**Assignment 1: Software Installation**

The tasks for the week will mostly be installation and maybe some introductory reading and running a bit of code just to make sure things are installed correctly.

I wrote some background for clarification, with instructions below.

**Background**

We will be installing and working with two things: **OpenCV** and **Jupyter Notebook.**

**OpenCV**

As the name suggests, OpenCV is an open source library and image processing/computer vision (CV) toolkit for C++ and Python. It contains lots of relevant functions we can use for common CV tasks. To use it after it’s installed, you just need to include or import it in your code, just like with any other library in C++/Python.

We will be using Python-OpenCV. Note that C++ is usually preferred over Python for time-sensitive computations like real-time image processing, detection, and/or tracking algorithms since C++ is a faster/higher performance language than Python. However, we’ll use Python because

1. Python-OpenCV’s performance is apparently about the same as C++ OpenCV (according to this post on stackoverflow <https://stackoverflow.com/a/13433330>). In any case, it should be fine for our purposes atm, and also

2. More people in the team are probably familiar with Python compared to with C++

**Jupyter Notebook**

Jupyter Notebook is an open source web application where you can create code documents that have runnable Python code, text, and code outputs all at the same time. This is nice because you can use it to make presentable program files that other people can run. Lots of tutorials you can find online use these notebooks so that people can follow along easily.

[Here's an example (not mine) — the whole thing (pictures, text, code) is one notebook file.](https://github.com/Tanu-N-Prabhu/Python/blob/master/Predicting_PewDiePie's_daily_subscribers_using_Machine_Learning_.ipynb)

Jupyter Notebook runs locally on your computer and opens up in a tab in your browser. To use it after it’s installed, you just type a command in the command line/terminal. The file extension for a notebook is “.ipynb”, which stands for ipython notebook.

**Jupyter Lab (optional)**

This is a newer version of Jupyter Notebook, which has a different interface and is more like an IDE. It can do whatever Notebook can and has more settings. You don’t have to use this, but I like it more— it also has a night mode theme so you don’t burn your eyes.

**Instructions**

First install either Anaconda or pip. Either can be used to install OpenCV and Jupyter Notebook. Then try to create a Jupyter Notebook file and run some OpenCV functions in it. Also, I wrote some notes about Anaconda/PIP below these instructions.

If you want to use Anaconda (PIP instructions are on the next page):

1. Install Anaconda
   1. Download and install here: <https://www.anaconda.com/products/individual>
   2. Jupyter Notebook should come included in the base (root) environment of Anaconda, so we don’t need to install Jupyter Notebook after installing Anaconda.
2. Try running Jupyter Notebook
   1. To run Jupyter Notebook, open Anaconda Prompt on your computer and type and enter:

jupyter notebook

A browser window or tab should open with the application. If none does, there should be a URL in Anaconda Prompt that you can paste into any browser to access Jupyter Notebook.

* + 1. You have to use Anaconda Prompt and not a regular command prompt for this to work, since Anaconda Prompt activates and uses the virtual environment that has Jupyter Notebook installed.
  1. Now close the tab and Anaconda Prompt to exit the web application.

1. Install OpenCV
   1. In an Anaconda Prompt window, type and enter:

conda install -c anaconda opencv

* 1. OpenCV should be installed in the base (root) Anaconda environment now. If you open Jupyter Notebook, you should be able to create code files where you can write import cv2 and use OpenCV functions in your code.

If you want to use PIP:

1. Install PIP
   1. Open a terminal/command prompt. You might already have PIP installed, since it’s included in Python 2 >=2.7.9 and Python 3 >=3.4. Run

pip help

and see if PIP is there (it would give you a list of PIP commands). If you already have PIP, you can try to update it using

python -m pip install --upgrade pip

in the command prompt.

* 1. If you don’t have PIP, follow the instructions here to get it: <https://pip.pypa.io/en/stable/installing/>

1. Install Jupyter Notebook
   1. To install Jupyter Notebook, type and enter:

pip install notebook

in the terminal.

1. Try running Jupyter Notebook
   1. To run Jupyter Notebook, type and enter:

jupyter notebook

in the terminal.

* 1. A browser window or tab should open with the application. If none does, there should be a URL in the terminal that you can paste into any browser to access Jupyter Notebook.
  2. Now close the browser tab and the terminal window to exit the web application.

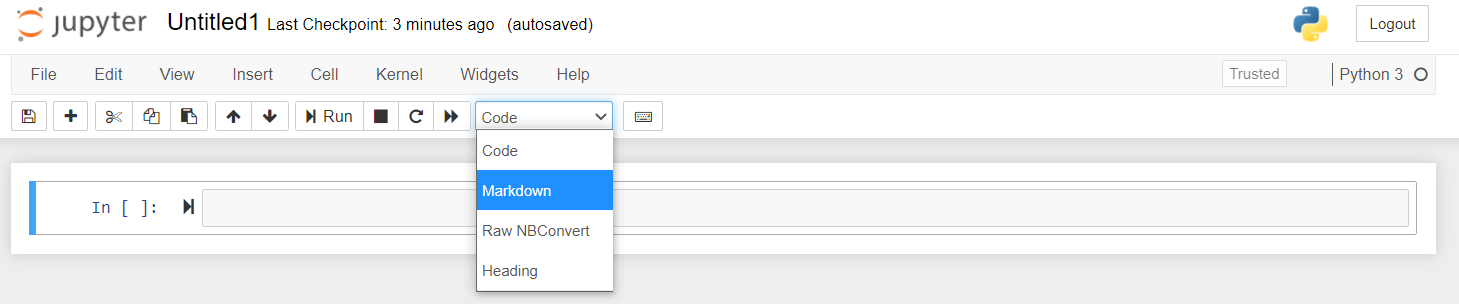
1. Install OpenCV
   1. In a terminal/command prompt window, type and enter:

pip install opencv-python

* 1. OpenCV should be installed now. If you open Jupyter Notebook, you should be able to create code files where you can write import cv2 and use OpenCV functions in your code.

Now that OpenCV and Jupyter Notebook are installed, try to make a new Python 3 Notebook file (.ipynb) and see if you can run some OpenCV functions.

1. Open Jupyter Notebook.
2. Navigate to the folder directory where you’d like to create the notebook file.
3. On the right side, click “New” and select “Python3” to make a Python notebook. It will open in another tab.
4. There should already be a **cell** in the new notebook. This is by default a **code cell**, and you can run it by pressing Shift+Enter while the cell is selected (there will be a green border around it), or by clicking the Run button, or Cell > Run Cells from the top menu bar. You can change the cell into a **markdown cell** by selecting Markdown in the dropdown menu as shown below.



1. Copy and paste the short sample code from this OpenCV tutorial <https://docs.opencv.org/master/db/deb/tutorial_display_image.html>

into the code cell.

Then edit the line

**img =** [**cv.imread**](https://docs.opencv.org/master/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56)**(**[**cv.samples.findFile**](https://docs.opencv.org/master/d6/dba/group__core__utils__samples.html#ga3a33b00033b46c698ff6340d95569c13)**("starry\_night.jpg"))**

so that you have

**img =** [**cv.imread**](https://docs.opencv.org/master/d4/da8/group__imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56)**(filename)**

This code cell will find an image file and display it in a window when it is run. You have to specify the image path by replacing **filename** with the name or path of the image you want to use. If you put your image on the same directory level (in the same folder) as this Notebook, then you just need to put the name and file extension of the image (i.e., “starry\_night.jpg”).

1. Try running the code cell and see if a window with the image appears.

**Additional Notes on Anaconda/PIP**

**Anaconda** and **PIP** (preferred installer program) are both software package managers that let you install things using the terminal / command line.

**Anaconda** also comes with other packages, so if you don’t want them then you can use Pip: Anaconda is a package and virtual environment (venv) organization software which should come with Jupyter Notebooks already installed, as well as other common data science packages. It also has a graphic user interface (GUI) you can use in place of the command line.

I personally use Anaconda because I also use it for other classes and projects, and I can use it to create venv’s for individual projects easily. Then for each project I can install whatever Python libraries I need in its venv, and “activate” it whenever I need those libraries to work on that project. This way I don’t have everything installed globally on my computer.