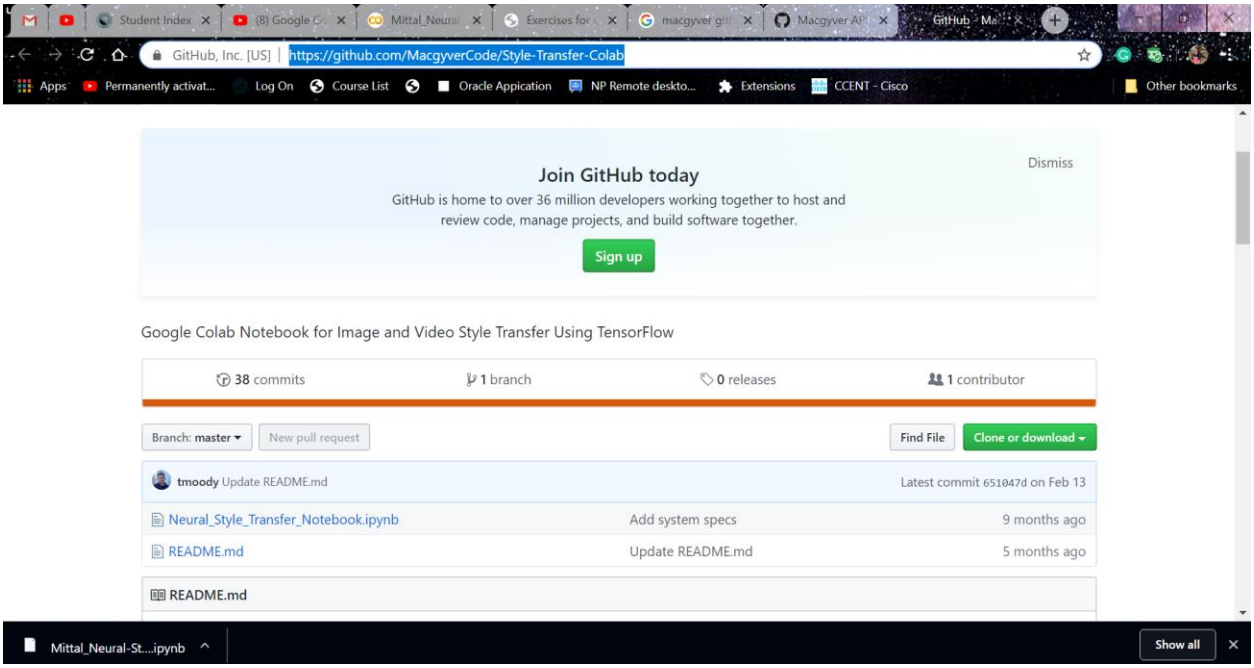
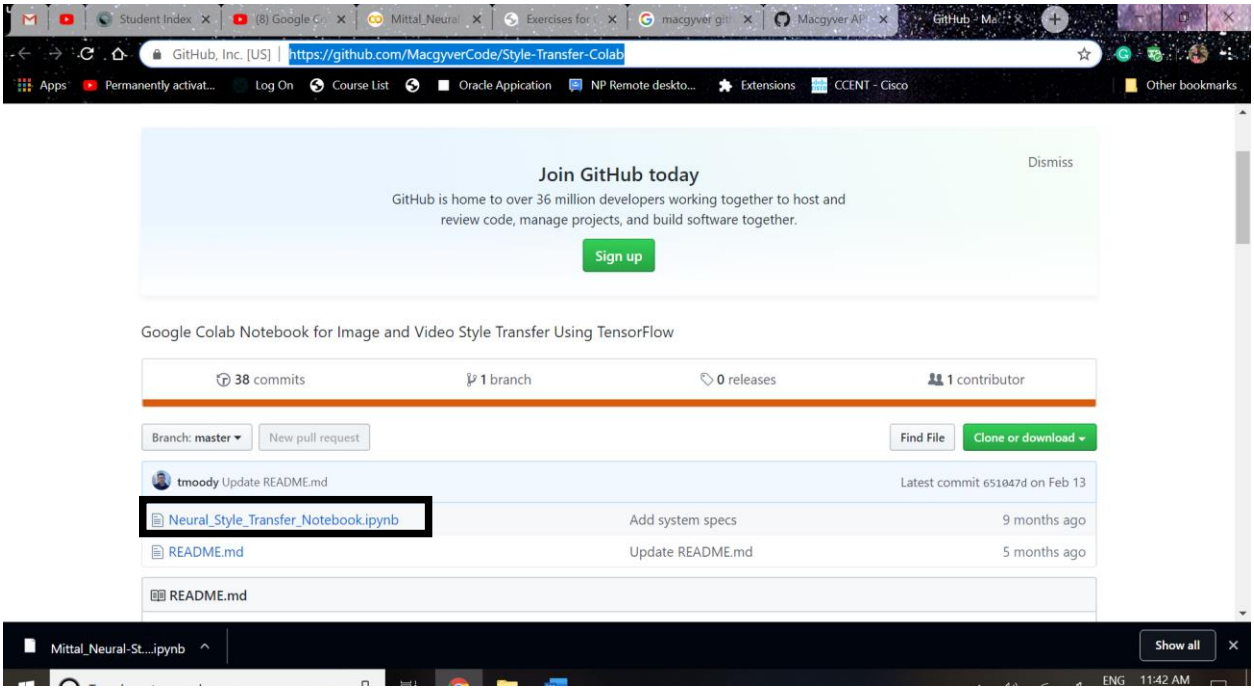
Google Colaboratory Notebook Tutorial with GPU

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

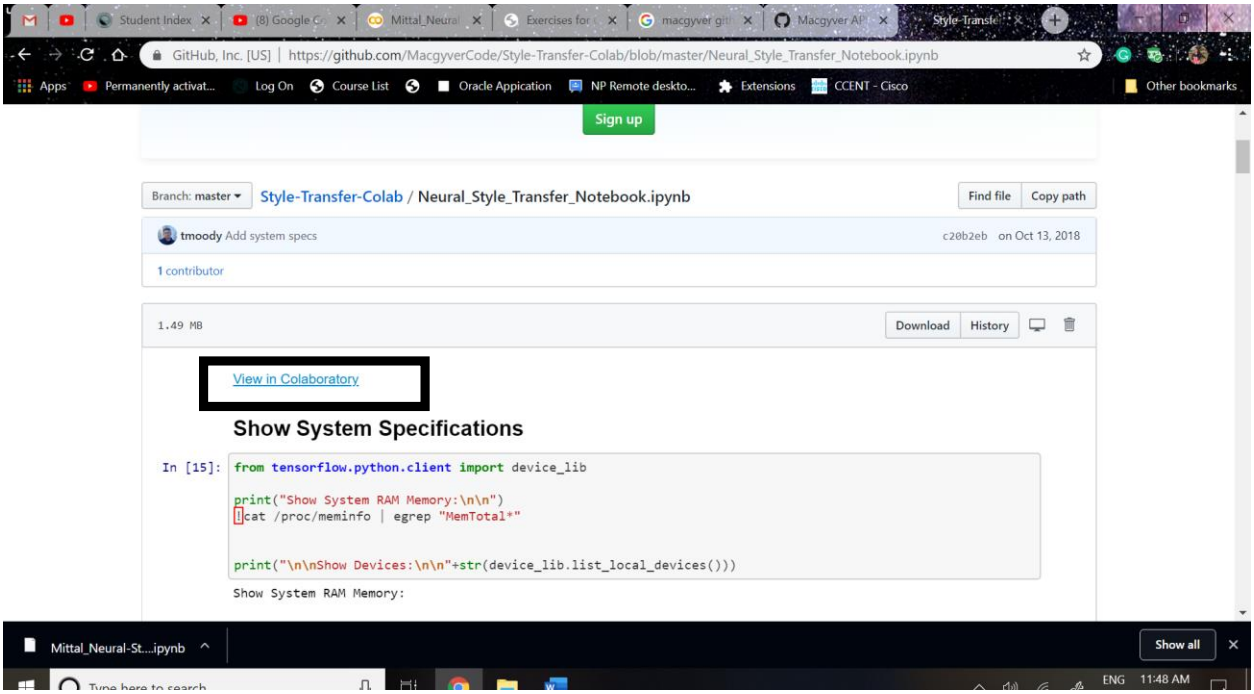
Step 2: Go to Style-Transfer-Colab Repository (https://github.com/MacgyverCode/Style-Transfer-Colab)



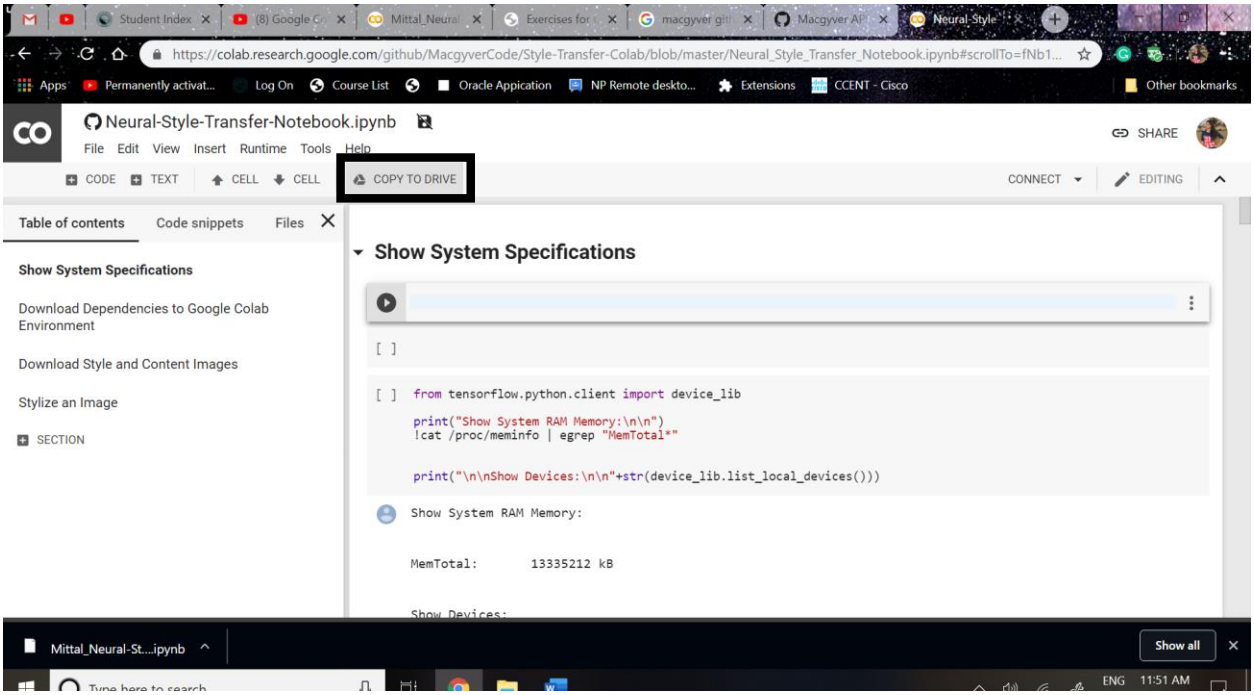
Step 3: Open Neural_Style_Transfer_Notebook



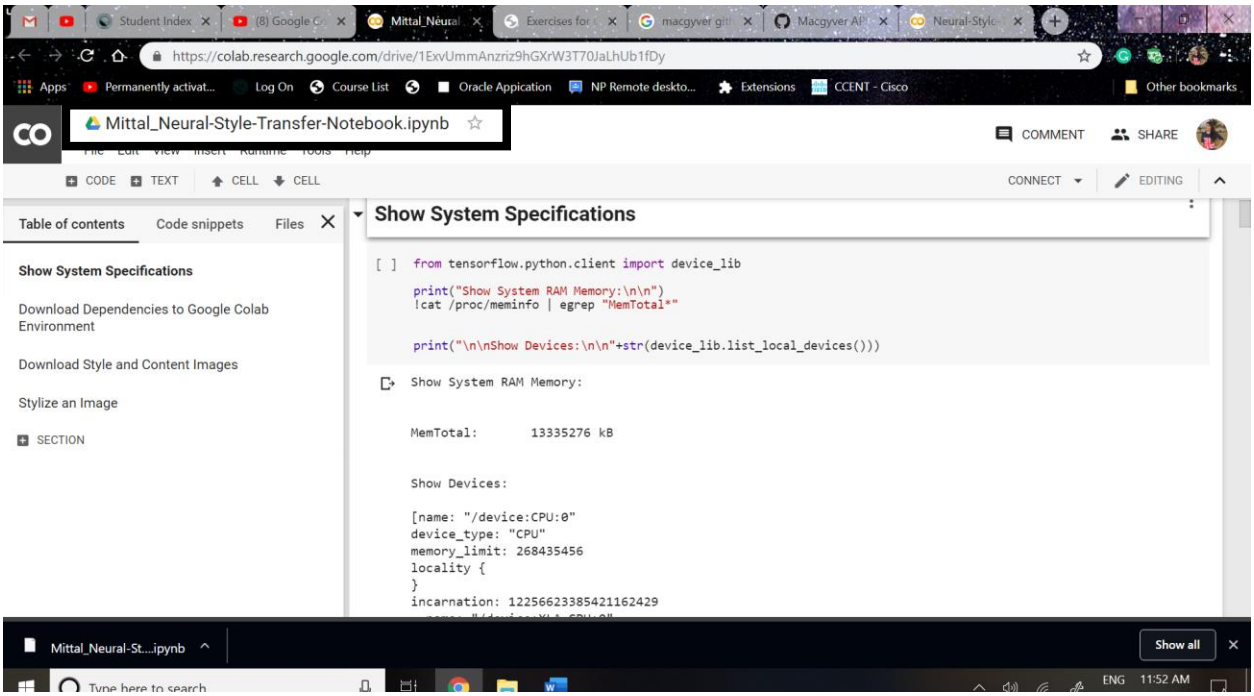
Step 4: Open Program in Google Colaboratory



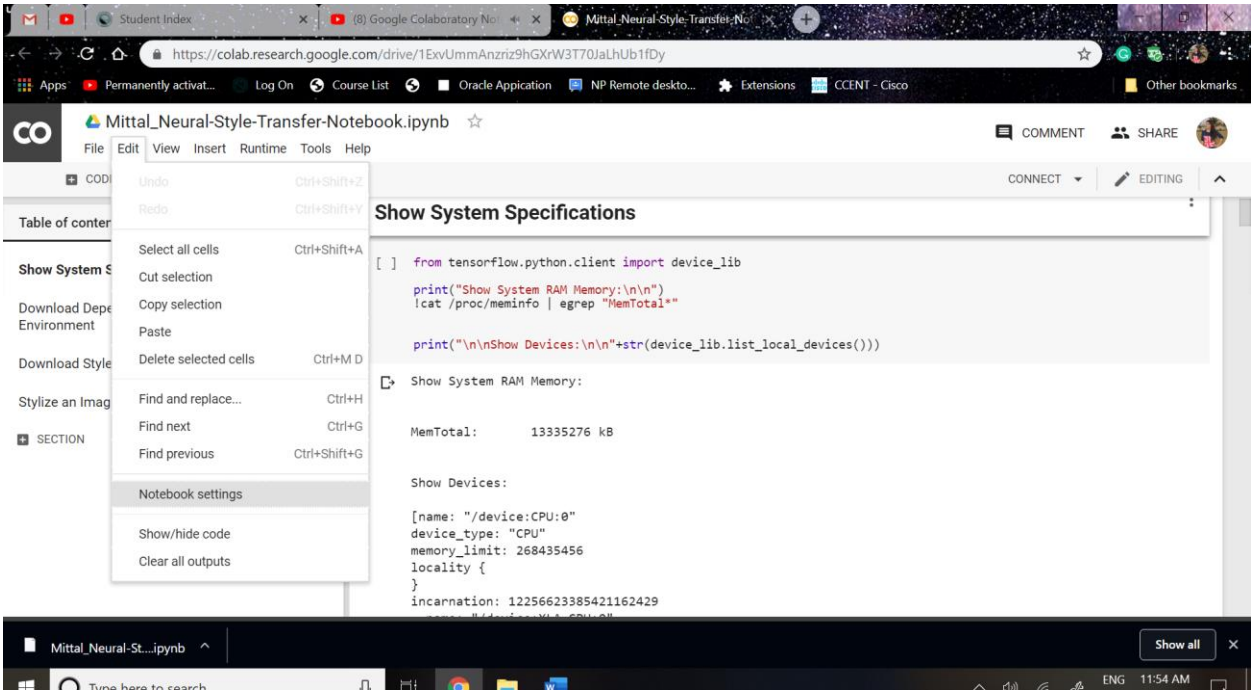
Step 5: Copy code in Google Drive

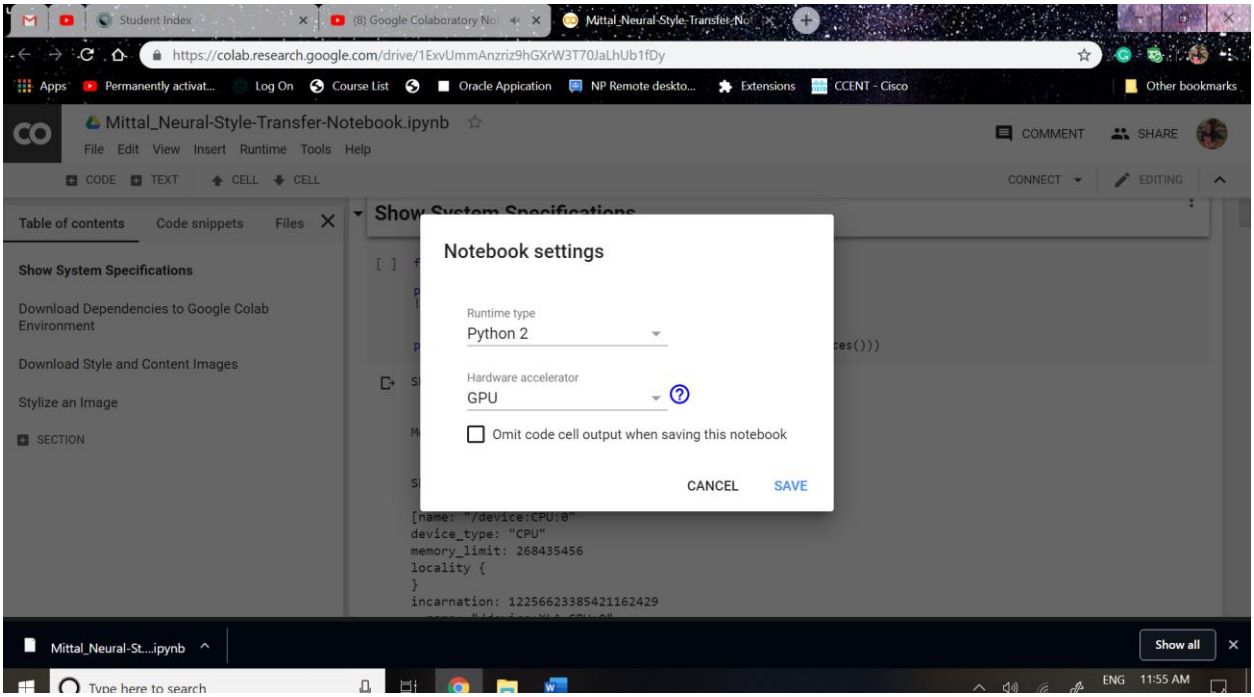


Step 6: Open from Google Drive and change the Name

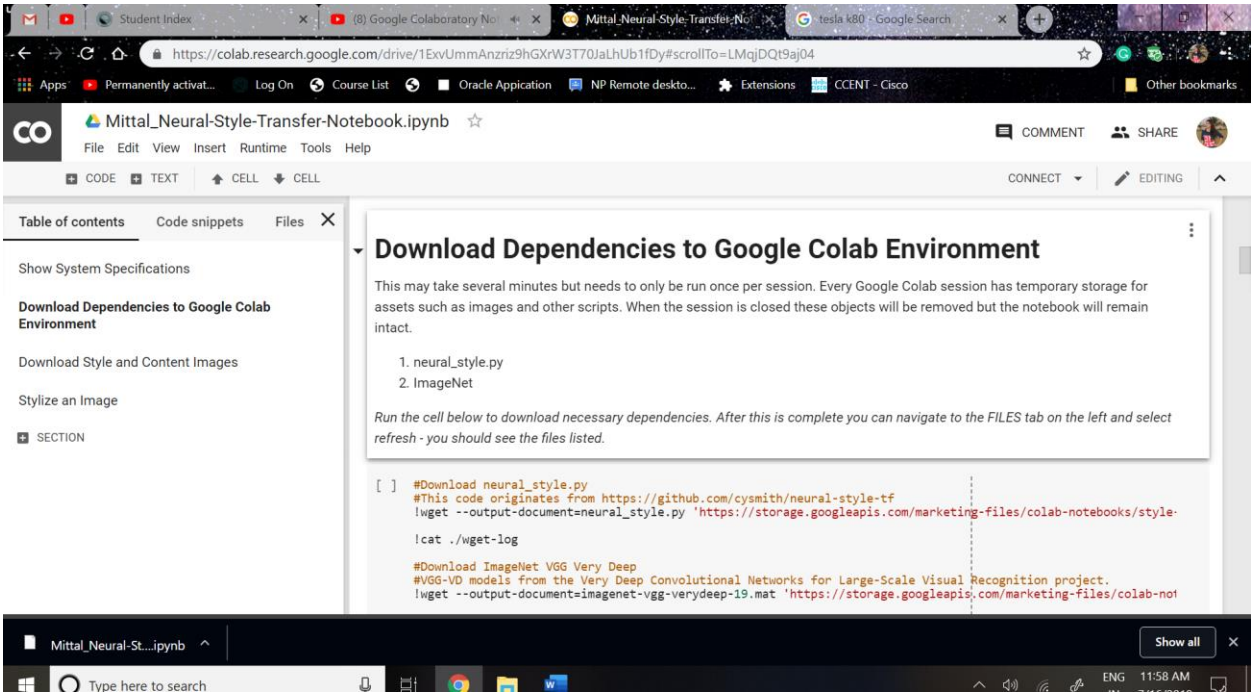


Step 7: Setup Environment for Notebook

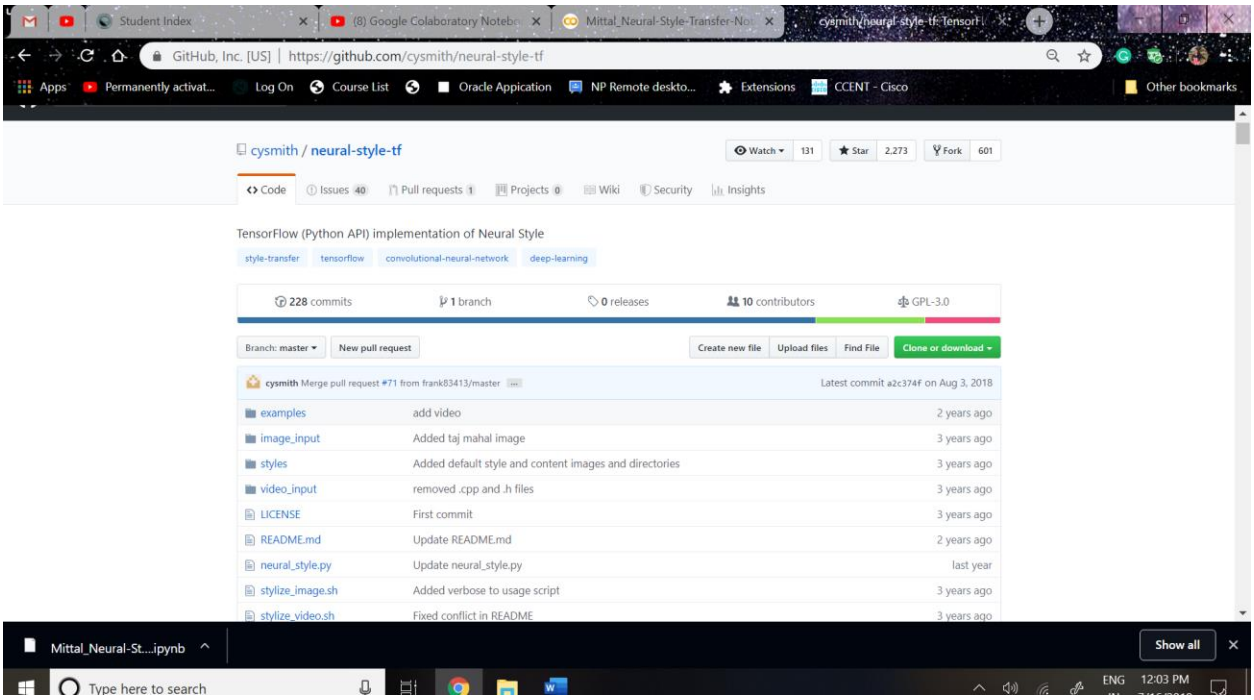




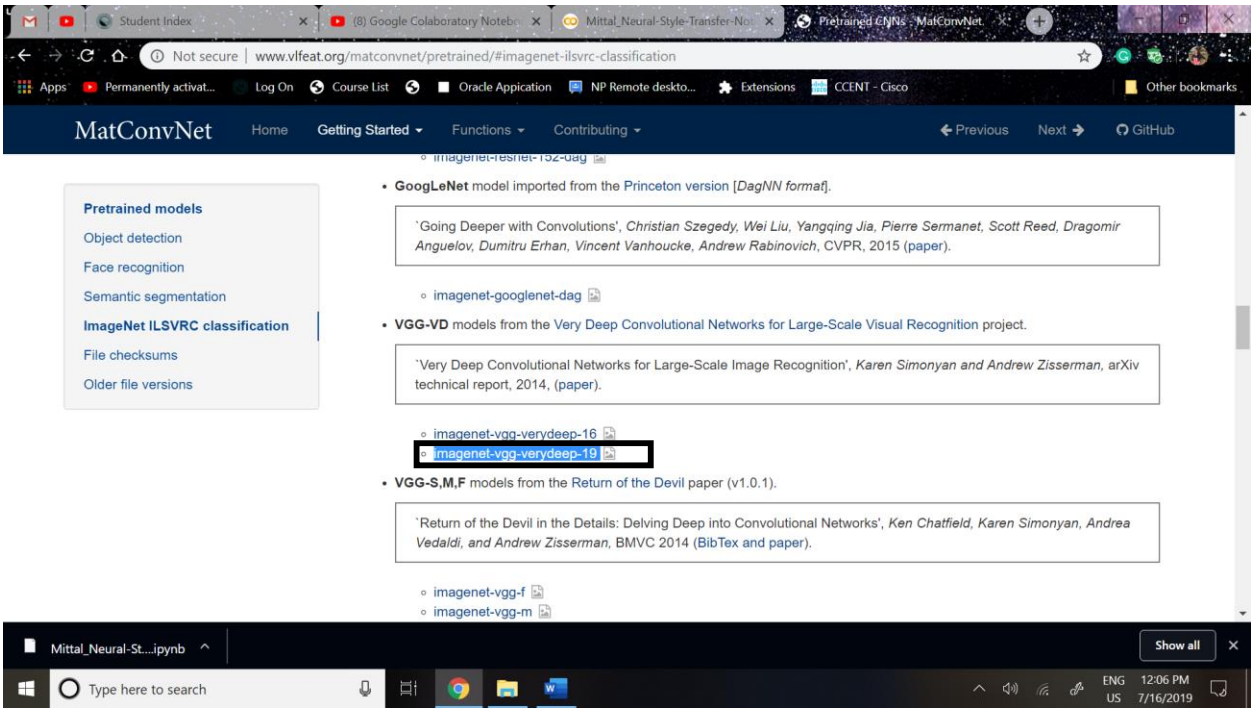
Step 8: There are two dependencies 1. neural_style.py (Python Code) and 2. ImageNet



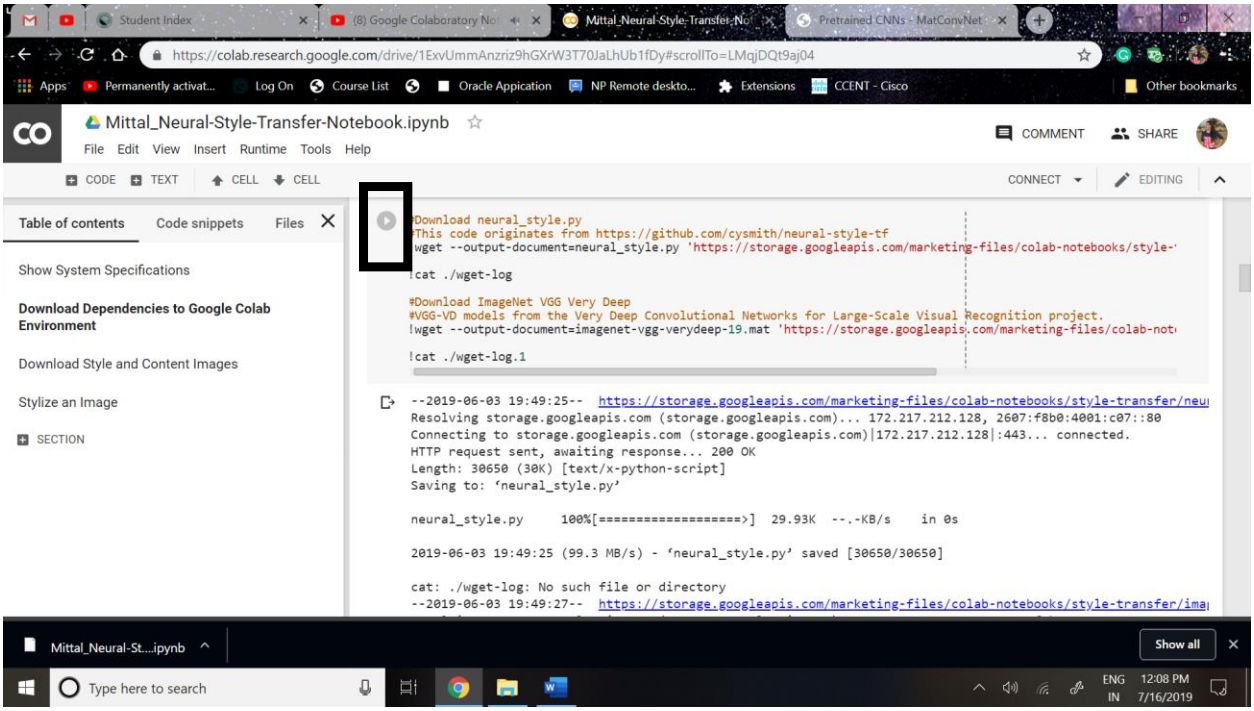
1. neural_style.py (Python Code): taken from the https://github.com/cysmith/neural-style-tf .



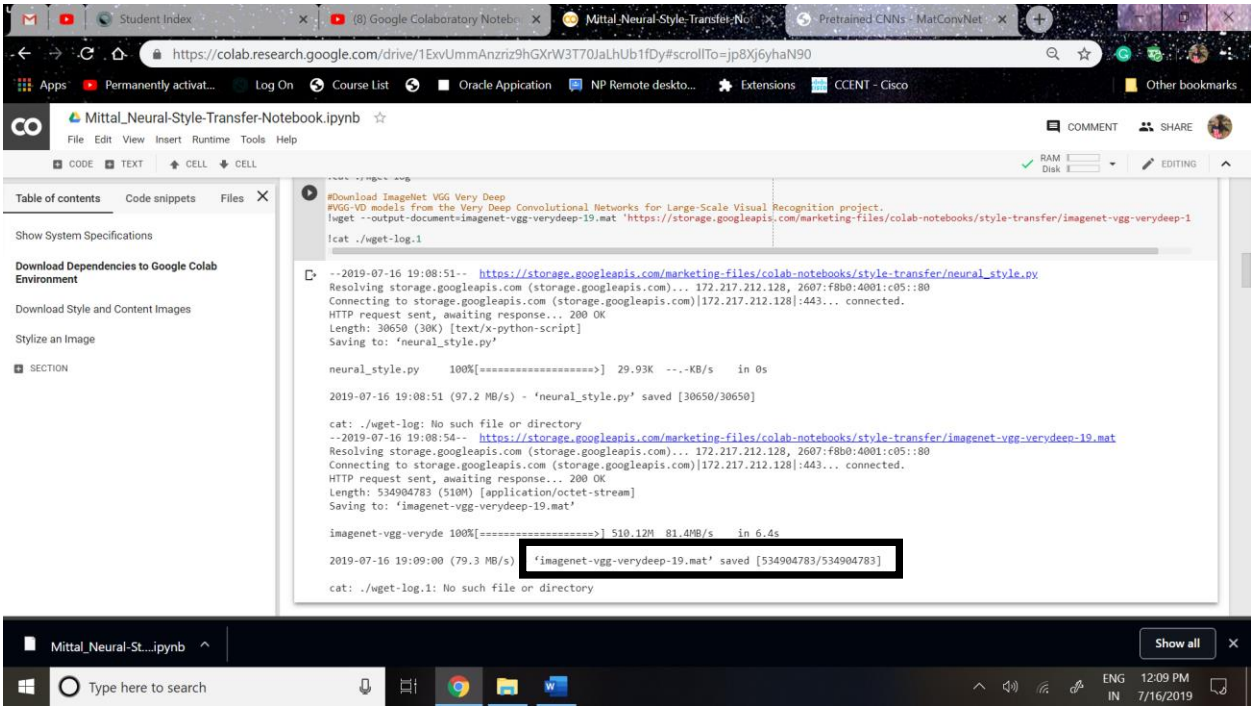
2. ImageNet Provide by MatConvNet (http://www.vlfeat.org/matconvnet/pretrained/#imagenet-ilsvrc-classification) and we use **imagenet-vgg-verydeep-19.**



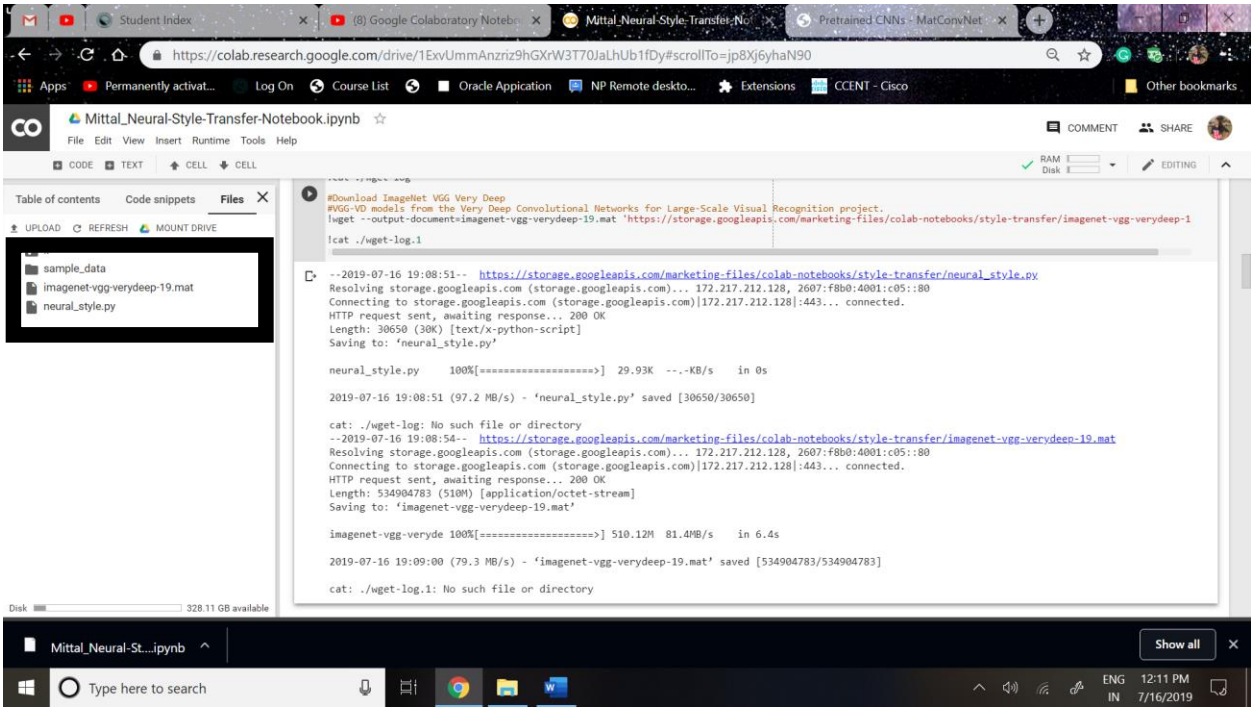
Step 9: Run the code first



Result:



Step 10: See the save files in file section

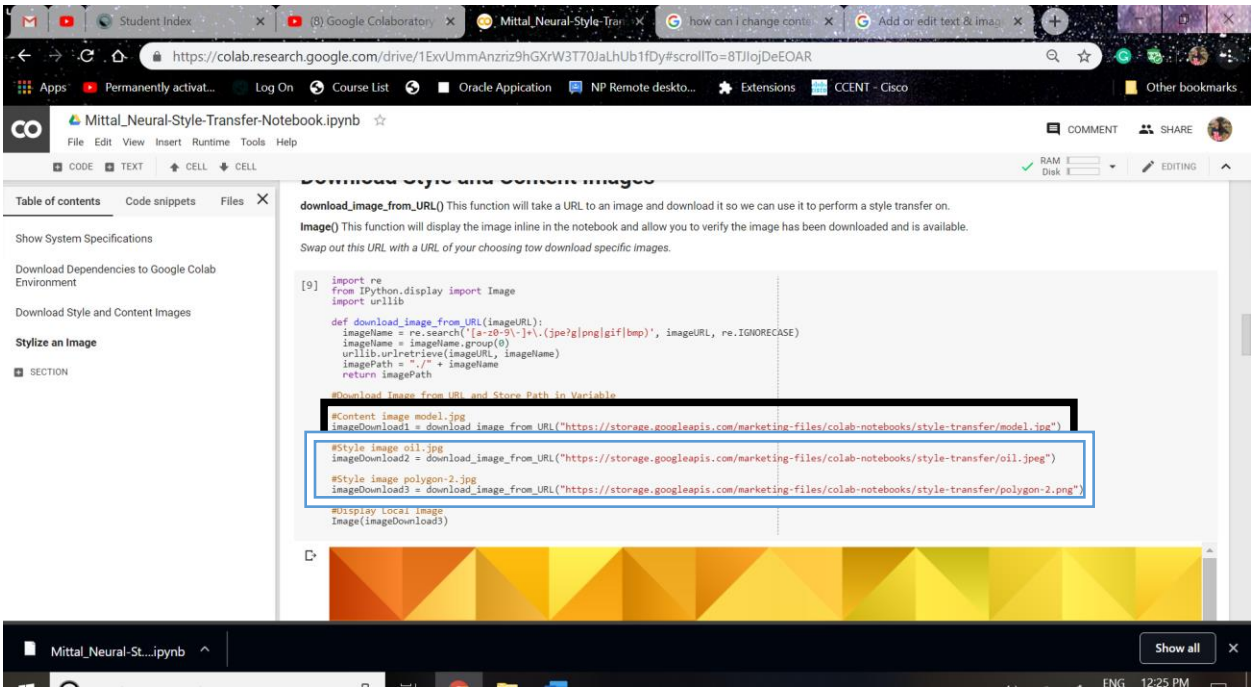


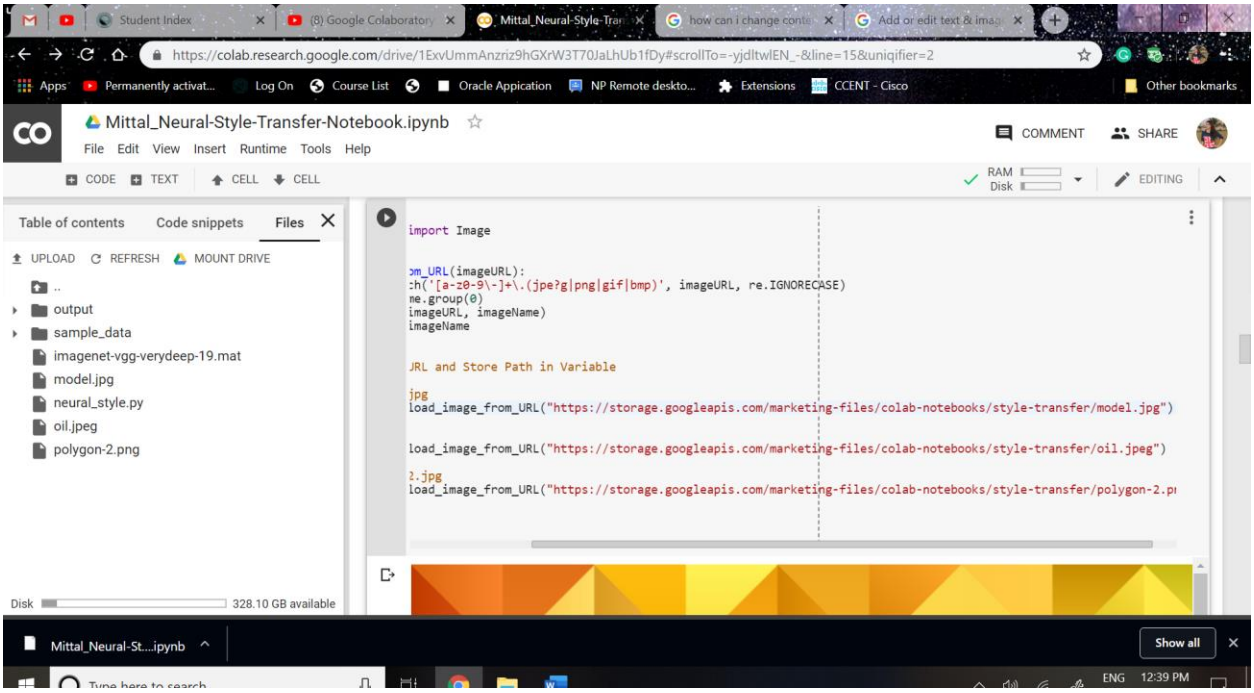
Step 11: Image Processing:

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)

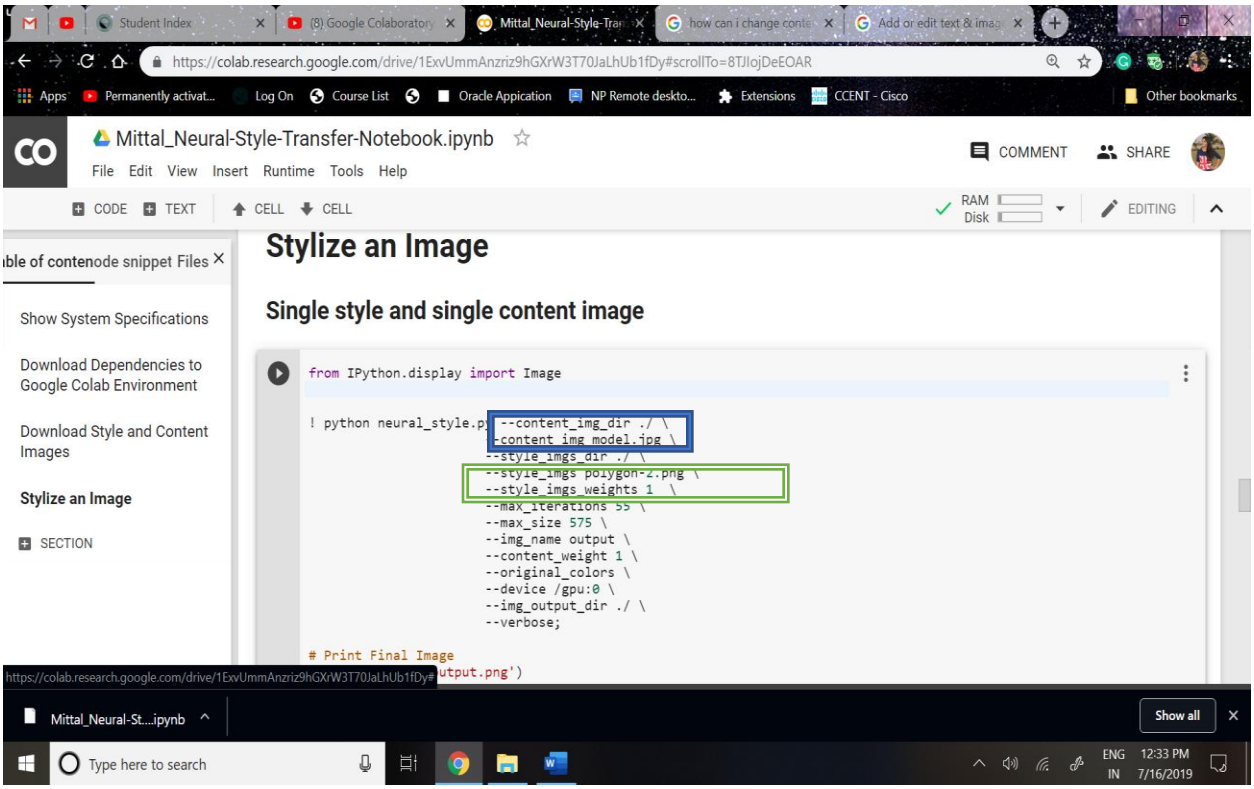
Run the Code and check all the images are saved in Files.



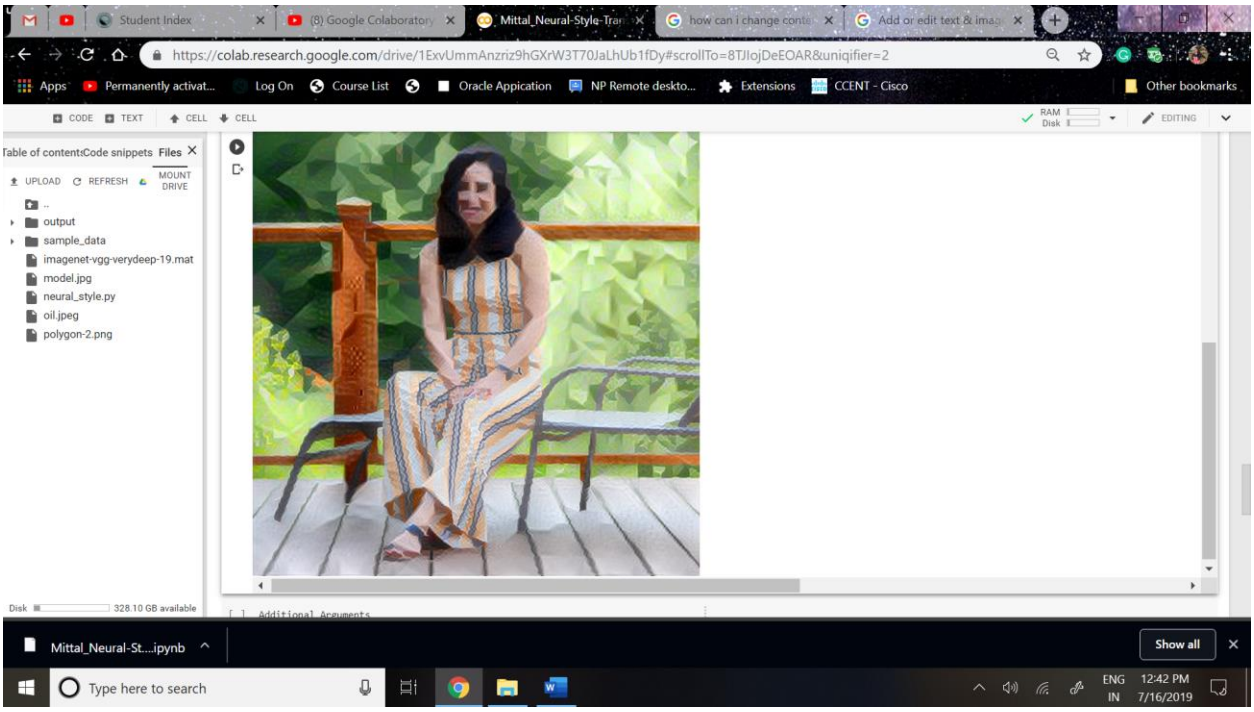


Step 12: Choose content image and style image

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

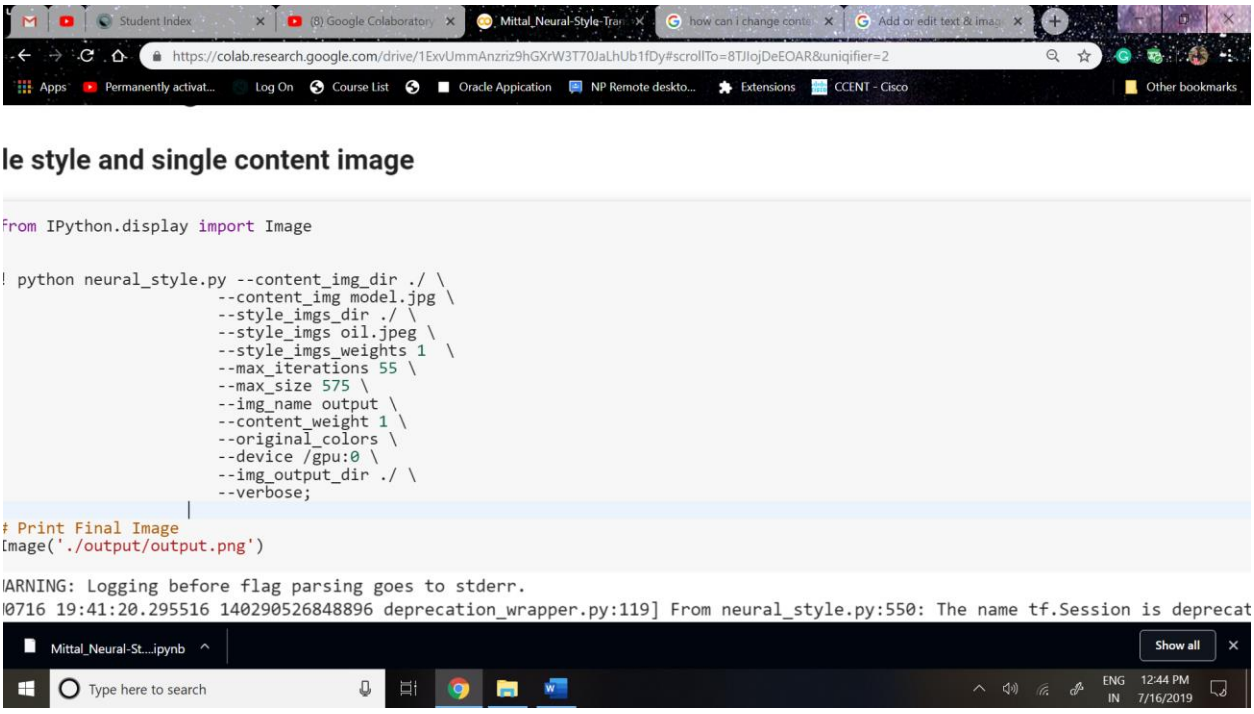


Step 13: Run the Code and See the Output

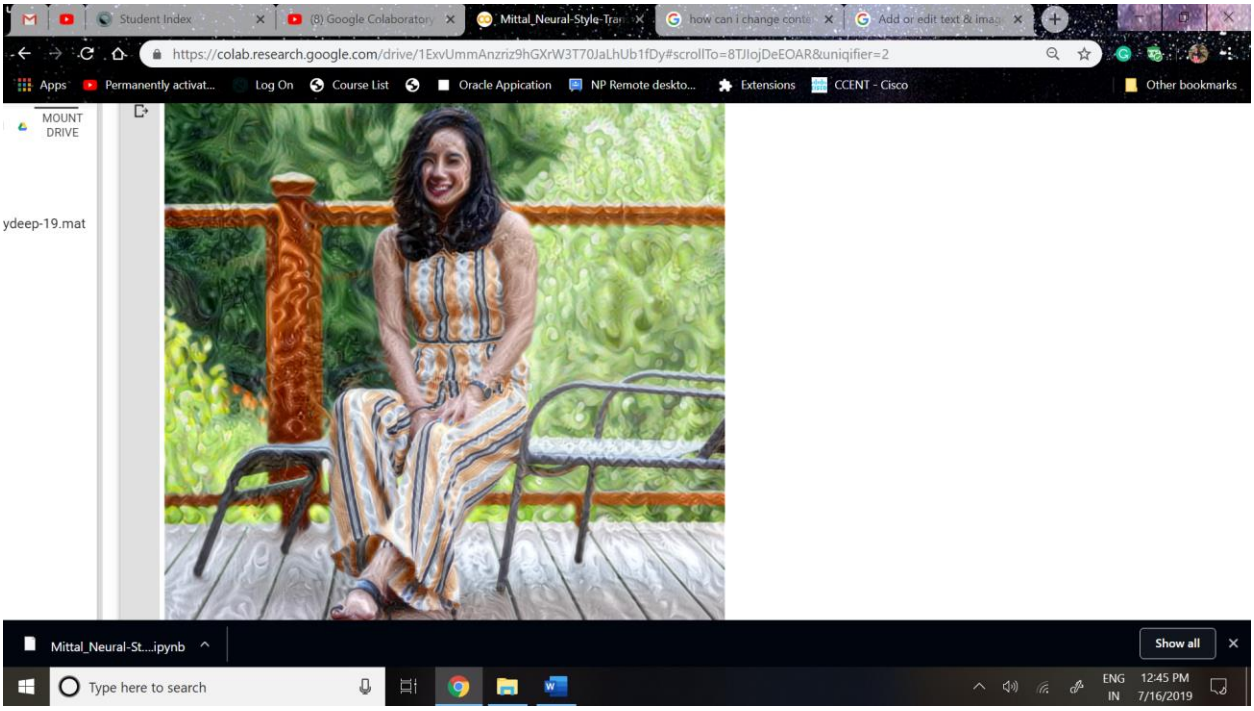


Extra: If I change the style image to oil.jpeg the output is different as shown in below image

Here I change style image to oil.jpeg



Output:


Student Index
(8) Google Colaboratory
Mittal_Neural-Style-Tra
how can i change conte
Add or edit text & imag
https://colab.research.google.com/drive/1ExvUmmAnzriz9hGXrW3T70JaLhUb1fDy#scrollTo=8TJIojDeEOAR&uniqifier=2
Apps
Permanently activat...
Log On
Course List
Oracle Appication
NP Remote deskto...
Extensions
CCENT - Cisco
Other bookmarks
MOUNT DRIVE
ydeep-19.mat
Mittal_Neural-St....ipynb
Show all
Type here to search
ENG
IN
12:45 PM
7/16/2019

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