DLCV HW4 Report

R11921008 羅恩至

Problem1

1. (5%) Please explain:
a. the NeRF idea in your own words
b. which part of NeRF do you think is the most important
c. compare NeRF's pros/cons w.r.t. other novel view synthesis work

a.
NeRF (Neural Radiance Fields) is a method that generates novel views of 3D scenes of one object from many different angles or different sides of views of the object's 2D images as input. The input data contains 5 dimensions (coordinate + direction), the coordinate is first fed into the MLP (multilayer perceptron) to get the interlevel features, then get the final rendering output by the interlevel features and the directions.

b.
I think the most important part of NeRF is the implementation skills of volume rendering. Rather than simply implement average sampling, NeRF optimize two network (coarse and fine network) to do hierarchical volume sampling. Using coarse model to predict the light field density, then implement resampling (more samples in light field density), then output the final 3D scene from fine network.

c.
Compared with other novel view synthesis methods like SRN, NV or LLFF, the advantages of NeRF are: NeRF is able to render higher frequency detail features, and output 3D scenes with higher resolution. And also NeRF consumes less memory than other methods. However there still exist disadvantages of NeRF compare with some later published novel view synthesis methods including DVGO or FastNeRF. The drawbacks of NeRF are: the network in NeRF must be retrained for every scene, the rendering speed is very slow (need to do many sampling) and these cause longer training time. The other is that scene of NeRF is static so it does not perform well on scenes with various depths.

2. (10%) 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.

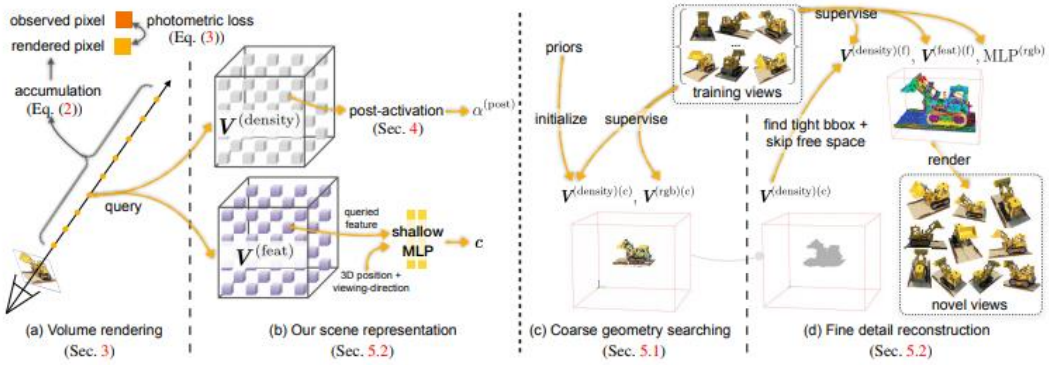

Fig 1. The implementation overview of DVGO

Direction Voxel Grid (DVGO) is a method to render 3D novel scenes from 2D images by voxel girds. DVGO treats the whole scene into 2 voxel grids, and get the density features color features of all sampling points on the ray by trilinear interpolation. In training steps, DVGO divides the training steps into coarse stage and fine stage. When training coarse model, using 2D images that contain different views to train the 2 voxels, and get the density field to determine which part belongs to the object or irrevelant background. And then train the fine model to get more detailed features and to successfully construct a 3D scene with finer features.

3. (15%) 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.

In this problem I had tried three kinds of hyperparameter settings to see the result. The three kinds of settings are set as the following mention:
Setting A (Default hyperparameters from DVGO github repo):
(Coarse model) lrate_density = 0.1, lrate_k0 = 0.1, N_iters = 5000, num_voxels = 1024000, (Fine model) N_iters = 20000, num_voxels = 160**3
Setting B (increase number of voxels and number of iterations):
(Coarse model) lrate_density = 0.12, lrate_k0 = 0.12, N_iters = 10000, num_voxels = 1200000, (Fine model) N_iters = 25000, num_voxels = 180**3
Setting C (decrease number of voxels):

(Coarse model) lrate_density = 0.12, lrate_k0 = 0.12, N_iters = 5000, num_voxels = 1200, (Fine model) N_iters = 25000, num_voxels = 1000

Setting	PSNR	SSIM	LPIPS
A	35.178	0.97414	0.0415
B	35.355	0.97499	0.0399
C	23.028	0.88103	0.1993

Table 1. Performance of different settings

According to the table above, from Setting A and B, it shows that when increasing the number of iteration and the number of voxels, the performance will slightly improved(PSNR and SSIM increases, LPIPS decreased). And from Setting B and C, if we largely reduce the number of voxels, the performance will significantly decreases. In conclusion, increase the number of iterations or the number of voxels would increase the performance.

Problem 2

1. (10%) 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)

In this problem, I tried BYOL as my SSL method. In pretraining steps, I did not implement lots of data augmentation with just simply resized the image into the size 128*128 and do normalization on the image. Then during pretraining, I set batch size into 64, and chose Adam as the optimizer with learning rate set to 3e-4. I had trained the backbone for about 245 epochs, without learning rate scheduler. The loss finally converge to about 0.2 and then started to oscillate between 0.2 and 0.3. Training backbone really takes time, which causing me about 2~3 days to fininsh training the backbone.

2. (20%) 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.

Below are the model hyperparameters that used for the comparison for different image classification seetings.

Batch size	128
Optimizer	SGD with momentum 0.9
Learning rate	0.05
Epoch	40

Table 2. Hyperparameter settings

Result:

Setting	**Pretraining (Mini-ImageNet)**	**Finetuning (Office-Home dataset)**	**Validation Accuracy (Office-Home dataset)**
A	-	Train full model	0.39655
B	w/ label	Train full model	0.47044
C	wo/ label	Train full model	0.47536 / 0.56158 (200epoch)
D	w/ label	Fix the backbone	0.24138
E	wo/ label	Fix the backbone	0.41133

Table 3. Training results of all 5 settings

For all 5 settings, I fixed the batch size to 128 and trained for 40 epochs in all settings. Using SGD as optimizer with learning rate 0.05 and momentum 0.9, and implementing 3 kinds of AutoAugment methods (Imagenet, SVHN, Cifar) to further strengthen the training sets. From the result table above, it shows that with same number of epochs, the accuracy ranks of validation set is Setting C > Setting B > Setting E > Setting A > Setting D.

Because Setting A directly train on Office-Home Dataset without pre-training backbone model, which means that the model did not learn any prior information, so it is not surprising that the performance of Setting A is not very well. As in Setting B, C, D and E, the result shows that if we fix the backbone parameters while fine-tuning the model, the performance is worse than those without fixing the backbone parameters. This is because the model with no parameters fixed would be more flexible to learn on new domain. So Setting B is better than setting D, and setting C is better than setting E. And comparing the result of Setting B and Setting C, it shows that the backbone pretrained with BYOL SSL method performs better than the backbone weights provided from TA, which is trained with label information. So we can conclude that the backbone trained with SSL-based method is more powerful than those trained with label-based. Finally in Setting D, it is unexpected that the performance is worse than Setting A, I think this is probably because the backbone weights from TA may not be robust enough or quite suitable on Office-Home Dataset, so if the parameters is fixed during fine-tune session, after fine-tuning for a few

epochs, the performance becomes worse than those backbone without any prior information and without fixing the backbone parameters, thus causing the unusual performance result.