Colab Walkthrough

Neural Style Transfer

Let's say that for your final project you want to work on Neural Style Transfer. As a starting point, you find a repository on Github (https://github.com/anishathalye/neural-style) that can serve as a starting point for your group. If you want to run this repository on your computer, you can do so using the following steps.

Step 1: Installing git
Step 2: Using git to download the package
Step 3: Installing python3
Step 4: Installing pip3
Step 5: Installing virtualenv
Step 6: Creating a new virtual environment
Step 7: Installing the project's package dependencies
Step 8: Downloading the project's dataset / pretrained models
Step 9: Running the model

As you might notice, there are quite a few steps involved with testing something, and it can be difficult to setup before running the project. Further, if you run the project on your computer, chances are that it will take a few hours to run. This is because you are running the project on your computer's CPU (these days models are mostly trained and evaluated on GPUs, which are several orders of magnitude faster).

Instead of installing packages on your computer, if you want to test a project quickly, there is an alternative - Google Colab.

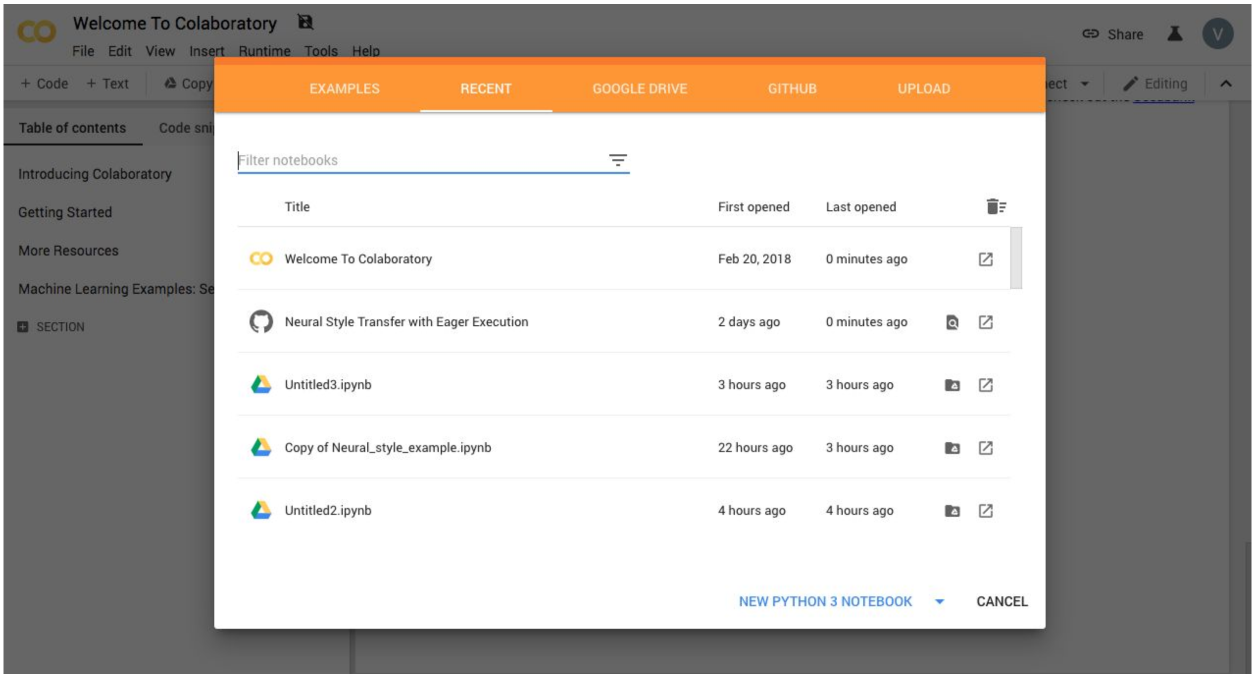
What is Google Colab?

Colab (short for Colaboratory) is a product offered by Google Research that allows machine learning researchers to work on projects in the browser. Similar to Google Docs, it allows you to share projects between many people, and best of all, it gives free access to GPUs for you to quickly train models without any signup.

When working on your projects, we recommend testing your code on a portion of your dataset on your own computer, or on Google Colab initially. You can use these tools as a quick way to see if your code has bugs or if the output of your model is reasonable. After testing initially, you can then copy your code and full dataset to your group's AWS instance to run the full version of your model (as testing locally / on Colab is free, whereas you will be using the course credits for running on AWS).

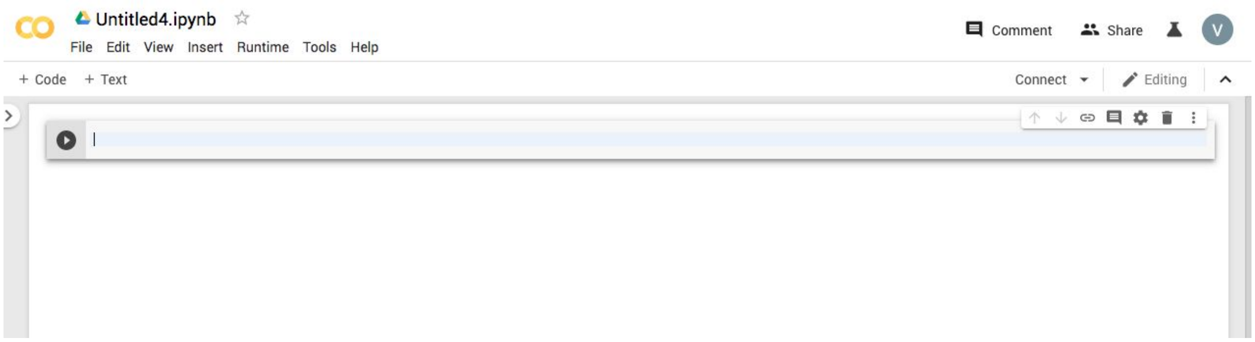
Getting Started

Visit http://colab.research.google.com. You should see a page like this. Click the "New Python 3 Notebook" in the bottom right corner.



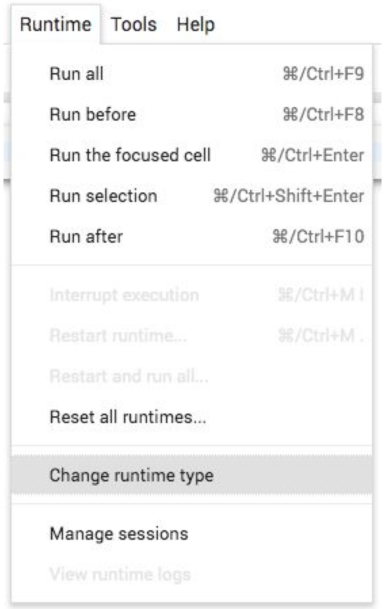
Title	First opened	Last opened
Welcome To Colaboratory	Feb 20, 2018	0 minutes ago
Neural Style Transfer with Eager Execution	2 days ago	0 minutes ago
Untitled3.ipynb	3 hours ago	3 hours ago
Copy of Neural_style_example.ipynb	22 hours ago	3 hours ago
Untitled2.ipynb	4 hours ago	4 hours ago

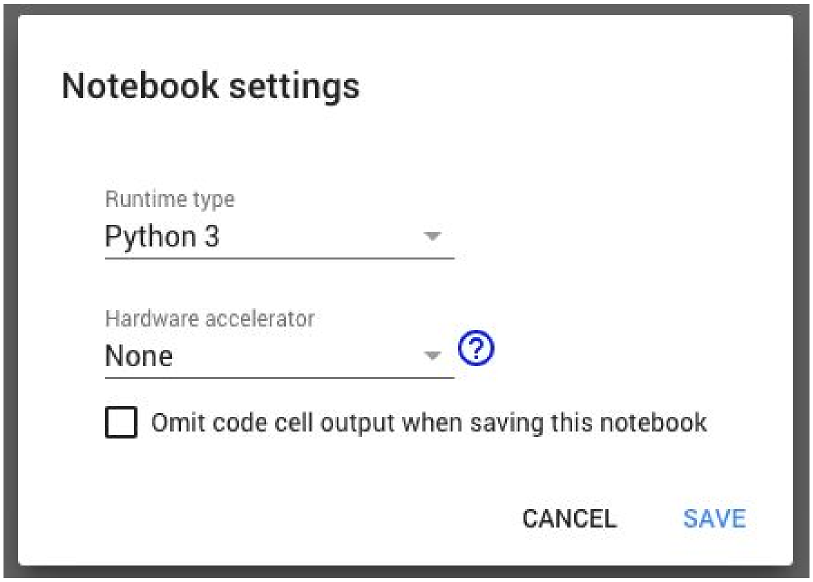
That will bring you to a page like this. This is your new project's notebook, a document that you can code in and run models in. You can click "Untitled.ipynb" in the top left corner to change the title of your document, or click "Share" in the top right corner to share it with other people.



Each notebook is composed of a list of cells. These cells can contain text (where you might describe what you're doing) and code (where you can type Python and run it). This notebook format is very similar to the Coursera homework assignments.

Before you start running code, there's one quick step to do first. Click on "Runtime" and then "Change runtime type". You should see the following window pop up:





Notebook settings

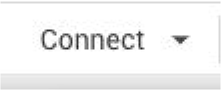
Runtime type
Python 3

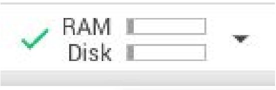
Hardware accelerator
None

□ Omit code cell output when saving this notebook

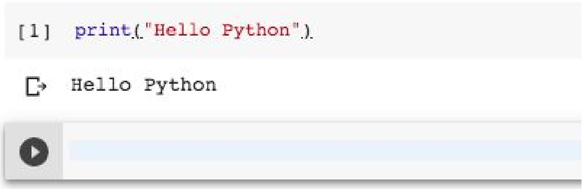
Make sure that "Python 3" is selected as the Runtime type (this is the version of Python we are using). Then, make sure "GPU" is selected for the Hardware accelerator. This will allow us to run deep learning models in the matter of minutes instead of hours. You can then click save.

Now, click the "Connect" button in the top right. After doing so, you should see a green checkmark, indicating you have successfully connected to a GPU to run your code.





To make sure everything is working, try writing some Python in the notebook (click on the open cell). You can run it by pressing SHIFT + ENTER. As an example, you might see:



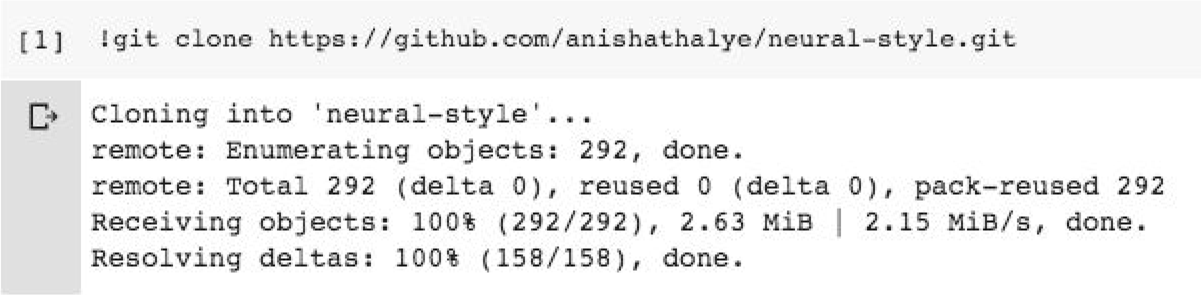
```
[1] print("Hello Python")

    Hello Python
```

Running Neural Style Transfer

Now going back to our original neural style transfer project, https://github.com/anishathalye/neural-style, let's say you want to try running this on Colab. There are a few steps to follow:

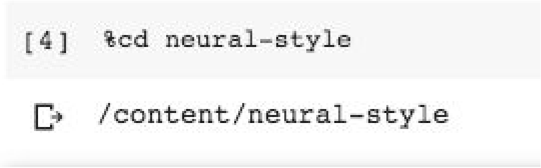
Step 1: Downloading the Project

By default, each cell in Colab runs Python code. However, we can also run special commands to download projects and install packages using a **!** before the command. For example, to download the project, you would run:



```
[1]  !git clone https://github.com/anishathalye/neural-style.git

     Cloning into 'neural-style'...
     remote: Enumerating objects: 292, done.
     remote: Total 292 (delta 0), reused 0 (delta 0), pack-reused 292
     Receiving objects: 100% (292/292), 2.63 MiB | 2.15 MiB/s, done.
     Resolving deltas: 100% (158/158), done.
```

Now that the project has been downloaded, you want to enter that project directory. You can do so using the **cd** command, which in colab requires a **%** before it:

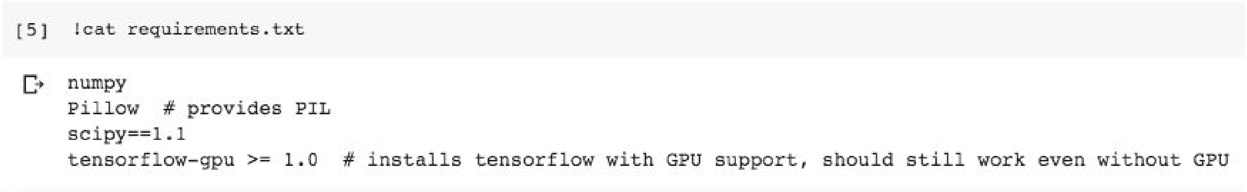


```
[4]  %cd neural-style

     /content/neural-style
```

The outputted */content/neural-style* is the new directory you are in.

Step 2: Installing the Package Requirements

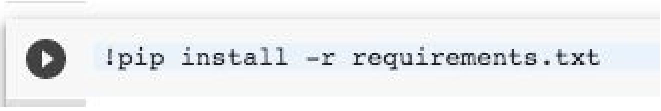
Generally Python projects come with *requirements.txt* files that indicate which packages that codebase depends on. You can see this file using the *cat* command, as shown below:



```
[5]  !cat requirements.txt
```

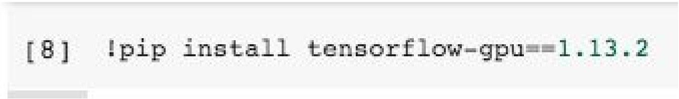
```
numpy
Pillow  # provides PIL
scipy==1.1
tensorflow-gpu >= 1.0  # installs tensorflow with GPU support, should still work even without GPU
```

This project depends on numpy (for math operations), Pillow (for manipulating images), scipy (for scientific operations), and tensorflow-gpu (for training neural networks on the GPU). You can install these packages by running the following. You might get some warnings/errors but you can ignore those.



```
!pip install -r requirements.txt
```

Now as one final step, you'll need to install tensorflow 1.13.2. In the requirements file, the author lists numpy, Pillow, scipy==1.1, and any version of tensorflow >= 1.0. Just recently a new version of tensorflow was released (2.0) and this codebase no longer works with 2.0. To install the right version of tensorflow, run:

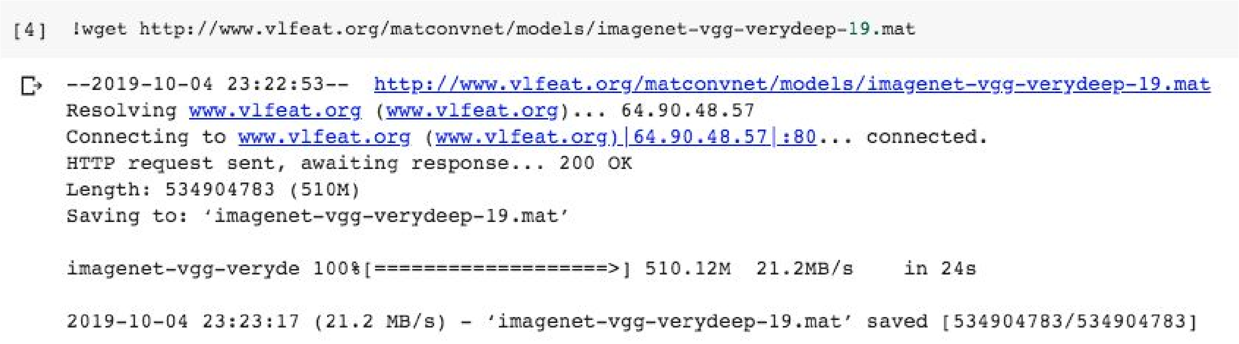


```
[8]  !pip install tensorflow-gpu==1.13.2
```

With that step complete, all the packages and source code you need to run the project should be downloaded and you should be ready to go.

Step 2: Downloading the Pretrained Model

This repo contains a pretrained model called VGG that you will need to download before running. You can do so using the *wget* command, as shown below:

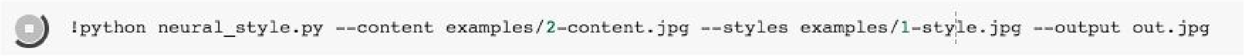


```
[4]  !wget http://www.vlfeat.org/matconvnet/models/imagenet-vgg-verydeep-19.mat
```

```
--2019-10-04 23:22:53--  http://www.vlfeat.org/matconvnet/models/imagenet-vgg-verydeep-19.mat
Resolving www.vlfeat.org (www.vlfeat.org)... 64.90.48.57
Connecting to www.vlfeat.org (www.vlfeat.org)|64.90.48.57|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 534904783 (510M)
Saving to: ‘imagenet-vgg-verydeep-19.mat’

imagenet-vgg-veryde 100%[===================>] 510.12M  21.2MB/s    in 24s

2019-10-04 23:23:17 (21.2 MB/s) - ‘imagenet-vgg-verydeep-19.mat’ saved [534904783/534904783]
```

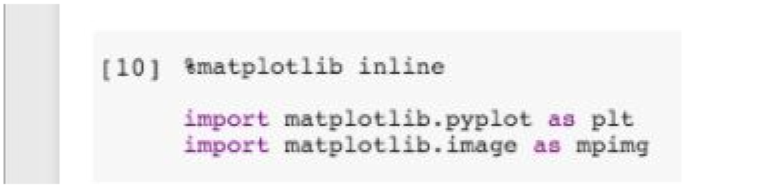
Step 3: Running and Visualizing the Output

To run the model:

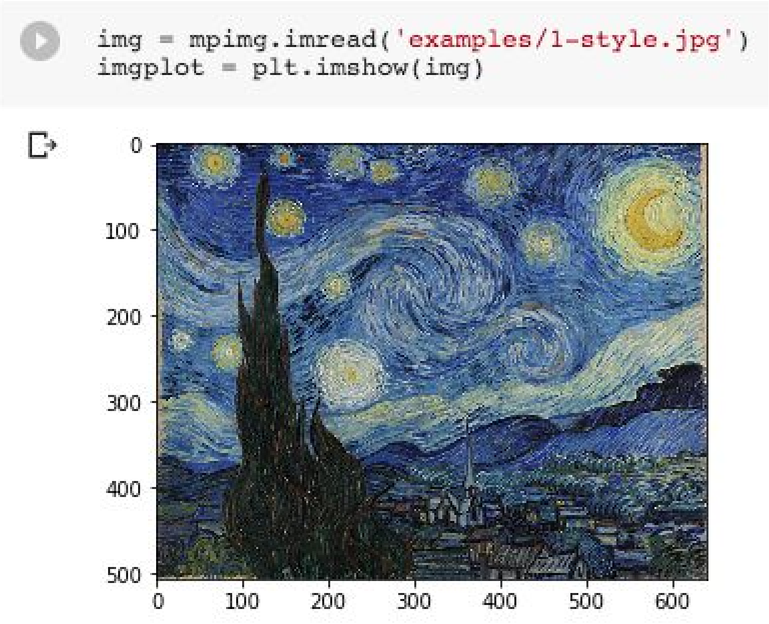


```
!python neural_style.py --content examples/2-content.jpg --styles examples/1-style.jpg --output out.jpg
```

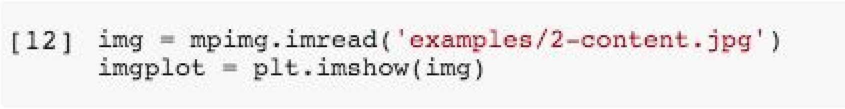
To visualize the inputs/outputs:



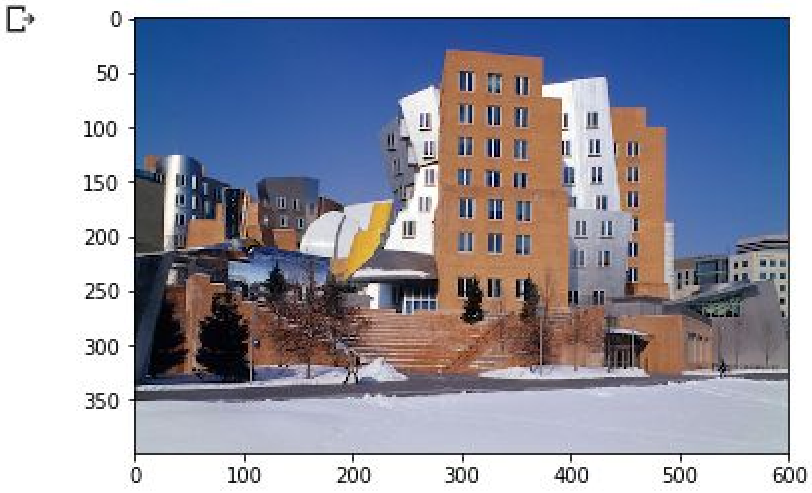
```
[10] %matplotlib inline

     import matplotlib.pyplot as plt
     import matplotlib.image as mpimg
```

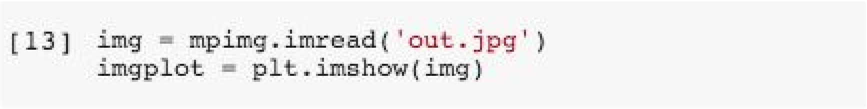


```
img = mpimg.imread('examples/1-style.jpg')
imgplot = plt.imshow(img)
```

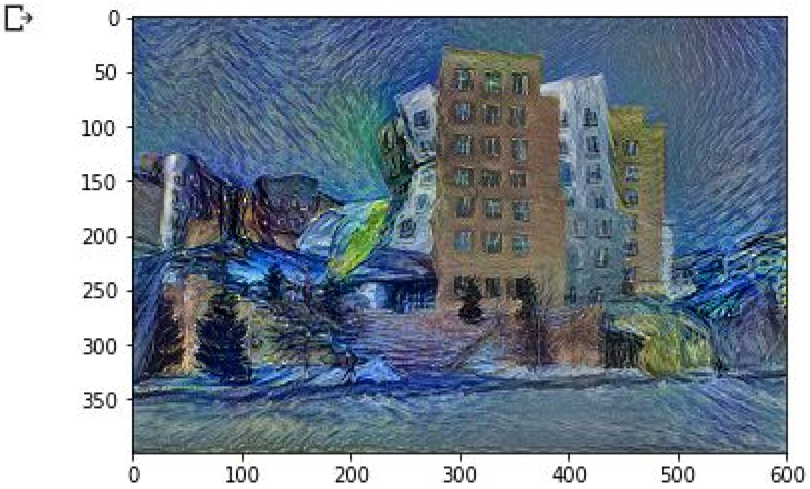


```
[12] img = mpimg.imread('examples/2-content.jpg')
     imgplot = plt.imshow(img)
```





```
[13] img = mpimg.imread('out.jpg')
     imgplot = plt.imshow(img)
```


0
50
100
150
200
250
300
350
0
100
200
300
400
500
600