DURHAM COLLEGE
SUCCESS MATTERS

Model Evaluation for Neural Style Transfer

Project Title	:	Neural Style Transfer
Course Name	:	Capstone II
Course Facilitator	:	Marcos Bittencourt
Group Number	:	Group - 9
Student Names and IDs	:	1) Dilsher Singh – 100768315
		2) Rajan Walia – 100727112
		3) Fulya Caliskan– 100766627
		4) Peter Kusiak – 100399575
Date of Submission	:	February 21, 2020.

Model Evaluation for Neural Style Transfer

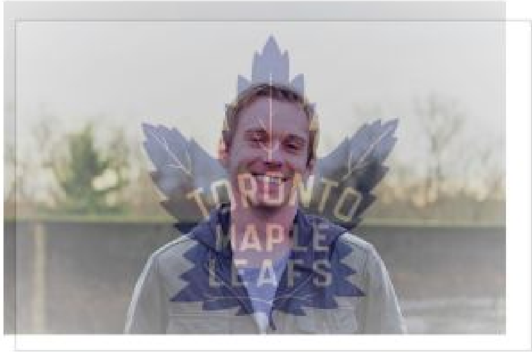
Background and Scope

The aim of this project is to merge the sport-like image as a background to the content image and apply a style transfer to the merged image. Below is the brief visualisation of the project:

Content-image

TORONTO
MAPLE
LEAFS

Sports image



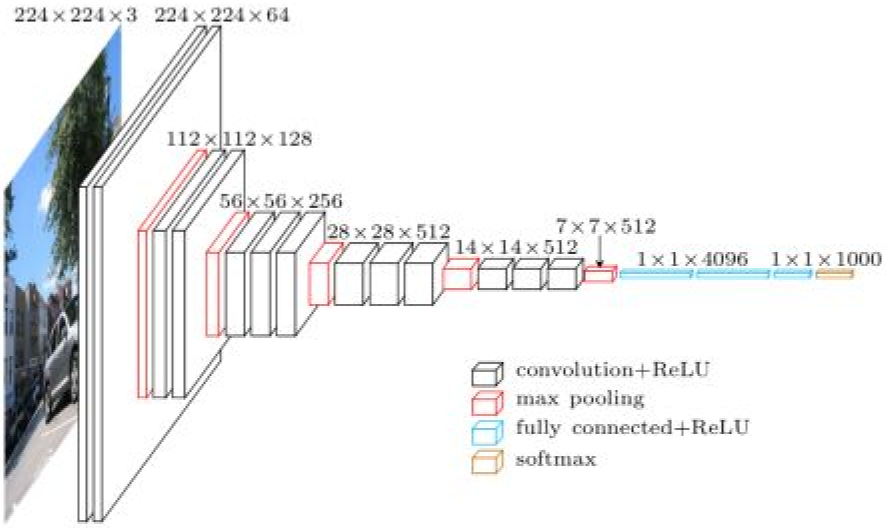
Merged images

Firstly the background of the content image will be removed, and a sports filter will be added behind it (such as the image of the Toronto Maple Leafs logo, or a famous NHL player's jersey/number). Then the image will be passed through Neural Style Transfer, where the image will be styled through merging an artistic image.

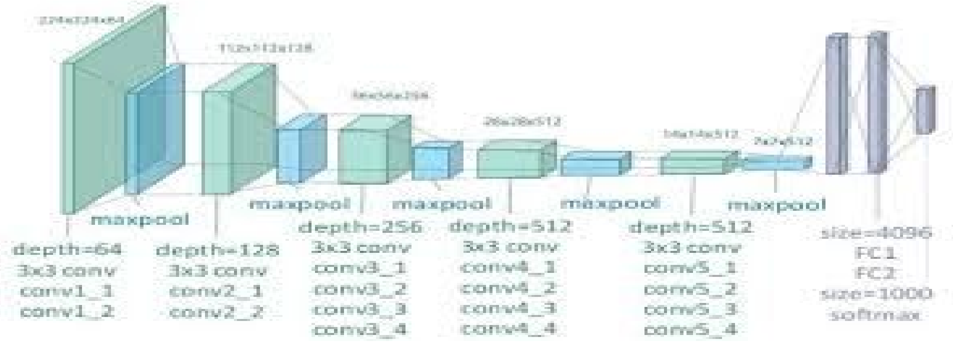
Pre-trained models

Using pre-trained models is one of the prerequisites for the project. Below are pre-trained models and examples of their architecture that may be used as a part of this exercise:

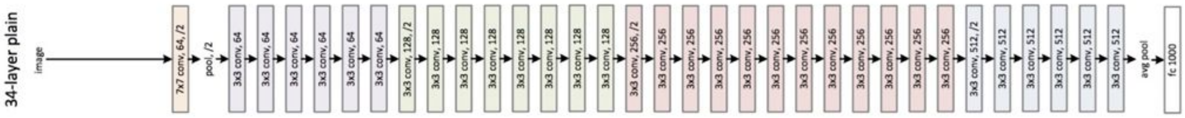
- VGG - 16



- VGG- 19



- ResNet



Model Evaluation

The evaluation is made on the basis of below key metrics -

Content Loss

The content loss is a function that describes the distance of content from our input image x and our content image:

$$L^l_{content}(p, x) = \sum_{i,j}(F^l_{ij}(x) - P^l_{ij}(p))^2$$

X represents any image. F^l_i□(x)∈ C□□(x)and P^l_i□(x) ∈ C□□(x) describe the respective intermediate feature representation of the network with inputs x and p at layer *l*.

Style Loss

The style loss is the mean squared distance between the feature correlation map of the style image and the input image:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j}(G^l_{ij} - A^l_{ij})^2$$

G^l_i□ and A^l_i□ are the respective style representation in layer *l* of input image *x* and style image *a*. Nl describes the number of feature maps, each of size Ml=height∗width.

The goal is to minimize the sum of Style Loss and Content Loss. The Gradient Descent algorithm as an optimization technique will be used to minimize the loss.

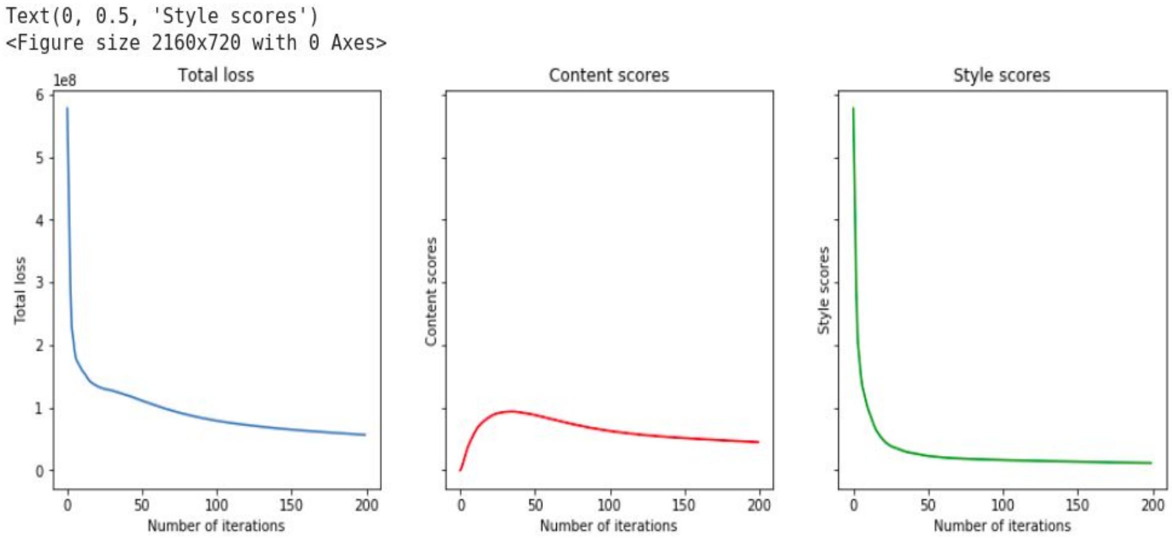
Loss Curves

Deep Learning is used only in the second part of the project (ie to style the image), hence the below part is only related to the second part. The two sample images used are the *“Green Sea Turtle grazing seagrass*” and “*The Great Wave off Kanagawa*”

The loss is minimized till 1000 iterations. For better visualisations, the learning curves are plotted only till 200 iterations. Below are the loss curves and the images generated for each of the pre-trained models used -

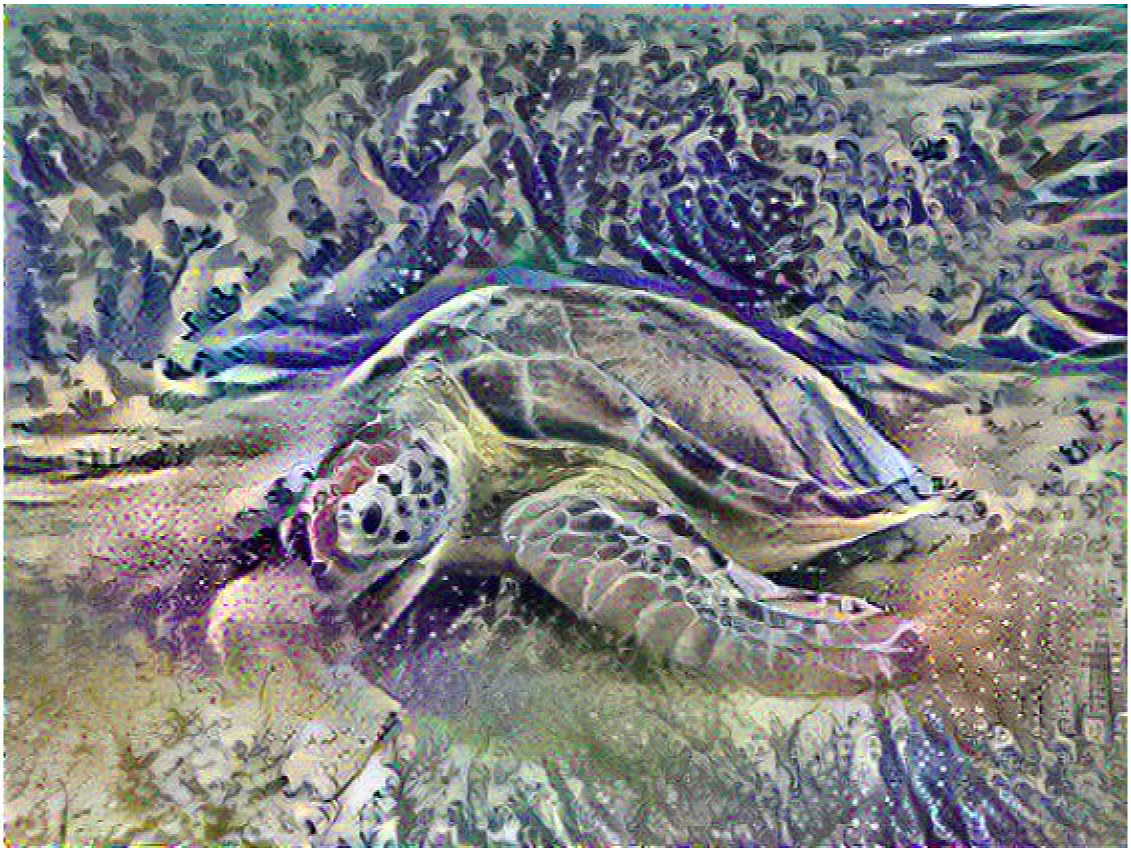
VGG - 19

Loss Curves-



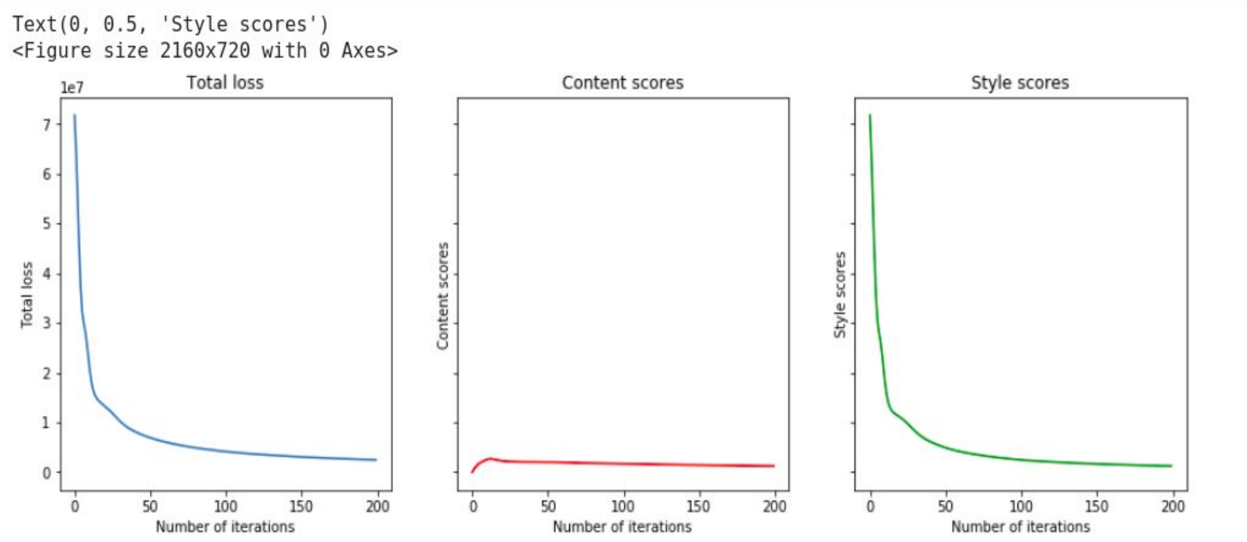
```
Text(0, 0.5, 'Style scores')
<Figure size 2160x720 with 0 Axes>
```

Image generated -



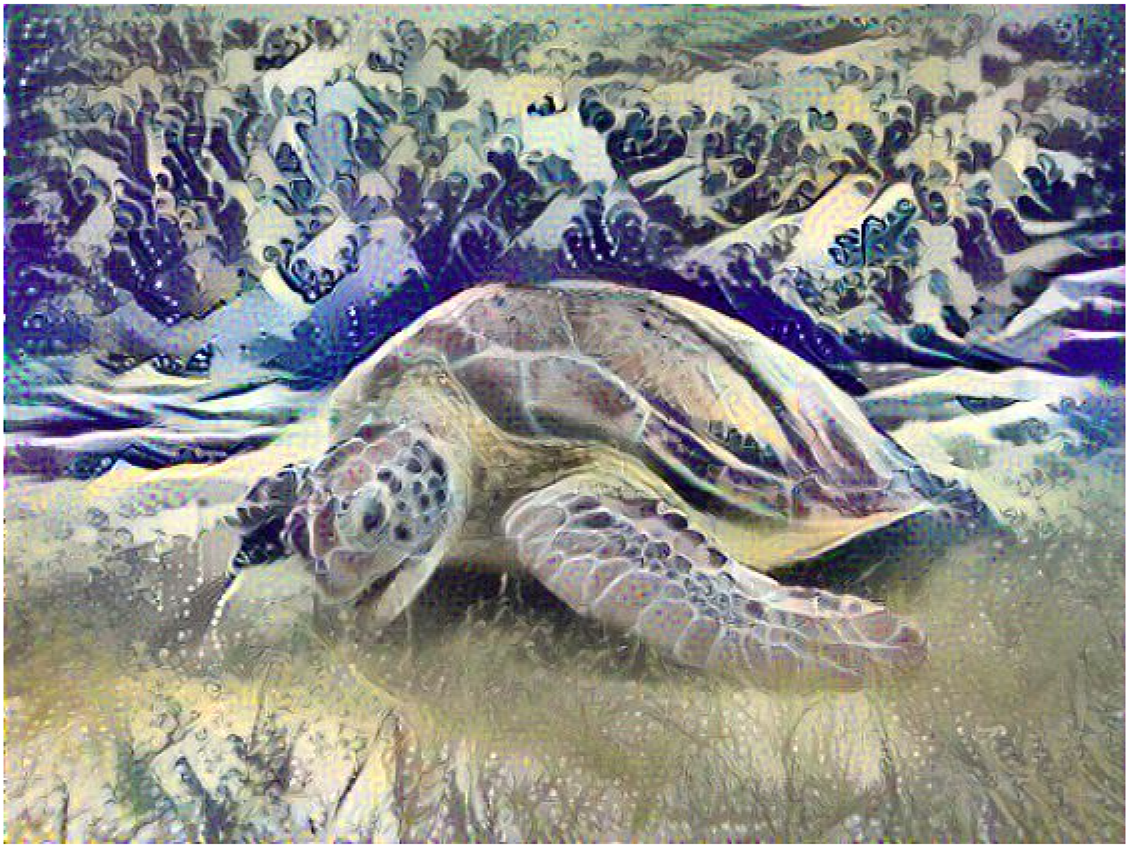
VGG - 16

Loss Curves-



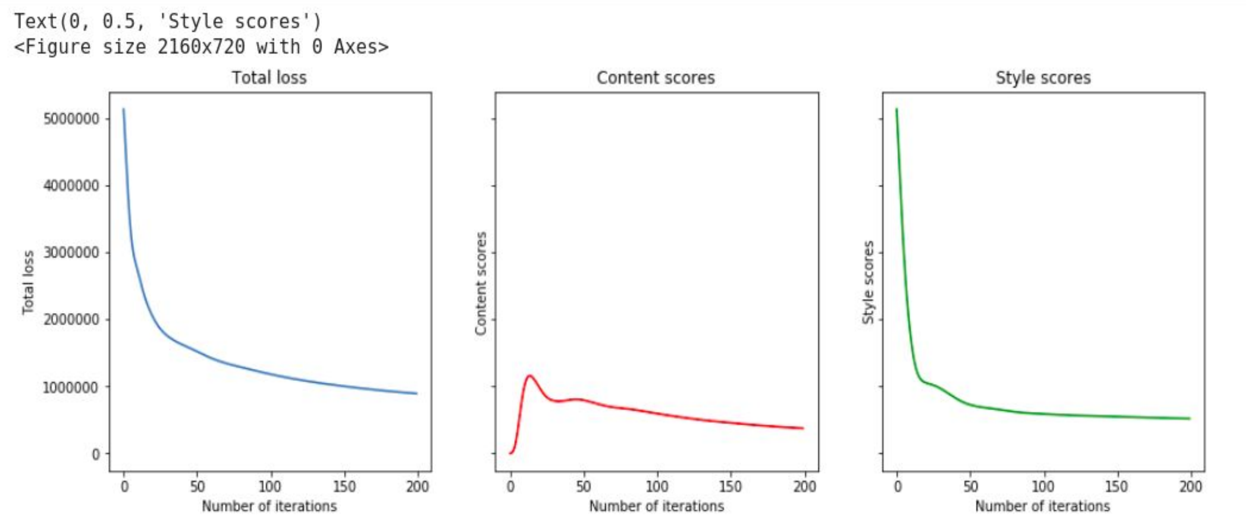
```
Text(0, 0.5, 'Style scores')
<Figure size 2160x720 with 0 Axes>
```

Image generated -



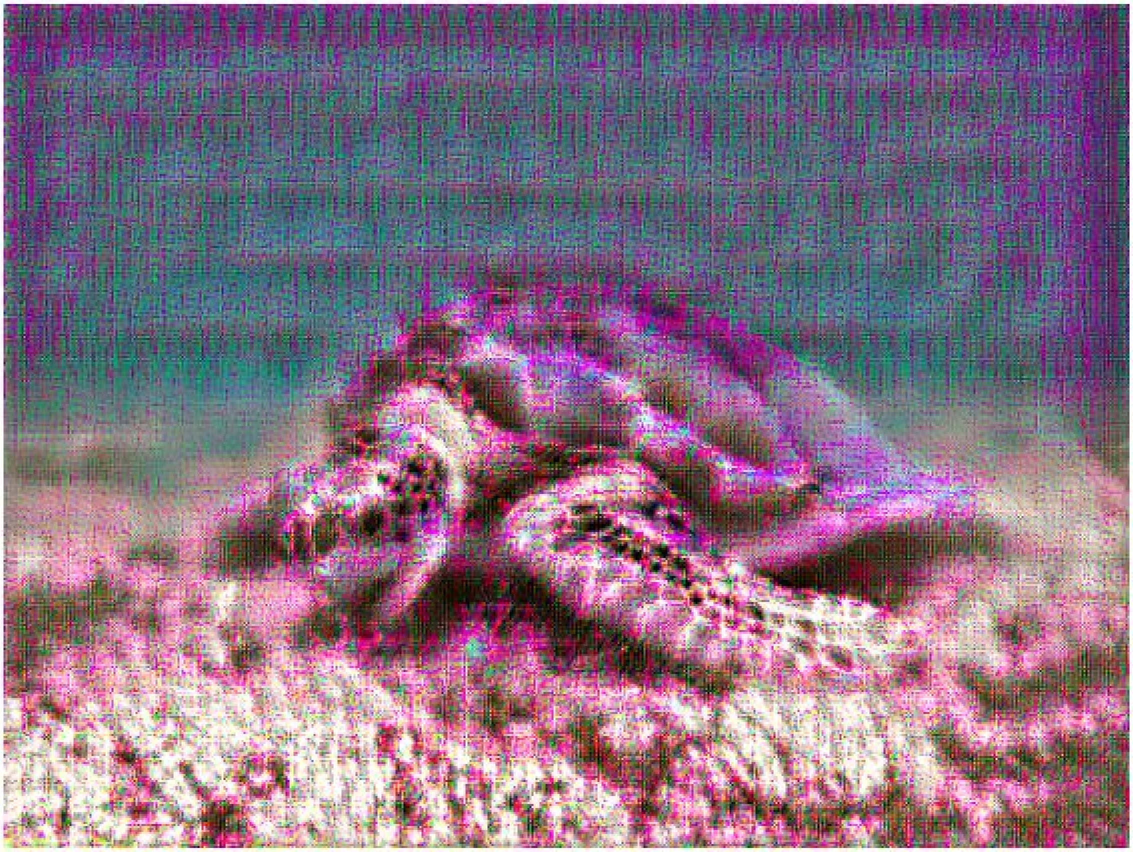
Resnets

Loss curves -



```
Text(0, 0.5, 'Style scores')
<Figure size 2160x720 with 0 Axes>
```

Image generated -



Visualizing outputs for different layers

In CNN architectures, different layers have different layer outputs. Therefore, the layer which we select will have a profound effect on the image generated. Below are the images generated for each layer and the learning curves for each of the following. The outputs are gathered form VGG 16 model.

Layer 1

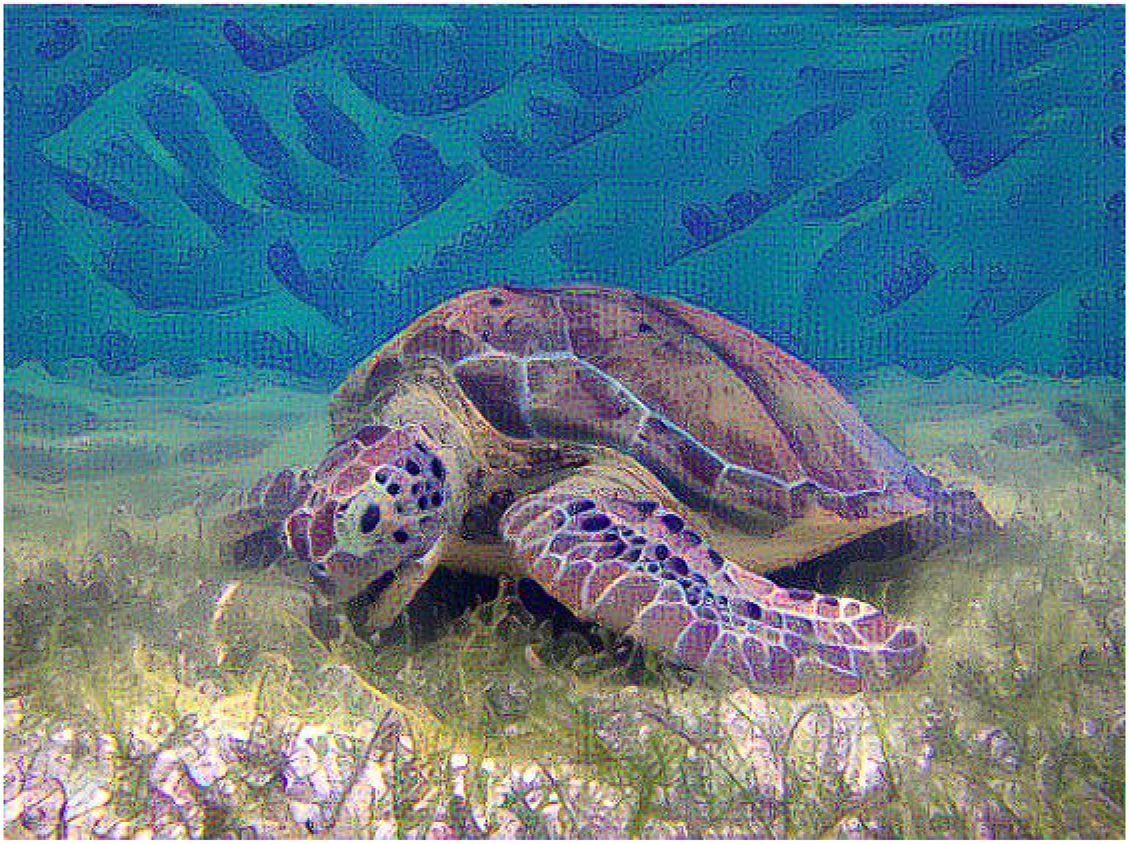
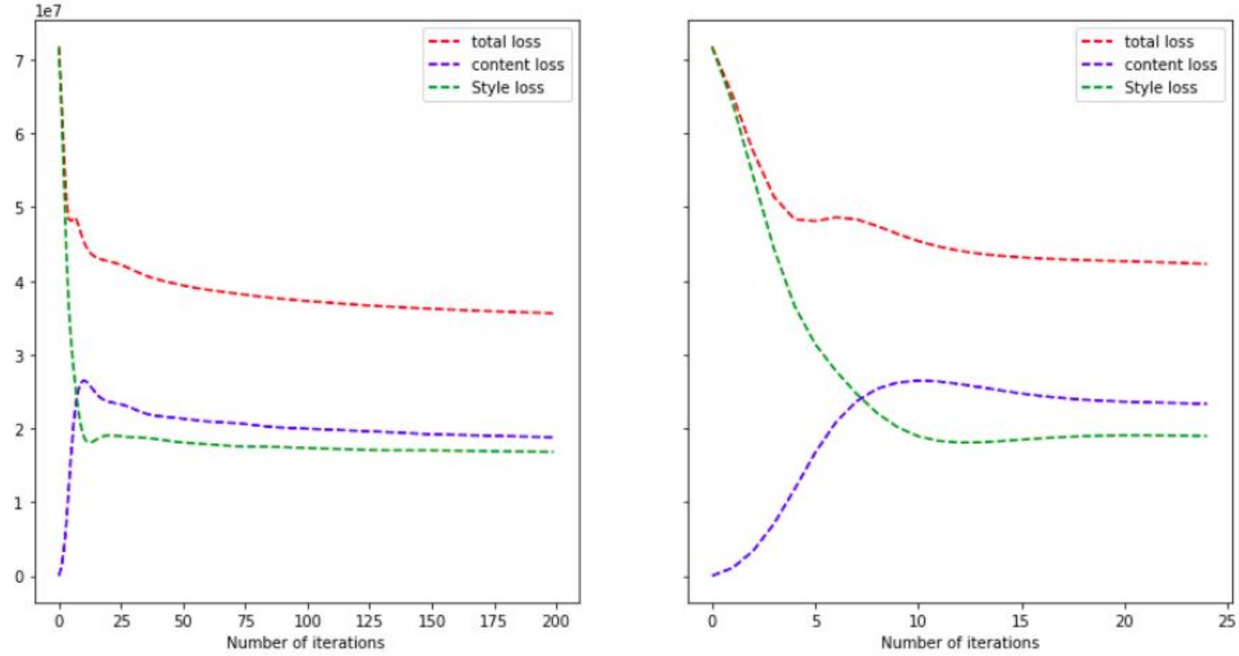


Image from layer 1 output. As we can see that the content is preserved more than the style because initial layer will output very sharp images, and there is no enough space to fill the gap for the style



Loss curves for the first 200 iterations and first 25 iterations respectively.

Layer 5

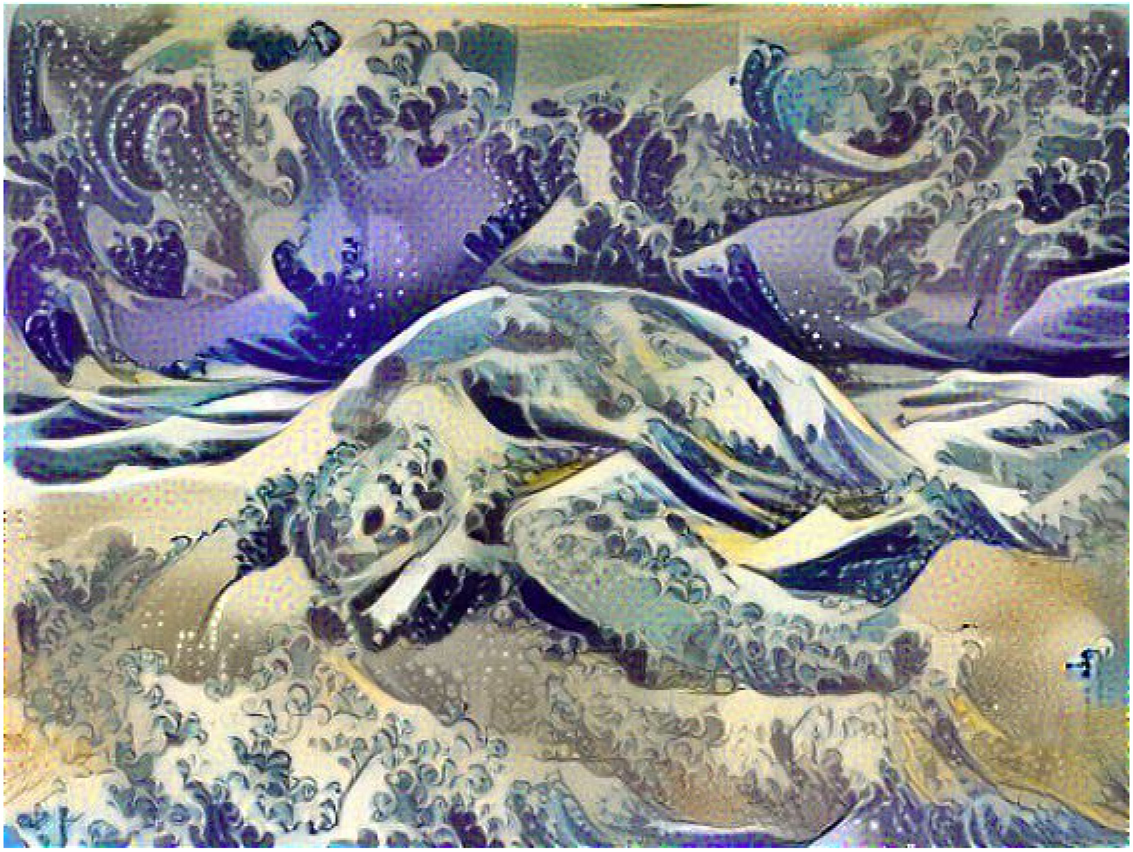
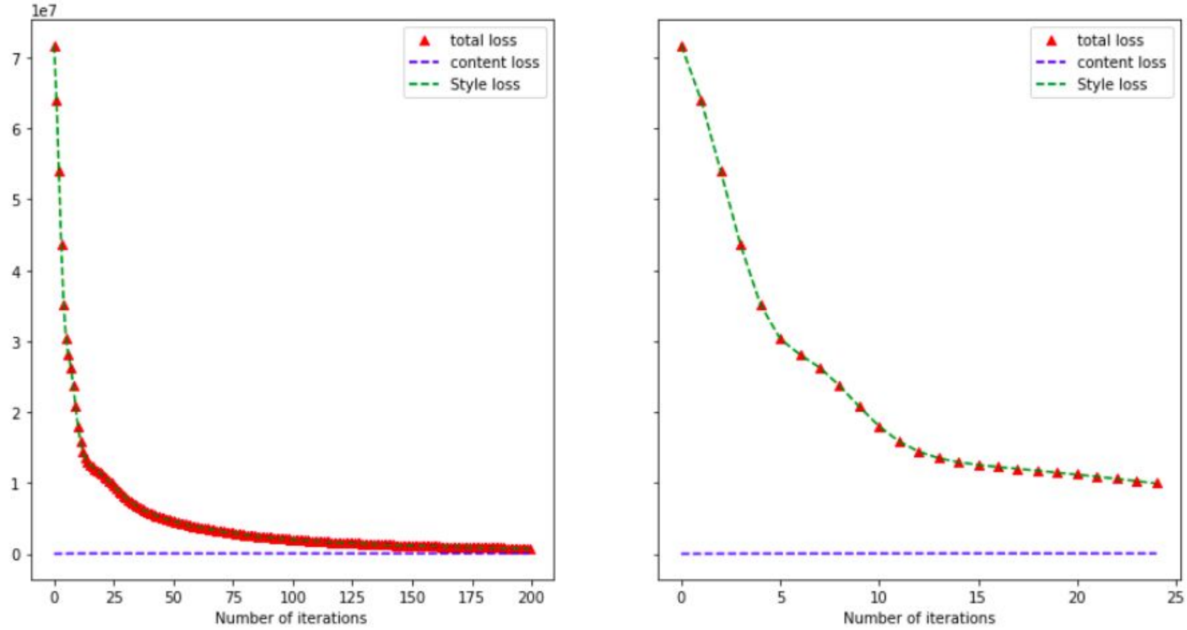


Image from layer 5. There are more elements of style than that of the content. This is because, the deeper layer produces more blurred output as compared to the initial layers, hence the leftover space is occupied by the style

```
<matplotlib.legend.Legend at 0x7f30359ede48>
```



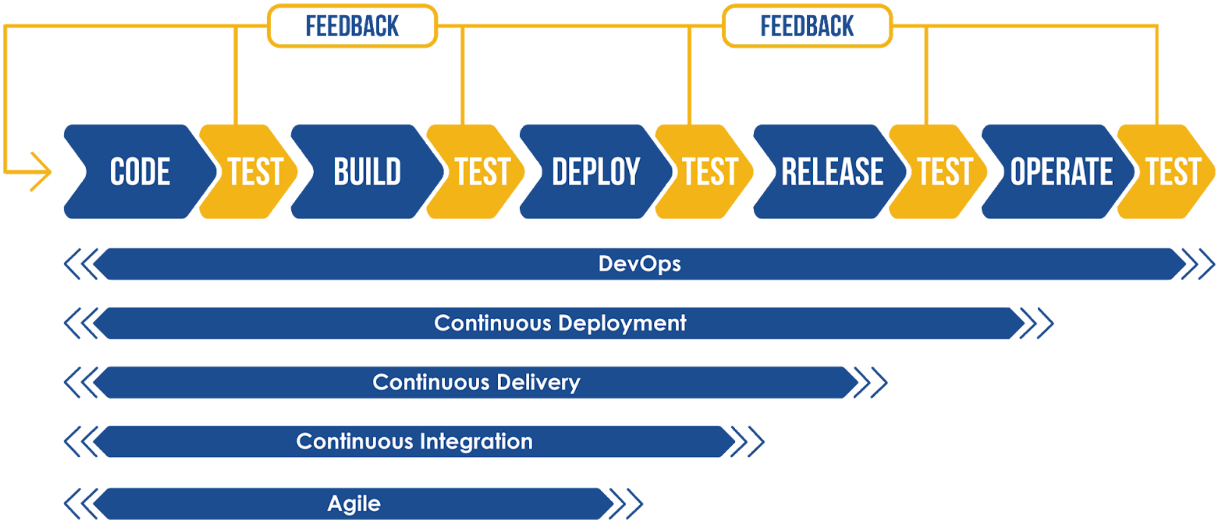
Loss curves for the first 200 iterations and first 25 iterations respectively.

Final Model

Based on the different outputs, the final model selected is VGG 16 based on the fact that it preserves the overall structure of the image. However, the amount of style required by the user is totally subjective. Therefore, we will provide that liberty to the user to select the layer output.

Software Deployment

The deployment strategy that we will be utilizing a CI/CD feedback loop.



Through continuous integration and continuous deployment utilizing GitHub, we will continuously get 'feedback' whether we are coding, building or deploying our NST model. With our product being completely subjective, the priority of the solution will be in its' deployment and how easily a user can utilize the NST solution.