# Arbitrary Style Transfer

...

Abinash Maharana
Aditya Gupta
Mohsin M. Hafiz
Kartik Agarwal

GitHub Link: https://github.com/Computer-Vision-IIITH-2021/project-team-vizzun

# Contents

# Overview

Style transfer is an optimization technique used to take two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but "painted" in the style of the style reference image.

# Problems Faced

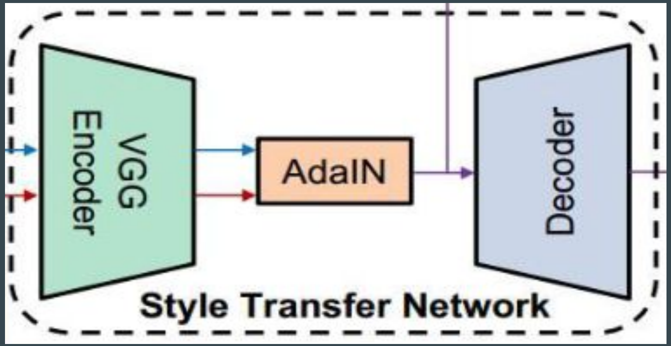
1. We were very new to deep learning and that’s where we faced the most difficulty. We had to go through several tutorials and learn,  and then start the implementation. We faced issues like the output images coming were not coming in proper colors,

2. The second  problem we faced was the lack of resources. We had limited resources such as google colab giving only limited time everyday and thus we had to switch several accounts. In the paper the model was trained for thousands of epochs while we could only manage around 50-100.

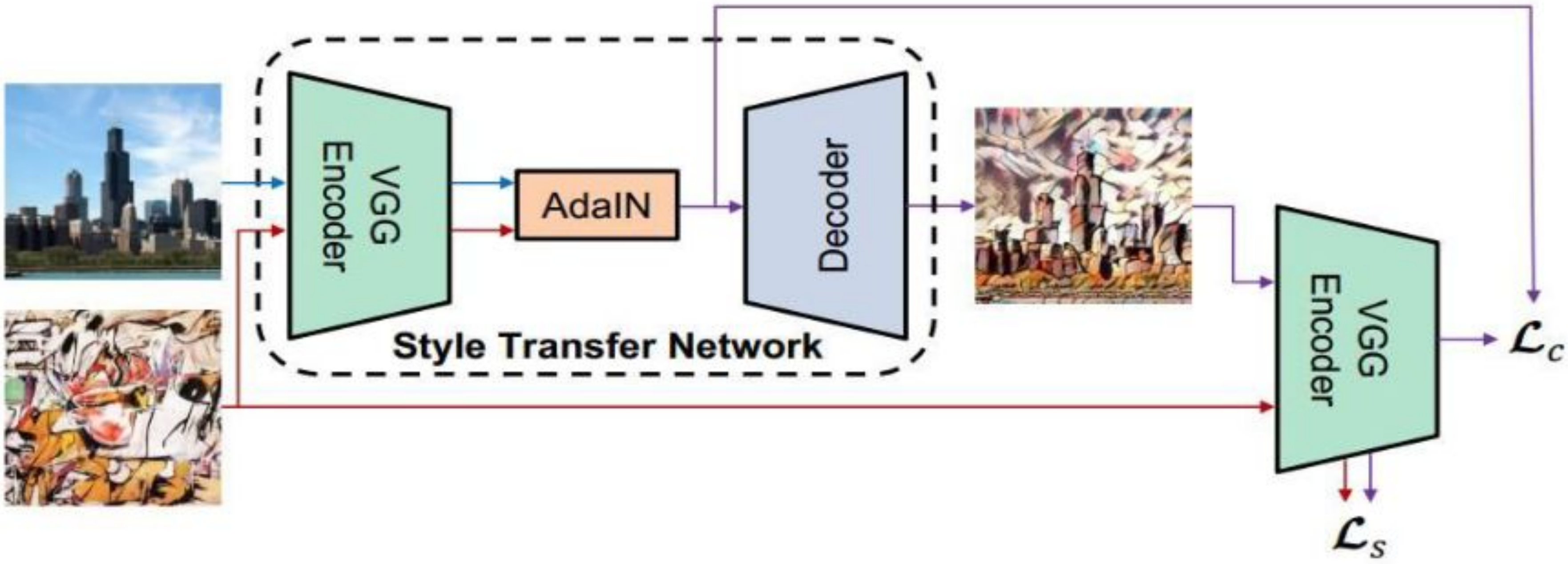
# Components

Main components of our architecture:

1. Encoder
2. Adaptive Instance Normalization train layer
3. Decoder and other utilities.



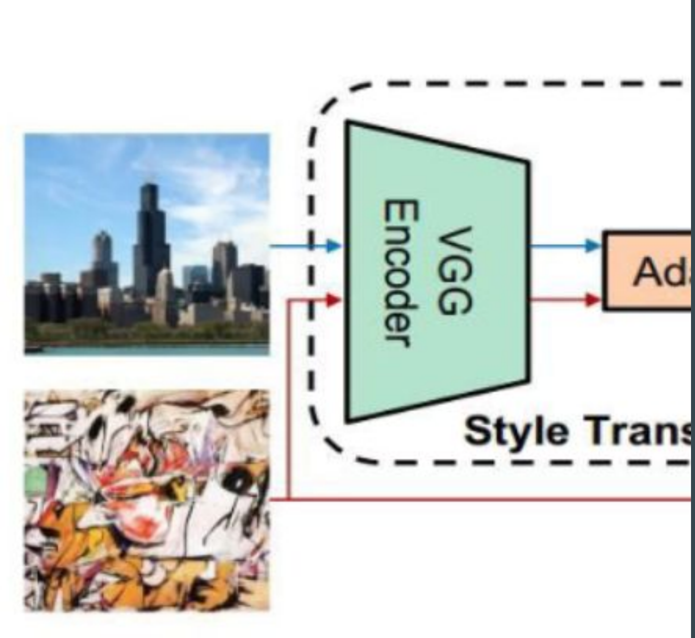

Our architecture is then trained on the MS-COCO dataset. After completion, our model would be run to get output to our content and style images.



VGG Encoder
AdaIN
Decoder
Style Transfer Network
VGG Encoder
$\mathcal{L}_c$
$\mathcal{L}_s$

# Encoder:

- Pre-trained VGG19 encoder.
- AdaIN layer works on output of encoder
- Training error calculated on encoded images



# AdaIN:

- AdaIN, or Adaptive Instance Normalization is a simple extension to Instance Normalization, which itself was an improvement over Batch Normalization.
- Previous methods learned affine parameters from data. These methods did that in different ways.
- AdaIN aligns the channel-wise mean and variance of the input image to match that of the style image.
- “Unlike BN, CN or IN, AdaIN has no learnable affine parameters.” - this helps it achieve real-time style transfer very fast.

$$\mathrm{BN}(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$

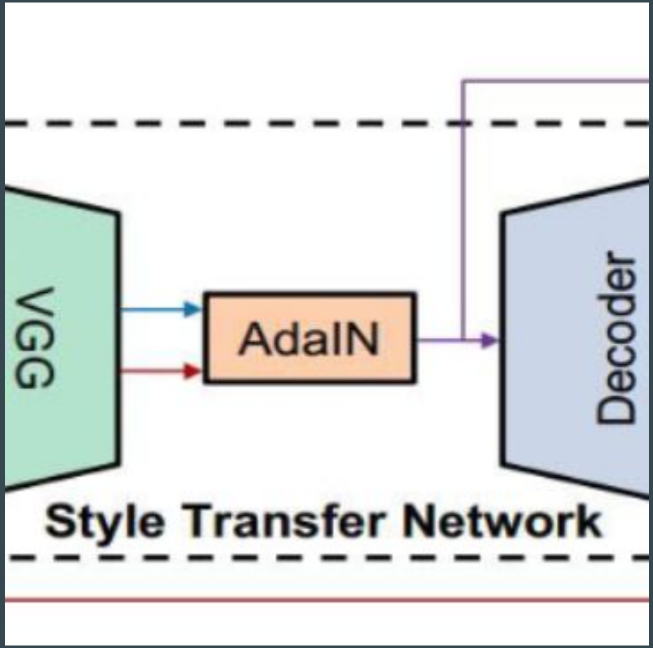
$$\mathrm{IN}(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$

$$\mathrm{CIN}(x; s) = \gamma^s\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta^s$$

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y)$$

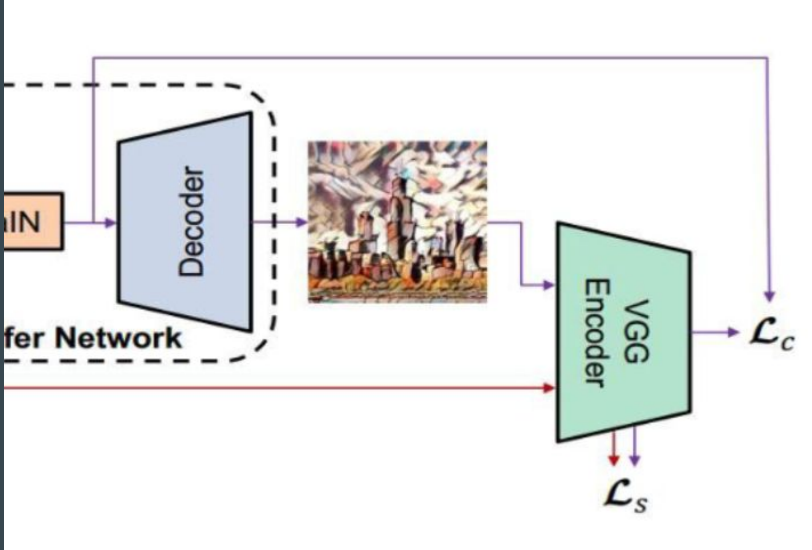
# AdaIN:

- Here we the change the feature statistics of the content image to that of the target image.
- We the multiply the variance of style image to our normalised content image and then add the mean of style image.
- This output is then send forward to decoder.



# Decoder:

- Architecture used was reverse of encoder by replacing the MaxPooling with upsampling.




- In training, output is send back to encoder for error calculation and training of the network.

# Results:

# References

- https://arxiv.org/pdf/1703.06868.pdf

Jo references paper mai diye hai voh daal sakte....

# Thank you.