

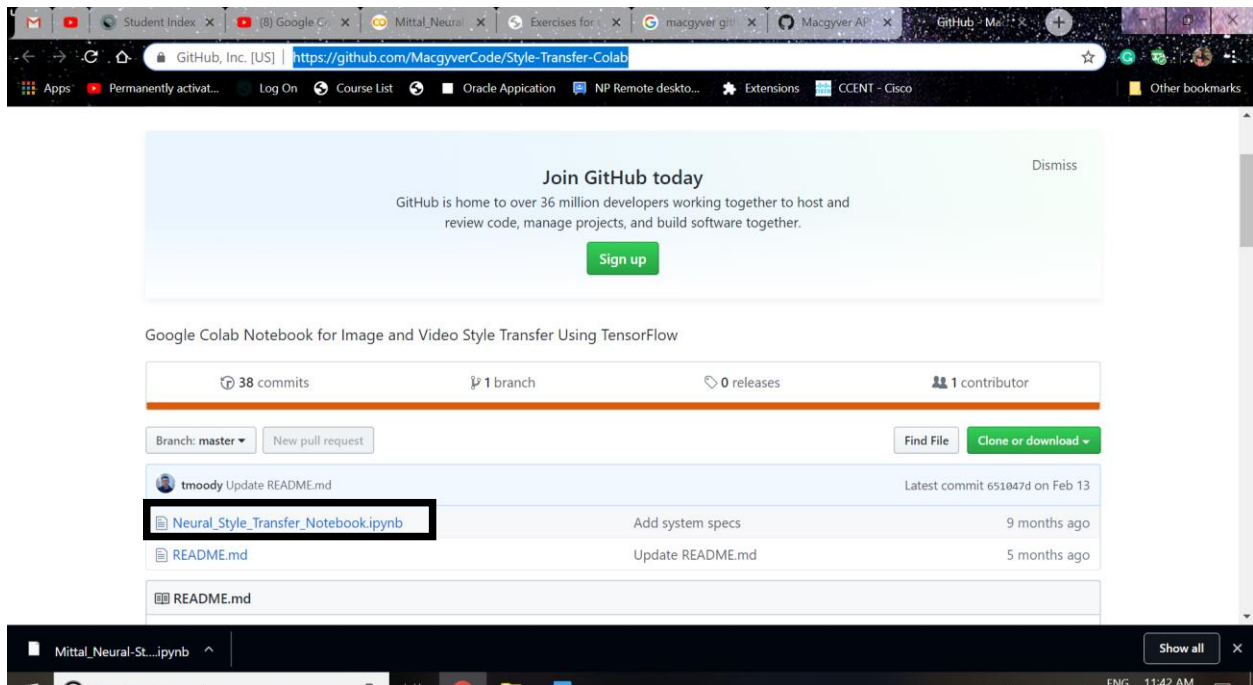
Google Colaboratory Notebook Tutorial with GPU

Step 1: Go to Macgyver API GitHub Profile (<https://github.com/MacgyverCode>)

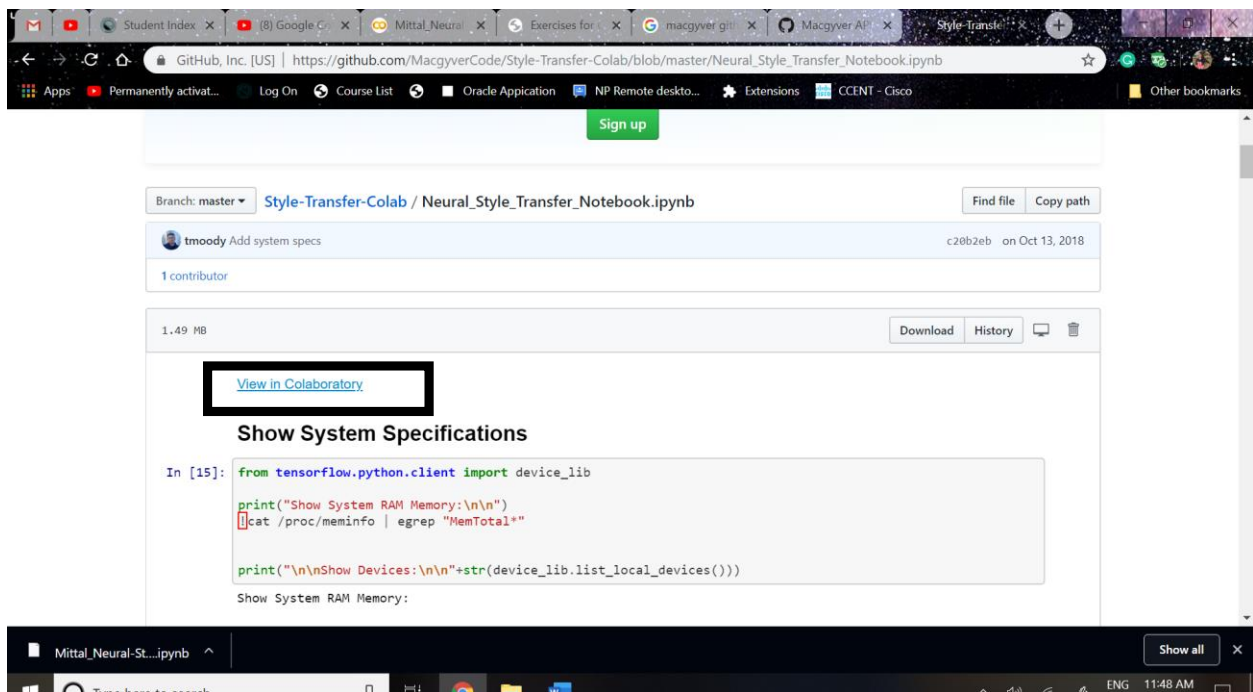
Step 2: Go to Style-Transfer-Colab Repository (<https://github.com/MacgyverCode/Style-Transfer-Colab>)

The screenshot shows a web browser window with the GitHub repository page for 'Style-Transfer-Colab' by MacgyverCode. The browser's address bar shows the URL <https://github.com/MacgyverCode/Style-Transfer-Colab>. The page features a 'Join GitHub today' banner at the top, followed by repository statistics: 38 commits, 1 branch, 0 releases, and 1 contributor. Below this, there are buttons for 'Find File' and 'Clone or download'. A list of recent commits is displayed, including 'tmoody Update README.md' (latest commit 651047d on Feb 13), 'Neural_Style_Transfer_Notebook.ipynb' (Add system specs, 9 months ago), and 'README.md' (Update README.md, 5 months ago). The bottom of the page shows a dark sidebar with the file 'Mittal_Neural-St...ipynb' selected and a 'Show all' button.

Step 3: Open Neural_Style_Transfer_Notebook



Step 4: Open Program in Google Colaboratory



Step 5: Copy code in Google Drive

The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: https://colab.research.google.com/github/MacgyverCode/Style-Transfer-Colab/blob/master/Neural_Style_Transfer_Notebook.ipynb#scrollTo=fNb1.... The notebook title is "Neural-Style-Transfer-Notebook.ipynb". The "COPY TO DRIVE" button is highlighted with a black box. The left sidebar shows a "Table of contents" with items like "Show System Specifications", "Download Dependencies to Google Colab Environment", "Download Style and Content Images", and "Stylize an Image". The main area displays the "Show System Specifications" section, which includes a code cell with the following Python code:

```
[ ] from tensorflow.python.client import device_lib
print("Show System RAM Memory:\n\n")
!cat /proc/meminfo | egrep "MemTotal*"

print("\n\nShow Devices:\n\n"+str(device_lib.list_local_devices()))
```

Below the code, the output shows "Show System RAM Memory:" with "MemTotal: 13335212 kB" and "Show Devices:".

Step 6: Open from Google Drive and change the Name

The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1ExUmmAnzriz9hGXrW3T70JaLhUb1fDy>. The notebook title is "Mittal_Neural-Style-Transfer-Notebook.ipynb". The left sidebar shows a "Table of contents" with items like "Show System Specifications", "Download Dependencies to Google Colab Environment", "Download Style and Content Images", and "Stylize an Image". The main area displays the "Show System Specifications" section, which includes a code cell with the following Python code:

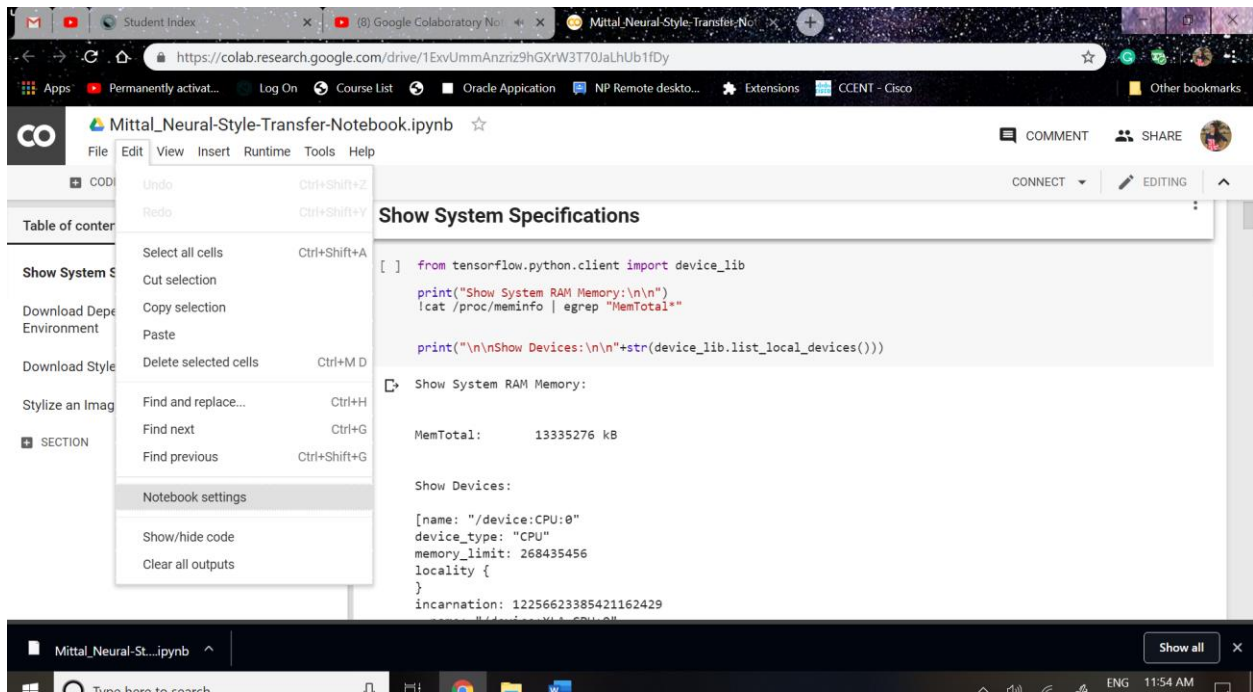
```
[ ] from tensorflow.python.client import device_lib
print("Show System RAM Memory:\n\n")
!cat /proc/meminfo | egrep "MemTotal*"

print("\n\nShow Devices:\n\n"+str(device_lib.list_local_devices()))
```

Below the code, the output shows "Show System RAM Memory:" with "MemTotal: 13335276 kB" and "Show Devices:" with the following JSON output:

```
{
  "name": "/device:CPU:0",
  "device_type": "CPU",
  "memory_limit": 268435456,
  "locality": {
  },
  "incarnation": 12256623385421162429
}
```

Step 7: Setup Environment for Notebook



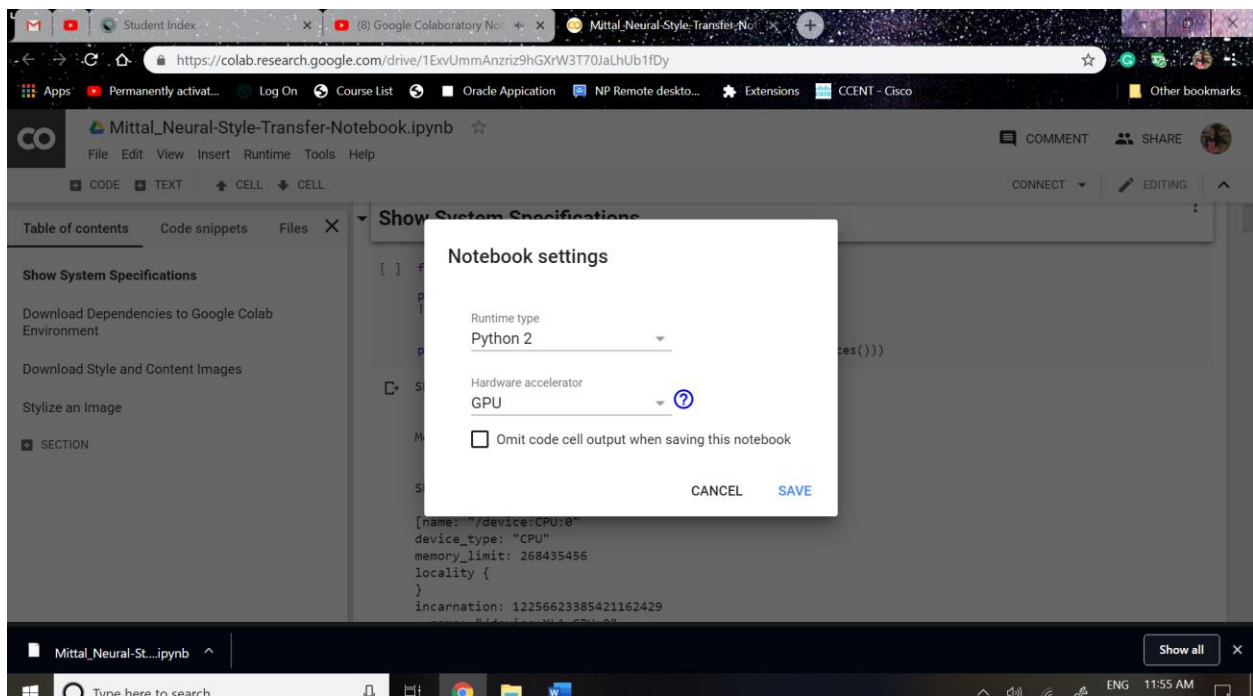
The screenshot shows a Google Colaboratory notebook titled "Mittal_Neural-Style-Transfer-Notebook.ipynb". The left sidebar contains a "Table of contents" with items like "Show System Specifications", "Download Dependencies to Google Colab Environment", and "Download Style and Content Images". The main editor area displays the "Show System Specifications" section, which includes the following code:

```
[ ] from tensorflow.python.client import device_lib  
  
print("Show System RAM Memory:\n\n")  
!cat /proc/meminfo | egrep "MemTotal:"  
  
print("\n\nShow Devices:\n\n"+str(device_lib.list_local_devices()))
```

Below the code, the output shows the system RAM memory and the list of devices:

```
MemTotal:      13335276 kB  
  
Show Devices:  
  
[name: "/device:CPU:0"  
 device_type: "CPU"  
 memory_limit: 268435456  
 locality {  
 }  
 incarnation: 12256623385421162429  
 ...]
```

A menu is open over the code editor, showing options like "Undo", "Redo", "Select all cells", "Cut selection", "Copy selection", "Paste", "Delete selected cells", "Find and replace...", "Find next", "Find previous", "Notebook settings", "Show/hide code", and "Clear all outputs".



The screenshot shows the same Google Colaboratory notebook, but with the "Notebook settings" dialog box open. The dialog box has the following settings:

- Runtime type: Python 2
- Hardware accelerator: GPU
- ☐ Omit code cell output when saving this notebook

The "SAVE" button is highlighted in blue. The background shows the same code and output as the previous screenshot.

Step 8: There are two dependencies 1. `neural_style.py` (Python Code) and 2. ImageNet

Download Dependencies to Google Colab Environment

This may take several minutes but needs to only be run once per session. Every Google Colab session has temporary storage for assets such as images and other scripts. When the session is closed these objects will be removed but the notebook will remain intact.

1. `neural_style.py`
2. ImageNet

Run the cell below to download necessary dependencies. After this is complete you can navigate to the FILES tab on the left and select refresh - you should see the files listed.

```
[ ] #Download neural_style.py
#This code originates from https://github.com/cysmith/neural-style-tf
!wget --output-document=neural_style.py 'https://storage.googleapis.com/marketing-files/colab-notebooks/style-'
!cat ./wget-log

#Download ImageNet VGG Very Deep
#VGG-VD models from the Very Deep Convolutional Networks for Large-Scale Visual Recognition project.
!wget --output-document=imagenet-vgg-verydeep-19.mat 'https://storage.googleapis.com/marketing-files/colab-notebooks/imagenet-vgg-verydeep-19.mat'
```

1. `neural_style.py` (Python Code): taken from the <https://github.com/cysmith/neural-style-tf>.

cysmith / neural-style-tf

131 Watch 2,273 Star 601 Fork

Code Issues 40 Pull requests 1 Projects 0 Wiki Security Insights

TensorFlow (Python API) implementation of Neural Style

style-transfer tensorflow convolutional-neural-network deep-learning

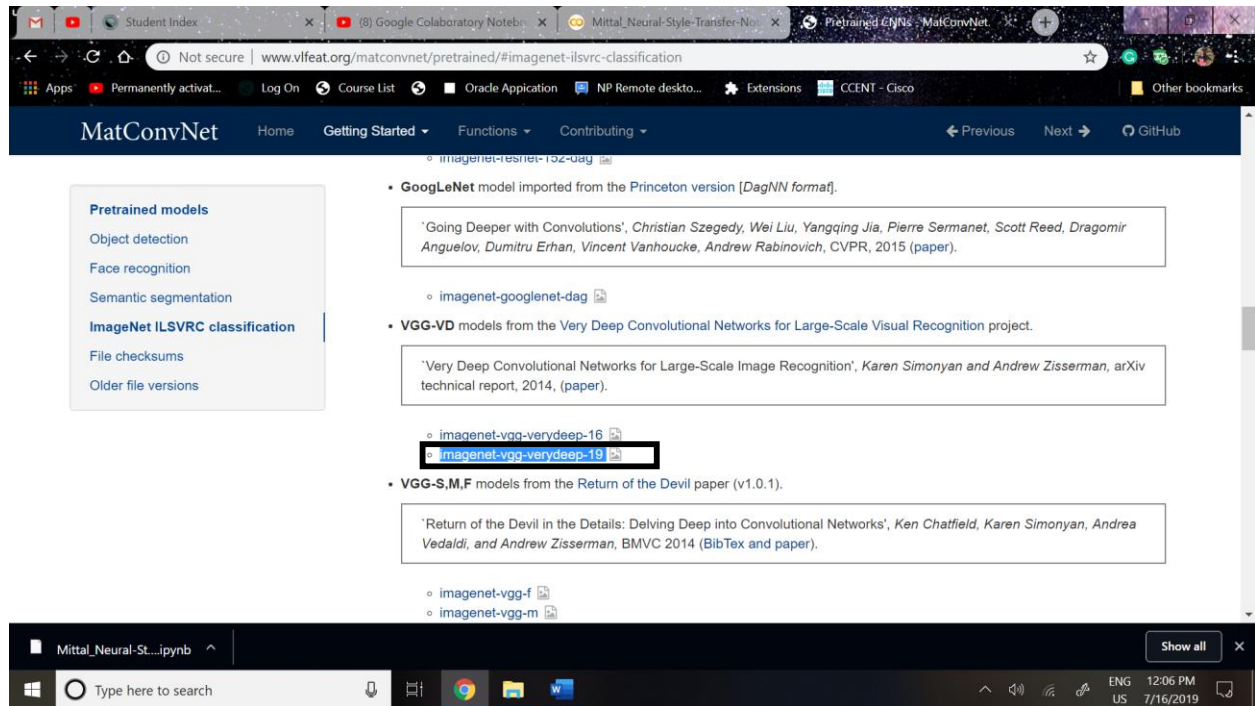
228 commits 1 branch 0 releases 10 contributors GPL-3.0

Branch: master New pull request Create new file Upload files Find File Clone or download

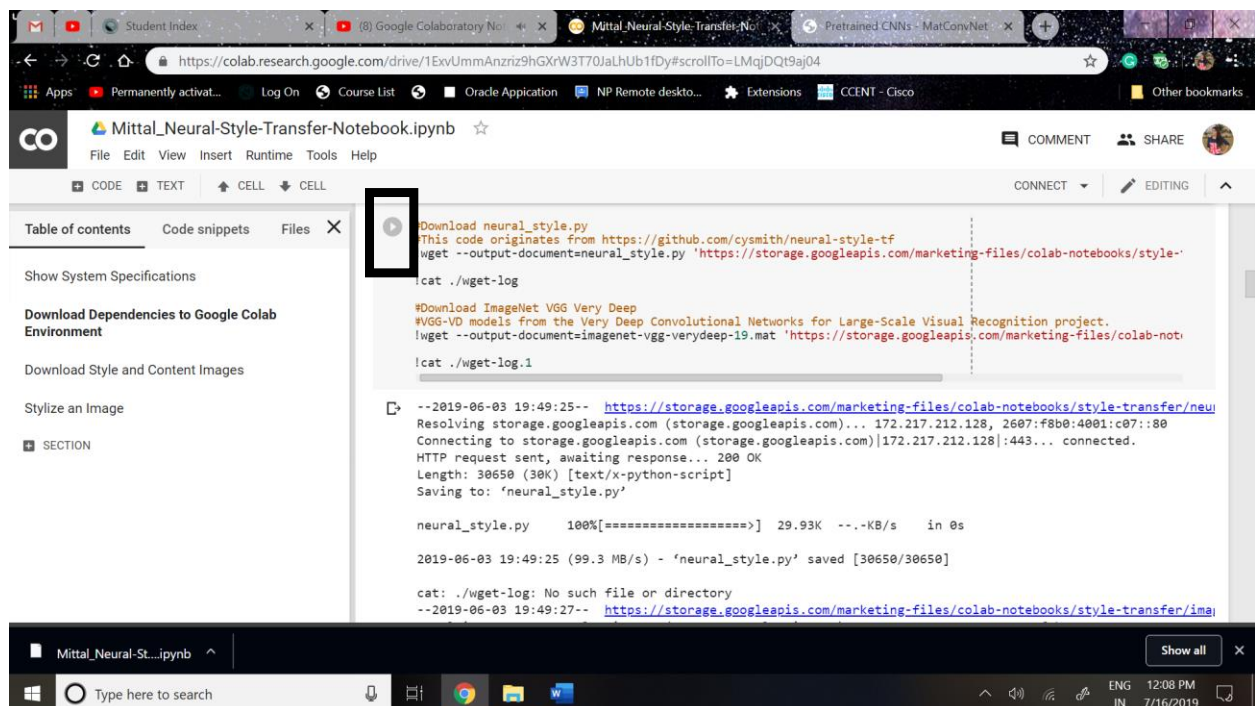
cysmith Merge pull request #71 from frank83413/master Latest commit a2c374f on Aug 3, 2018

File	Commit Message	Time Ago
examples	add video	2 years ago
image_input	Added taj mahal image	3 years ago
styles	Added default style and content images and directories	3 years ago
video_input	removed .cpp and .h files	3 years ago
LICENSE	First commit	3 years ago
README.md	Update README.md	2 years ago
neural_style.py	Update neural_style.py	last year
stylize_image.sh	Added verbose to usage script	3 years ago
stylize_video.sh	Fixed conflict in README	3 years ago

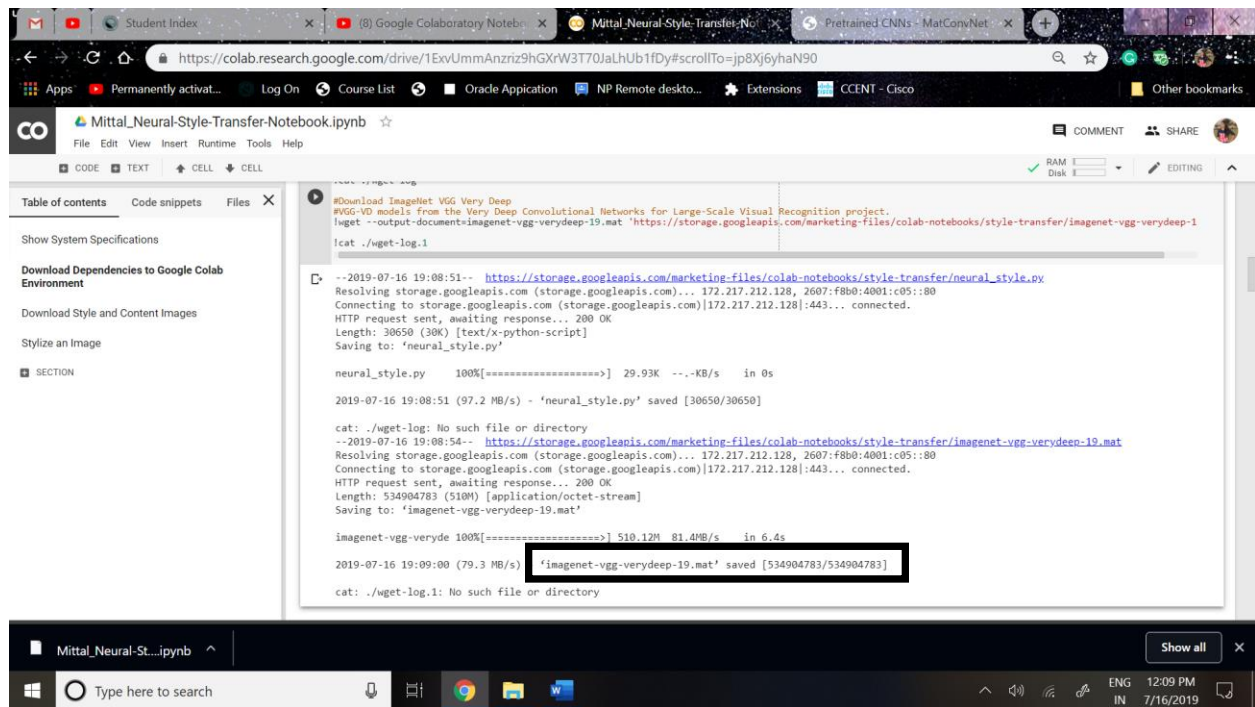
2. ImageNet Provide by MatConvNet (<http://www.vlfeat.org/matconvnet/pretrained/#imagenet-ilsvrc-classification>) and we use **imagenet-vgg-verydeep-19**.



Step 9: Run the code first



Result:



The screenshot shows a Google Colab notebook titled 'Mittal_Neural-Style-Transfer-Notebook.ipynb'. The terminal output displays the following commands and results:

```
#Download ImageNet VGG Very Deep
#VGG-VD models from the Very Deep Convolutional Networks for Large-Scale Visual Recognition project.
!wget --output-document=imagenet-vgg-verydeep-19.mat 'https://storage.googleapis.com/marketing-files/colab-notebooks/style-transfer/imagenet-vgg-verydeep-19.mat'
!cat ./wget-log.1

--2019-07-16 19:08:51-- https://storage.googleapis.com/marketing-files/colab-notebooks/style-transfer/imagenet-vgg-verydeep-19.mat
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.212.128, 2607:f8b0:4001:c05::80
Connecting to storage.googleapis.com (storage.googleapis.com)[172.217.212.128]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30650 (30K) [text/x-python-script]
Saving to: 'neural_style.py'

neural_style.py  100%[=====] 29.93K  --.-KB/s  in 0s

2019-07-16 19:08:51 (97.2 MB/s) - 'neural_style.py' saved [30650/30650]

cat: ./wget-log: No such file or directory

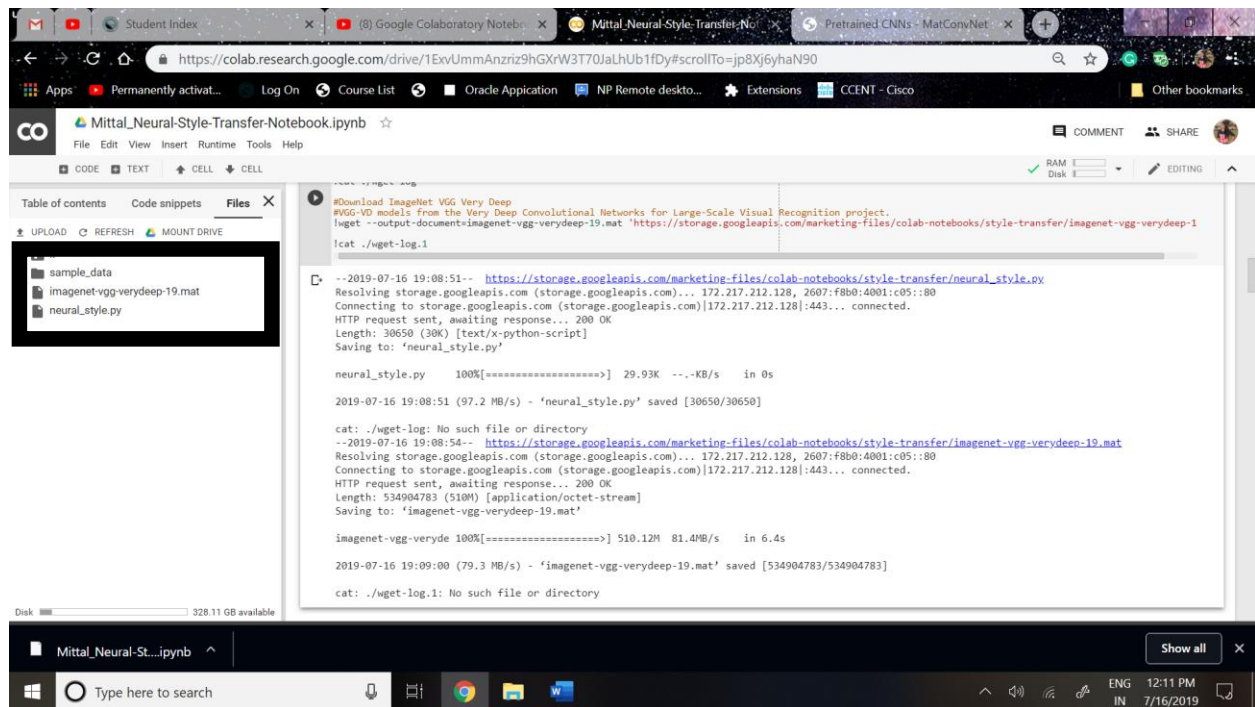
--2019-07-16 19:08:54-- https://storage.googleapis.com/marketing-files/colab-notebooks/style-transfer/imagenet-vgg-verydeep-19.mat
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.212.128, 2607:f8b0:4001:c05::80
Connecting to storage.googleapis.com (storage.googleapis.com)[172.217.212.128]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 534904783 (510M) [application/octet-stream]
Saving to: 'imagenet-vgg-verydeep-19.mat'

imagenet-vgg-veryde 100%[=====] 510.12M  81.4MB/s  in 6.4s

2019-07-16 19:09:00 (79.3 MB/s) - 'imagenet-vgg-verydeep-19.mat' saved [534904783/534904783]

cat: ./wget-log.1: No such file or directory
```

Step 10: See the save files in file section



The screenshot shows the same Google Colab notebook, but with the 'Files' tab selected. The file section displays the following files:

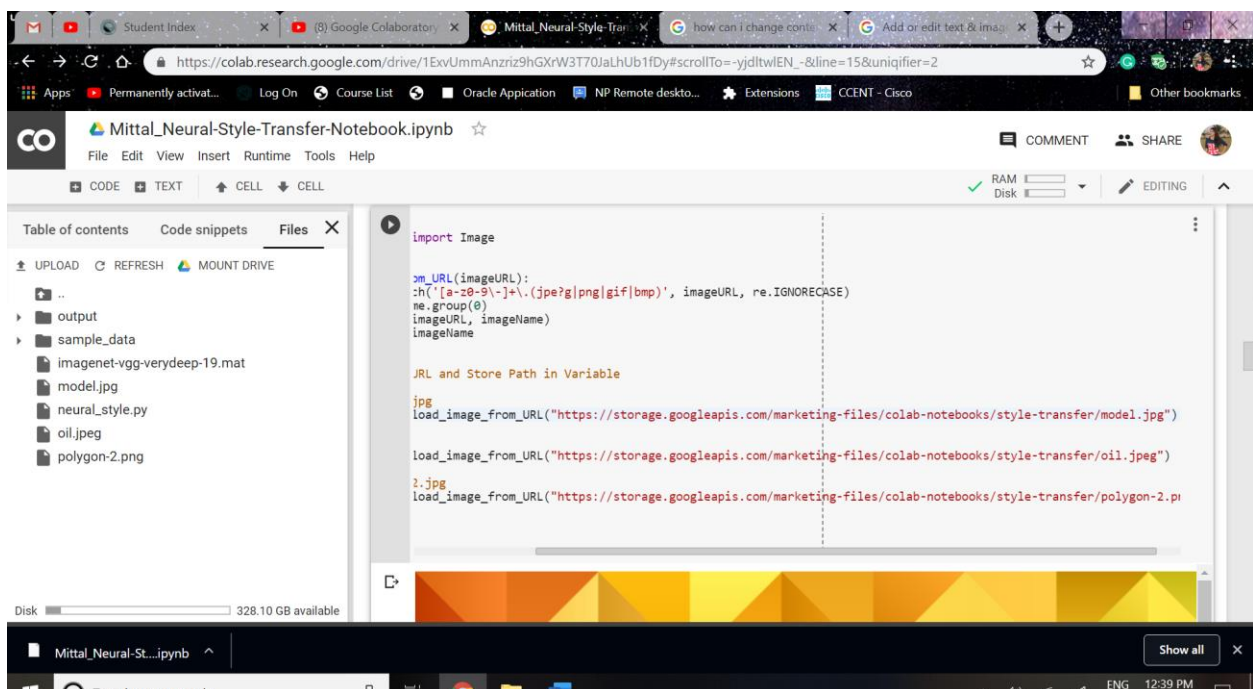
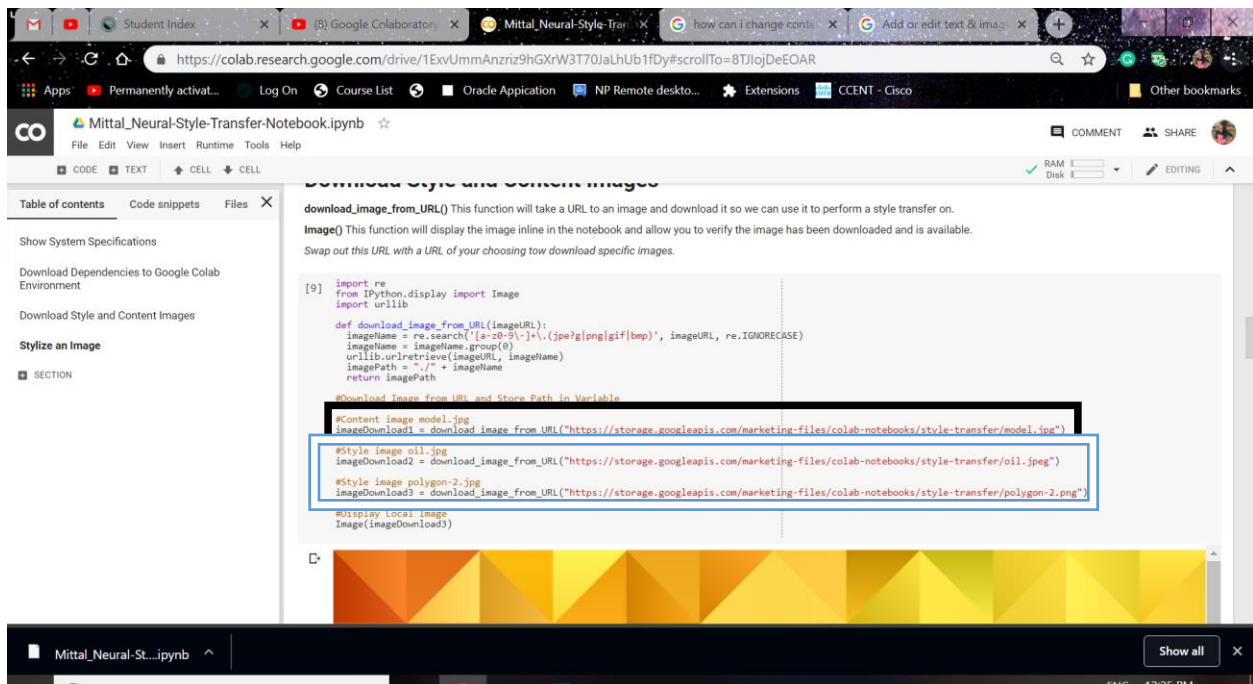
- sample_data
- imagenet-vgg-verydeep-19.mat
- neural_style.py

Step 11: Image Processing:

Here, Content Image is image on which we want to perform the style transfer (model.jpg is content image)

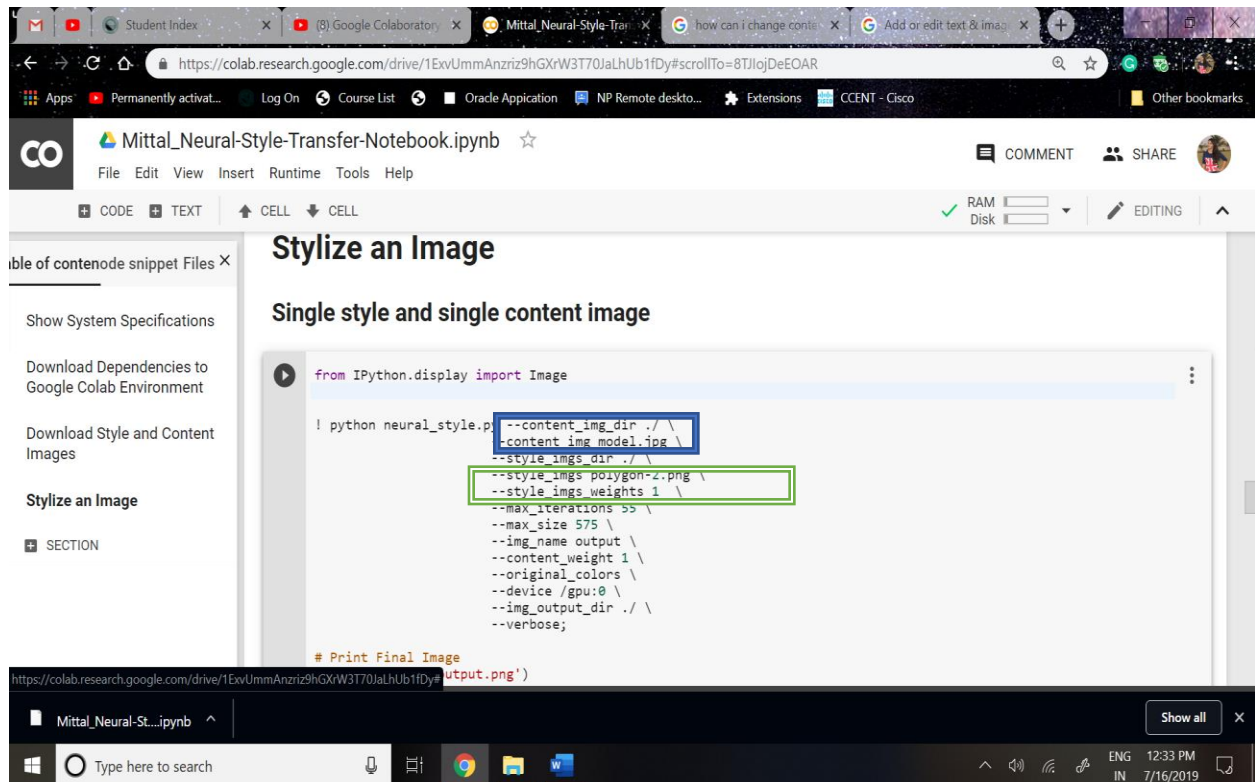
Style images are the images which we use to perform different style on content images. (polygon-2.png and oil.jpeg are the style image)

Run the Code and check all the images are saved in Files.



Step 12: Choose content image and style image

Here, I have taken model.jpg for content image and polygon-2.png for style image



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1ExvUmmAnzriz9hGxRW3T70JaLhUb1fDy#scrollTo=8TJlojDeEOAR>. The notebook title is "Mittal_Neural-Style-Transfer-Notebook.ipynb". The left sidebar contains a "Table of contents" with the following items: "Show System Specifications", "Download Dependencies to Google Colab Environment", "Download Style and Content Images", "Style an Image", and "SECTION". The main area is titled "Style an Image" and contains a code cell with the following Python code:

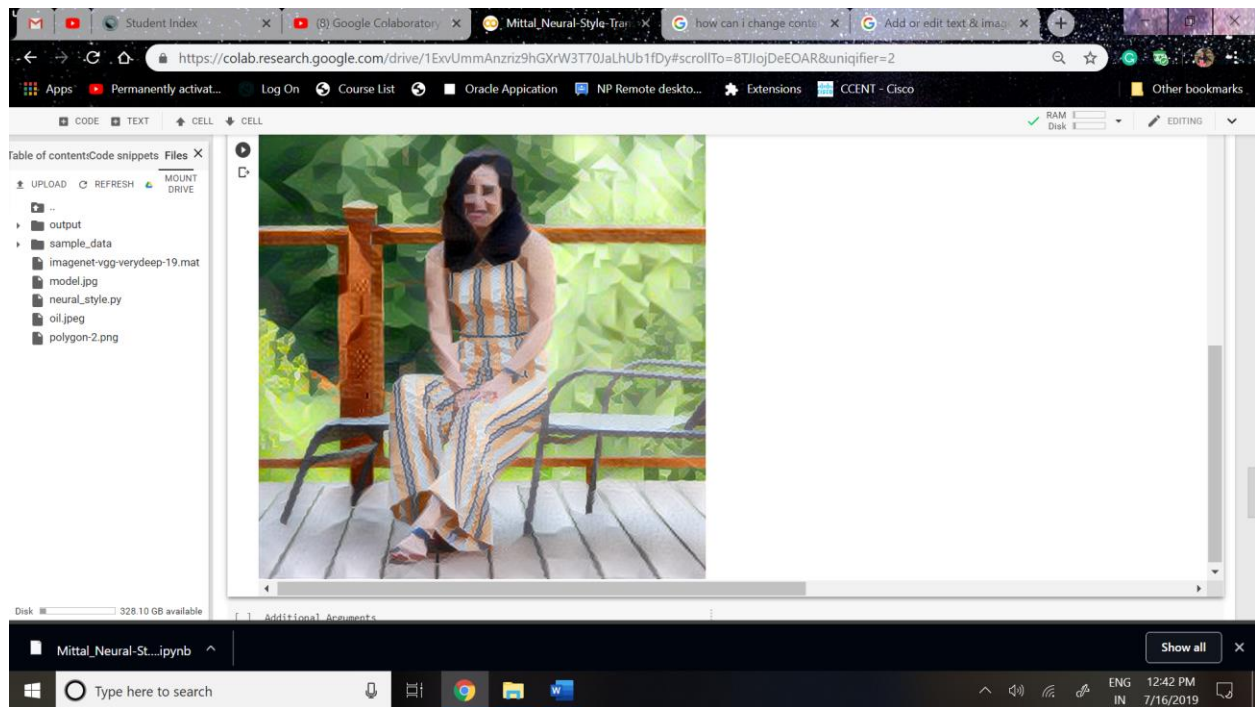
```
from IPython.display import Image

! python neural_style.py --content_img_dir ./ \
                        --content_img model.jpg \
                        --style_imgs_dir ./ \
                        --style_imgs polygon-2.png \
                        --style_imgs_weights 1 \
                        --max_iterations 55 \
                        --max_size 575 \
                        --img_name output \
                        --content_weight 1 \
                        --original_colors \
                        --device /gpu:0 \
                        --img_output_dir ./ \
                        --verbose;

# Print Final Image
Image(filename='output.png')
```

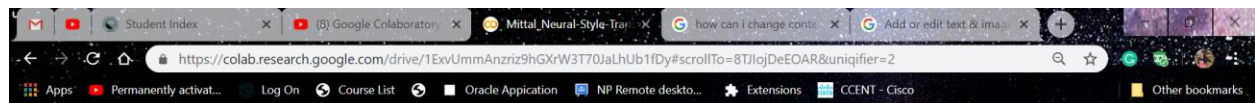
The code cell is highlighted with a blue border. The output of the code cell is not visible in the screenshot. The bottom status bar shows the time as 12:33 PM on 7/16/2019.

Step 13: Run the Code and See the Output



Extra: If I change the style image to oil.jpeg the output is different as shown in below image

Here I change style image to oil.jpeg



le style and single content image

```
from IPython.display import Image
```

```
! python neural_style.py --content_img_dir ./ \
  --content_img model.jpg \
  --style_imgs_dir ./ \
  --style_imgs oil.jpeg \
  --style_imgs_weights 1 \
  --max_iterations 55 \
  --max_size 575 \
  --img_name output \
  --content_weight 1 \
  --original_colors \
  --device /gpu:0 \
  --img_output_dir ./ \
  --verbose;
```

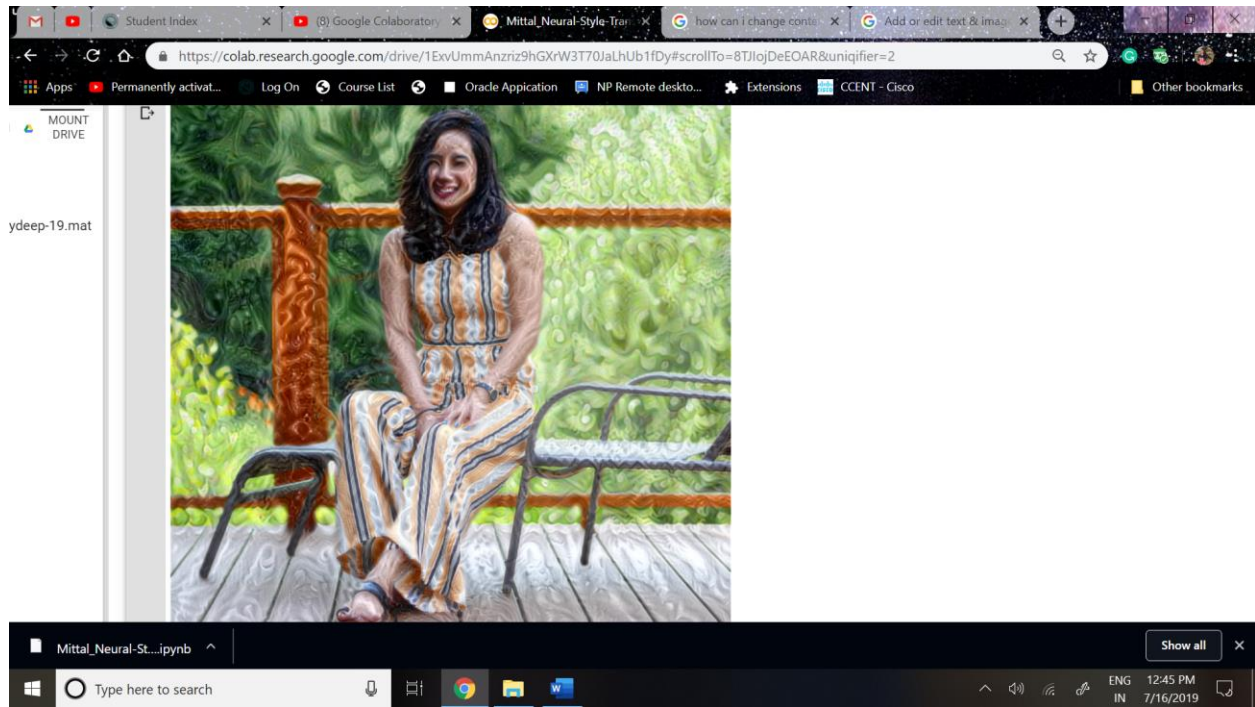
```
# Print Final Image
Image('./output/output.png')
```

WARNING: Logging before flag parsing goes to stderr.

0716 19:41:20.295516 140290526848896 deprecation_wrapper.py:119] From neural_style.py:550: The name tf.Session is deprecated



Output:



Q2. Evaluation of models for predicting Loan Application based on accuracy

Age	Salary (K)	Approved	Distance (48,140)	Distance (60,60)	Distance (34,20)	Distance (25,20)	Distance (37,80)
25	40	N	103	40	22	20	42
35	60	N	81	25	40	41	20
45	80	N	60	25	61	63	8
20	20	N	123	57	14	5	62
35	120	N	24	65	100	100	40
52	18	N	122	43	18	27	64
23	95	Y	51	51	76	75	21
40	62	Y	78	20	42	45	18
60	100	Y	42	40	84	87	30
48	220	Y	80	160	200	201	140
33	150	Y	18	94	130	130	70
47	38	N	102	26	22	28	43
24	88	Y	57	46	69	68	15
37	62	Y	79	23	42	44	18
60	56	N	85	4	44	50	33
48	22	N	118	40	14	23	59
33	65	Y	76	27	45	46	16

For (48,140)

Calculate the Distance from each point i.e. (25,40), (35, 60) ... from (48, 140)

Here K=3 and K=5 means We need to choose 3 small Number for **Model 1** and 5 number for **Model 2**

Here We have choose 3 small Number for Model 1

18, 24, 42

18	Y
24	N
42	Y

Here We have Majority 'Y', So Prediction for K=3 for (48,140) is 'Y'.

Repeat Same Process for K=5 and other Data Given

Age	Salary (K)	Approved	Predicted (Y/N) for K=3	Predicted (Y/N) for K=5
48	140	Y	Y	Y
60	60	N	Y	N
34	20	N	N	N
25	20	N	N	N
37	80	Y	Y	Y

Confusion Matrix for K=3

	Y	N
Y	2	1
N	0	2

Confusion Matrix for K=5

	Y	N
Y	2	0
N	0	3

So, K=5 is more accurate than K=3