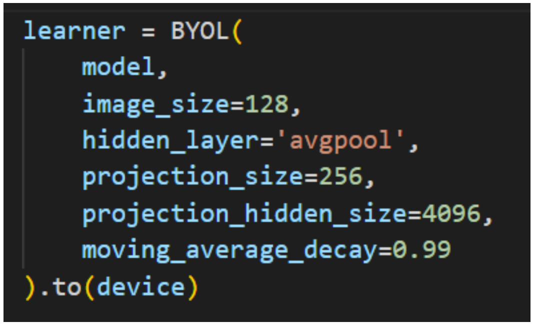
| setting | PSNR | SSIM | LPIPS vgg/alex |
|---|---|---|---|
| A | 35.2165 | 0.9746 | 0.0409/0.0219 |
| B | 35.3098 | 0.9754 | 0.0384/0.0186 |
| default | 35.1588 | 0.9741 | 0.0416/0.0226 |

As we know set the iteration more can let model train better and set voxel number more can let model synthesis more complicated. And, the result prove that

## Problem 2

### Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.

**model setting**
**seed :** 123
**optimizer :** Adam
**learning rate :** 1e-4
**loss function :** cross entropy loss
**epoch :** 100
**batch size :** 64
**BYOL setting**



```
learner = BYOL(
    model,
    image_size=128,
    hidden_layer='avgpool',
    projection_size=256,
    projection_hidden_size=4096,
    moving_average_decay=0.99
).to(device)
```

### 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.

| setting | valid acc |
|---|---|
| A | 30.54 |
| B | 36.70 |
| C | 45.07 |
| D | 32.02 |
| E | 13.05 |

Compared fixed backbone or not, we can see that fixed backbone will degrade model performance. I think that is because backbone capture some importtant data domain feature. So, we should train all to fine-tune that.

Compared supervised and self-supervised, we can see that self-supervised has better performance. I think that is because self-supervised learn some deep feature of image and label information may affect supervised a lot.