Instruction pipeline and requirements for HW3

In this homework, you are required to implement a depth estimation ConvNet proposed in Hu, et al. In this paper, the authors propose an improved network design (Section 3.1) and an additional loss function for depth estimation (Section 3.2). You only have to implement the network in Sec 3.1 but not the additonal loss function. L1 Loss is good enough in this assignment.

You can work on this assignment in one of the following ways: remotely on our AI Cluster, or on your **local machine (recommended)**.
Because this is a shallow network, training will not be siginificantly faster on our GPU server. For reference, training would take around **4 hours** on our local quad-core i3 cpu with our instructors' implementation.
We recommend Python and Pytorch for implementing ConvNets in this assignment. You can find a nicely-written tutorial for image classification here.
However, if you feel more familiar with any other frameworks (e.g., tensorflow or mxnet), feel free to use them. There is no restriction on the packages/programming languages you use in this assignment.

Step A: Setting up your environments

Option 1: on your local computer (recommended)

```
# cpu only
pip3 install torch

# if you have gpu >= GTX950
# Install CUDA following instructions from nvidia
# https://developer.nvidia.com/cuda-10.2-download-archive?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1604&target_type=deblocal
pip3 install torch==1.10.0+cu102 -f https://download.pytorch.org/whl/cu102/torch_stable.html
```

run code snippet

Option 2: on our AI cluster

You need to first fill in the doc at https://docs.qq.com/sheet/DUUtnWVIXUlBjUlhT?tab=BB08J2 due on next Tuesday, Nov.26. We will then set up the account for you.
Before getting started, you are **required** to read through the documentations for AI Cluster. A tutorial for Jupyter Notebook is also available here.
Several things to notice:
1) Only the admin node has internet access. If you need pip install or apt get something, do it on the admin node.
2) Do not run your code on the admin node. Go to the compute node and finish your work there.
3) Always save your code or results to the /p300/* directory.

```
# first ssh into your server
ssh -p $YOUR_PORT root@10.15.89.41

# install pytorch
pip install torch==1.5.1+cu101 -f https://nelsonliu.me/files/pytorch/whl/torch_stable.html
# install packages if needed
pip install opencv-python ....

# ssh into compute node and run your scripts here.
ssh nodeXX
CUDA_VISIBLE_DEVICES=$GPUID python train.py
```

run code snippet

Step B: Datasets

The datasets are already available on your AI server at `/group/cs172hw3/`. We also provide an additional copy at OneDrive. There should be a total of 1690 images for training and 654 for testing. RGB Images are named as xxxx_color.jpg. Depth

maps are named as xxxx_depth.png. We provide a sample script for reading color imgs and depth maps in python in our github repo.

Step C: Training and Evalution
Note that the original paper uses ResNet50 as the encoder(E) for feature extraction. You only need to implement ResNet18 for now. The authors propose an additional loss in Section 3.2. This is not required here. L1 loss is enough in this assignment. For evalution, you will need to implement the mean relative error (REL) and mean log10 error (log10) as in the paper. Report these metrics on the test set in your final report.

Step D: Point Cloud Visulization
Now you have the images and their 2D depth maps. Choose one of the prediction results from the test set, reproject the points back into the 3D world coordinate system with the pinhole camera model. Camera intrinsics are provided in the github repo. Visualize the point cloud and include the screen shot in your final report. Matplotlib/Open3d/Meshlab is recommended for visualization.

Step E: Question
Which pixels would have the most impact on the output depth? Why?

Your report should include your results, how to run the code, the hyparameters you use (optimizer, learning rate, ...), problems you found, your solution and your new findings, etc.
Also, please use the CVPR template(which we have provided in github classrom), other formats are not acceptable.

Submission
Here are the things you need to submit:
1. A PDF-formatted report which describes what you have done ->Gradescope.
2. Homework code for hw3 -> github classroom.

For the first part, you should add CS172 course in your Gradescope. The course entry code is X355ND. Later we will add hw3 report submission channel so that you can submit your report. For the second part, you should accept this link https://classroom.github.com/a/uBqCWQ4_ to get the repo, which contains a README.md and a report template. You SHOULD fill your name and your student ID into README.md, otherwise we can't identify to whom this repo belongs and may give you 0 score for this homework. Please fill in your name and ID as soon as you clone the repo so that things mentioned above won't happen. Besides, you should include all your code into a zip file, naming it as hw3_studentID.zip(for example: hw3_2017123456.zip), so that we can keep a copy of your homework.

~~This assignment is due on **Monday, Nov. 15 2021 at 11:59pm.**~~
Edited: This assignment is due on **Monday, Nov. 22 2021 at 11:59pm.**

hw3

~ An instructor (赵子伯) thinks this is a good note ~

Updated 19 days ago by Lei Jin

followup discussions *for lingering questions and comments*

Resolved Unresolved

Chenyu Wang 1 month ago
We have received a total of 15 demand applications（before the line 18）in the doc of the step A, and we will not accept future applications. Once the accounts have been set up, we would update this post.
good comment | 1

Chenyu Wang 29 days ago AI cluster accounts have been set up, you can find in http://10.15.89.41:8899/permission

The tutorial pdf you can find in https://piazza.com/class_profile/get_resource/kth0jrlv1e12r1/kvgak97ap0y46z and http://10.15.89.41:8898/ . It is worth noting that the former pdf file is borrowed from the content of tutorial 1 of the CS280 course.

Also, the dataset has been stored under the **'cs172hw3'** directory.

You can try 'ssh nodeXX -p xxxxx[container port]' to enter into the compute nodes.
good comment | 0

Anonymous Poet 27 days ago May I ask what is the default password of our AI cluster accounts?

helpful! | 0

Anonymous Poet 27 days ago Nevermind found it in the tutorial of CS280

helpful! | 0

Lei Jin 27 days ago it should be sist

good comment | 0

Resolved Unresolved

姚凯欣 1 month ago
Hello! I can not download the dataset from the onedrive since it said that it is not shared to my microsoft mail, is there any solution or other ways to get the data set ? Thanks!

helpful! | 1

Lei Jin 1 month ago Hi, I edited the link above. This one should work just fine. https://hy999-my.sharepoint.com/:f:/g/personal/rxqy_hy999_onmicrosoft_com/EhRjv1OKUAJKkGHGzCc_GcoBg7pLjQA7e8rSpompqAyrpw?e=MNXasn

good comment | 0

姚凯欣 1 month ago Thanks a lot!

helpful! | 0

Resolved Unresolved

Anonymous Scale 29 days ago
Hi! After i read the question and the paper, i find that resnet50 can input random size of the picture and return the depth of each pixel, however , and the out put of resnet18 fc is 1x1000 which is not suitable for return back to the depth each pixel(like 224x224x1), how can we or can we revise resnet 18 model to meet the requirement?

helpful! | 0

Anonymous Scale 29 days ago Like up sample?

helpful! | 0

Lei Jin 29 days ago Hi, you have to replace the classification head (avgpool + fc) with the decoder (D, MFF, R in fig 2) proposed in the paper. There are several upsample layers in the decoder. So you can get a final output with the same size (or precisely 1/2) as the input image.

good comment | 1

Anonymous Scale 29 days ago Got it ! Thank you!

helpful! | 0

Resolved Unresolved

Anonymous Beaker 28 days ago
Hello! I wonder if we should implement the data augumentation methods mentioned in the paper ?(Flip, Rotation, Color Jitter)

helpful! | 0

Lei Jin 28 days ago Hi, it's optional.

good comment | 0

Resolved Unresolved

Anonymous Calc 26 days ago

Can we use resnet18 model in torchvision?

helpful! | 0

Lei Jin 26 days ago yes you can

good comment | 0

Resolved Unresolved

Anonymous Atom 22 days ago

Would you be so kind to give some keywords, instructions or some similar task examples written by pytorch? It's almost impossible for a freshman in CV/pytorch to finish this task without enough instructions… I've spent a huge amount of time on this homework but cannot even build the network, let alone training and debug. I mean I know there're several layer which do encoding or feature fusion, but try to solve EVERY problem turns out to be hundreds of other question. Please, it's shouldn't be the situation that this class assumes everyone is familiar with the implementation of CNN, and I really really need your help. Thank you ☹

helpful! | 5

Lei Jin 22 days ago Hi, you can check this tutorial from github pytorch-tutorial/main.py at master · yunjey/pytorch-tutorial · GitHub and this tutorial from stanford A detailed example of data loaders with PyTorch (stanford.edu). Note that there are several basic components of the first tutorial. a) dataset and loader (L17-31) b) ConvNets (L35-56) c) loss and optimizer (L61,62) d) training and evalution(L65-end). You can observe a similar pattern in the second tutorial from "A proposition of code template that you can write in your script is shown below. xxxx"

So a good starting point would be to replace these components with the one required in this homework.

For dataset and loader, tutorial A uses the standard mnist from pytorch. So you can check the second tutorial, which provides a detailed implementation for a custom dataset (the "The complete code corresponding to the steps that we described in this section is shown below." part). In short, you should implement your own "__getitem__" function, which given an index i, returns the ith X(rgb image) and y (depth map) then warp your dataset with "torch.utils.data.DataLoader"

For convnet, you can check the resnet implementation from the same repo pytorch-tutorial/main.py at master · yunjey/pytorch-tutorial · GitHub. In this assignment, you will need to replace line 112,113 with the proposed module in the paper (some extra conv and unsample layers.) You can implement these layers following the ResidualBlock in this repo. Or you can simply use one single upsample layer Upsample — PyTorch 1.10.0 documentation for easy debugging. You can still get a good performance without these complicated modules from the paper.

For loss and optimizer, pytorch provides L1 Loss L1Loss — PyTorch 1.10.0 documentation. You can use Adam optimizer as in the first toturial.

For training and evaluation, you can pretty much borrow all of the code from the first tutorial, but you will still need to implement another two metrics instead of accuracy.

Feel free to post any more questions here.

good comment | 2

连奕航 21 days ago

`Upsample()` seems not able to change the number of channels. How can I simply replace complicated layers with one single upsample layer as you said?

helpful! | 0

Lei Jin 21 days ago Yes that's true. To be more specific, you'll need to first use several convolution layers to deduct the num of channels to 1(since the target depth has only one channel) and then upsample to the original image size. You can implement such a network but you won't get full grades.

good comment | 0

Anonymous Helix 12 days ago do we need to do maxpooling before block1?

helpful! | 0

Lei Jin 11 days ago It depends. The standard implementation of ResNet would add a maxpooling layer before block1, while some would prefer to use a strided conv instead of maxpool.
good comment | 0

Resolved Unresolved

Anonymous Comp 12 days ago I can use only cpu to run my model, but still want to ask if this error solvable if I want to run the model on my gpu: "RuntimeError: CUDA out of memory. Tried to allocate 300.00 MiB (GPU 0; 2.00 GiB total capacity; 922.59 MiB already allocated; 167.19 MiB free; 934.00 MiB reserved in total by PyTorch) If reserved memory is >> allocated memory try setting max_split_size_mb to avoid fragmentation. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF".
helpful! | 0

Lei Jin 12 days ago Hi, that means you are running out of your GPU memory. You can check your current usage with 'nvidia-smi -l'. It says you have a total of 2G of memory while pytorch needs more to fit the network in. You can try to resize your input image to a smaller size or use a smaller batch size. For reference, our instructors' implementation with batch size 2, input image size 320x240 and resnet18 backbone requires only around 1.4G of GPU mem.
good comment | 0

Resolved Unresolved

Anonymous Gear 10 days ago
Will the performance of the model be affected if I do convolution first to deduct the num of channels to 1, then do Upsample()? Or maybe it's better if I first do Upsample and then convolution, just like the net presented in the paper?
helpful! | 0

Lei Jin 10 days ago The latter one should be better.
good comment | 0

Anonymous Gear 10 days ago Thank you!
By the way, I'd like to ask another question. Is there a recommanded batch size or number of epochs? On my computer, being limited by the GPU memory (4GB) , I can only set batch size to 2 or 1, which is hardly enough for the loss function to converge no matter how large the number of epochs is. How to solve this problem?
helpful! | 0

Anonymous Comp 10 days ago By the way, what's the kernel size of conv2~6 is recommended? Must it be 7? How do I know which kernel size should I choose? For the up-projection blocks, will a bilinear up-sampling work? Must we use unmaxpooling? (Since it is hard to use upmaxpooling without knowing indices and resize a picture to a size of odd number like the paper...)
helpful! | 0

Lei Jin 10 days ago In general, we would perfer a larger batch size (more stable batch norm stastics) and choose the epoch with the best validation accuracy. Bs=2 and around 5~10 epochs are good enough for this assignment. If you are running into convergence issues, a good starting point would be to pick only one image and check if your network can overfit to this single sample. If that fails, then try check your optimizer (lr too small/large?) or your network implementaiton.

For conv layers, we perfer a 3x3 kernel in most cases because it has less parameters. We can still stack multiple of them to build a deeper network. You can use a 3x3 kernel if not mentioned (And this shouldn't make too much difference in this assignment.) I'm not sure what "unmaxpooling" is but you can use bilinear upsampling (F.interpolate in pytorch) in the up-proj blocks.
good comment | 1