**Part 1: Getting Started**

**Introduction:**

*This Part tutorial starts with this document, where preliminaries of conda environments, python packages, and running a jupyter notebook server are covered. The remainder of this tutorial takes place within* ***Part\_1\_Getting\_Started.ipynb****. Jupyter notebooks with the* ***ipynb*** *extension will introduce and exemplify all remaining Parts. You will submit all deliverables as Jupyter notebooks, so developing a familiarity with notebook syntax and best practice in this Part is valuable.*

**Computer requirements:**

*You are encouraged to use Google Colab to run all tutorials in Parts 3, 4, and 5 if your computer hardware does not meet the following specifications:*

* *Central Processing Unit (CPU) — Intel Core i5 6th Generation processor or higher, or an AMD equivalent processor.*
* *RAM — 8 GB or higher is recommended.*
* *Graphics Processing Unit (GPU) — NVIDIA GeForce GTX 960 or higher. AMD GPUs are not able to execute the code in this project. Read* [*this guide*](https://developer.nvidia.com/cuda-gpus) *for more information on NVIDIA GPUs for deep learning.*
* *Operating System —* [*Ubuntu*](https://ubuntu.com/) *Linux (or Ubuntu derivatives such as* [*pop!OS*](https://system76.com/pop)*) or Microsoft Windows 10*

**Setting up a conda environments and packages:**

*You will use* ***conda*** *to create a virtual python 3.7 environment, This will help ensure all learners use the same environment and installed packages. To ensure an environment that works well on multiple platforms, we will use both* ***conda*** *and* ***pip*** *to install packages. Conda is used to install the base environment, python and a few other core dependencies, and pip is used to install the remaining packages to complete the environment.*

*Refer to* [*this guide*](https://www.anaconda.com/understanding-conda-and-pip/) *to understand the differences between these two different package/environment management tools. We will use conda to install the bare bones (an environment, python and core dependencies that are more difficult to install with pip), and* ***pip*** *for everything else.*

*Windows and Linux users are encouraged to use the* [*anaconda*](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)) *python 3.7 distribution. Windows users should refer to this comprehensive* [*installation guide*](https://katiekodes.com/setup-python-windows-anaconda/)*. Debian-based linux users might find* [*this*](https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart) *useful.*

*Open an anaconda command window and make sure your* ***conda*** *environment is up-to-date:*

conda update -n base -c defaults conda

*If you are up to date, this will not do anything. More* ***conda*** *commands are* [*here*](https://conda.io/docs/user-guide/tasks/manage-environments.html)*.*

**Getting the LiveProject materials:**

*The instructions below describe how to obtain the liveproject materials and create a conda environment for this project on your computer or cloud computer. These instructions have been tested on Windows and Linux.*

*Download the project code and materials from here:*

INSERT URL HERE

*Read more about* ***conda*** *environments and the rationale behind them* [*here*](https://towardsdatascience.com/a-guide-to-conda-environments-bc6180fc533)*. Next change directory into your cloned repository:*

cd DeepLearningSatelliteImage

*Create an environment called ‘liveproject’ using a yml file:*

conda env create -f conda\_env/env.yml

*To activate the* ***conda*** *environment and clean up the downloaded packages to save computer memory, issue the following commands:*

conda activate liveproject

conda clean --all

*To deactivate the* ***conda*** *environment (recommended before you close the anaconda command window):*

conda deactivate

*If you want to remove the environment, you could use:*

conda env remove --name liveproject

or conda remove –name liveproject --all

**Installing necessary python libraries:**

*Use* ***pip*** *to install the rest of the required packages in* ***the requirements.txt*** *file:*

(on Windows) pip install –r requirements.txt

(on Linux) python -m pip install --user -r requirements.txt

*If you get timeout errors, change the timeout to a higher number, for example:*

pip install -r requirements.txt --default-timeout=100

*If you have a good* ***Nvidia Graphics Card*** *on your machine and you do not wish to use free access to GPUs on Google Colab, instructions for Windows users are provided below:*

1. *Install* [*CUDA Toolkit*](https://developer.nvidia.com/cuda-downloads)*. Choose your version (9.0 or 8.0) depending on your Operating System and GPU. Here is a guide to check that if your version support your. For downloading other versions,* [*follow this*](https://developer.nvidia.com/cuda-toolkit-archive)*. To avoid errors, choose the CUDA version according to your Nvidia GPU version.*
2. *Download the latest version of* [*cuDNN*](https://developer.nvidia.com/rdp/cudnn-download)*. Choose your version depending on your Operating System and CUDA version. Membership registration is required.* *Put your unzipped folder in your root C drive, for example C:\cudnn-9.0-windows10-x64-v7*
3. *Add cuDNN into your Environment Path. Open Run dialogue using (Win + R) and run the command sysdm.cpl. In System Properties, select the Advanced tab, the Environment Variables. Add the cudnn path in your Environment (e.g. C:\cudnn-9.0-windows10-x64-v7).*
4. *Install tensorflow-gpu==2.0 in the requirements.txt file, instead of tensorflow==2.0.*

**Jupyter notebooks:**

*The rest of the Project is conducted through a series of* ***jupyter*** *notebooks. To run* ***jupyter*** *notebooks on your PC through your browser, issue this command:*

jupyter notebook

*This should launch a new tab in your default web browser that displays a list of jupyter notebooks (with .ipynb extension). You should launch this from the highest level directory in order to access notebooks within subfolders. Once there, you can navigate through the list and launch your notebooks by double-clicking on the name. Alternatively, you can launch any notebook from the command line by navigating to the directory and issuing*

*jupyter notebook A\_Notebook.ipynb*

**From here, go to Part\_1\_Getting\_Started.ipynb**