CSE252C HW1

0. Homework instructions

1. Attempt all questions.
2. Please comment all your code adequately.
3. Include all relevant information such as text answers, output images in notebook.

4. **Academic integrity:** The homework must be completed individually.

5. **Submission instructions:**
   (a) Submit the notebook and its PDF version on Gradescope.
   (b) Rename your submission files as Lastname_Firstname.ipynb and Lastname_Firstname.pdf.
   (c) Correctly select pages for each answer on Gradescope to allow proper grading.

6. **Due date:** Assignments are due Mon, May 4, by 4pm PST.

Steps to access and complete homework

- Clone the homework repository
  - `git clone https://github.com/eric-yyjau/cse252c_hw1.git`
- The homework is in the Jupyter Notebook `hw1-CSE252C.ipynb`
- Follow the README (this file) for installation, data and compute instructions.

1. Installation instructions

1. Set up the environment

1. [Option 1] On your own machine

- (local) SSH into your machine
- Install SWIG
  - On Ubuntu: `sudo apt-get install swig` (sudo required)
  - On MacOS: `brew install swig`
    - You need to install Homebrew first with HomeBrew
- Install Python 3.X and Pip
- [Recommended] Create an environment (e.g. with Anaconda)
  - `conda create --name py36 python=3.6 pip`
  - `conda activate py36`
- Install Jupyter Notebook
  - `conda install jupyter`
- Install kernels for Jupter Notebook
  - `conda install nb_conda`
- Launch Jupyter Notebook server in the conda env of the cluster
  - `jupyter notebook`
  - You will be provided with a URL that you can open locally
  - In a opened notebook, change the kernel (on Menu: **Kernel** -> **Change Kernel**) to the name of the conda env you just created (in the case of this documentation it should be `py36`)

2. [Option 2] On the `ieng6.ucsd.edu` server

- (local) **(IMPORTANT) Connect your UCSD VPN**
- (local) Login with your credentials
  - `ssh {USERNAME}@ieng6.ucsd.edu`
- If you cannot launch a pod, set up the environment following these instructions
- Launch your pod. You should enter a node with 1 GPU
  - `launch-scipy-ml.sh -i ucsdets/cse152-252-notebook:latest -g 1`
- You will be provided with a URL that you can open locally:

- Click on the link. Then natigate to the jupyter notebook for a question which you are going to git clone as follows

2. Pull the repo and install dependencies

- `git clone https://github.com/eric-yyjau/cse252c_hw1.git`
- Install dependencies (Python 3.X with Pip)
  - `pip install -r requirements.txt --user`
- Compile and install `pyviso` for the SfM question
  - `cd pyviso/src/`
  - `pip install -e . --user`

2. Data

On the `ieng6.ucsd.edu` server, the datasets are located at - Q1: SfM - `/datasets/cse152-252-sp20-public/dataset_SfM` - Change the dataset path in jupyter notebooks to your paths - Q5: - `/datasets/cse152-252-sp20-public/sfmlearner_h128w416` - `/datasets/cse152-252-sp20-public/kitti`

3. How to run

Q1: SfM - Working folder: `./pyviso`

Launch Jupyter Notebook

There is a `hw1-CSE252C.ipynb` jupyter notebook file in the top-level directory `cse252c_hw1`.

Options

One toggle `if_vis = True/False` allows you to enable/disable the visualization. Disabling the visualization will make the for loop run significantly faster.

Output

The errors are printed and the visualizations are saved at `vis/`. The images should look like: □ To fetch the files you can use commands like `scp` to transfer files from the cluster to your local machine:

From your local machine:

```
scp -r <USERNAME>@dsmlp-login.ucsd.edu:/datasets/home/53/253/cs152sp20ta1/pyviso2/vis {LOCAL PATH}
```

Or from within server if your local machine has a fixed address or IP:

```
scp -r {REMOTE PATH TO THE vis FOLDER} <USERNAME>@<LOCAL ADDRESS>:{YOUR LOCAL PATH}
```

Q4: Optical Flow - Working folder: `./opticalFlow`

4. [Extra] How to run training sessions

1. Set up the environment

[Option 1] On the `ieng6.ucsd.edu` server

- Login with your credentials
  - `ssh {USERNAME}@ieng6.ucsd.edu`
- Launch TMUX

    - Reconmended for session management: you can come back anytime after you disconnect your session. Otherwise you have to keep your connection on for hours while training.
    - Just run `tmux`
    - To detach and come back later, use `ctrl + b` then `d`. To attach next time, use `ctrl + b` then `a`.
    - For more TMUX usages please refer to online tutorials like https://linuxize.com/post/getting-started-with-tmux/

- Launch your pod
    - Follow Section 1.1.2

[Option 2] On your own server

Just launch TMUX.

2. Start training

Now you can create conda env and do your training in there following Section 1.1