U-Net based Image Style Transfer Network

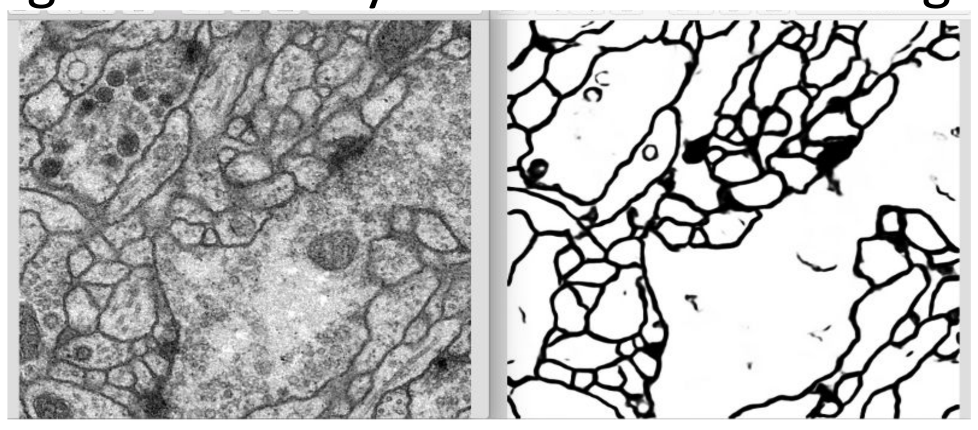
Kier Lindsay , Rafid Ashab Pranta, Vinaik Chhetri

INTRODUCTION

Image style transfer is a technique used to modify an input image based on a given style image. Gatys et al. introduced the model of image style transfer as an optimization problem. Furthermore, Johnson et al. trained a neural network to solve this optimization problem for a given style image. We tried to experiment on Johnson's model by using a different neural network, U-Net.

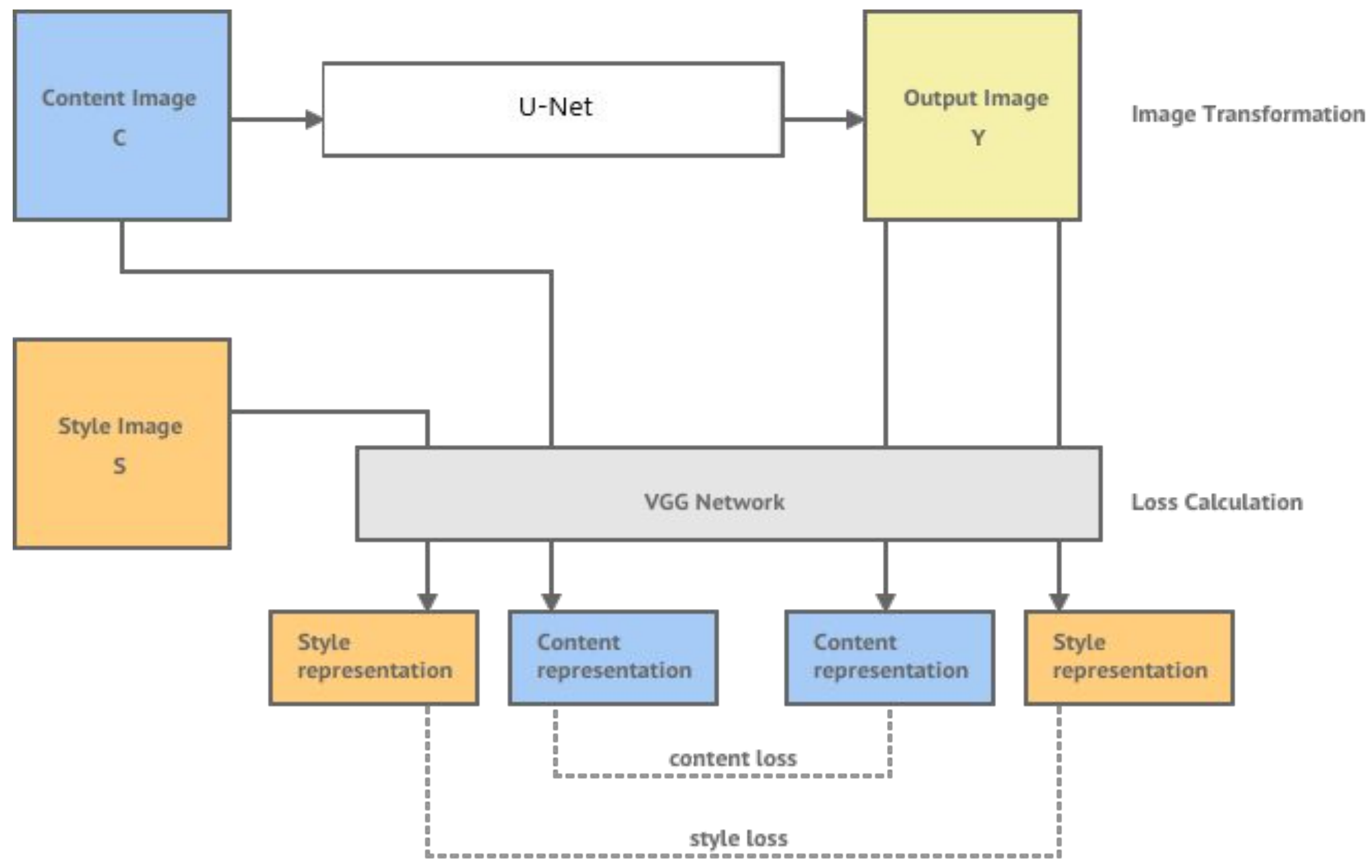
MOTIVATION

- Success of Vu-net using U-Net for its pose transformation.
- Precise segmentation of images provided by U-Net. Focusing on the defining content of the image and removing unnecessary details from the image.



- U-Net is fast and might improve performance.

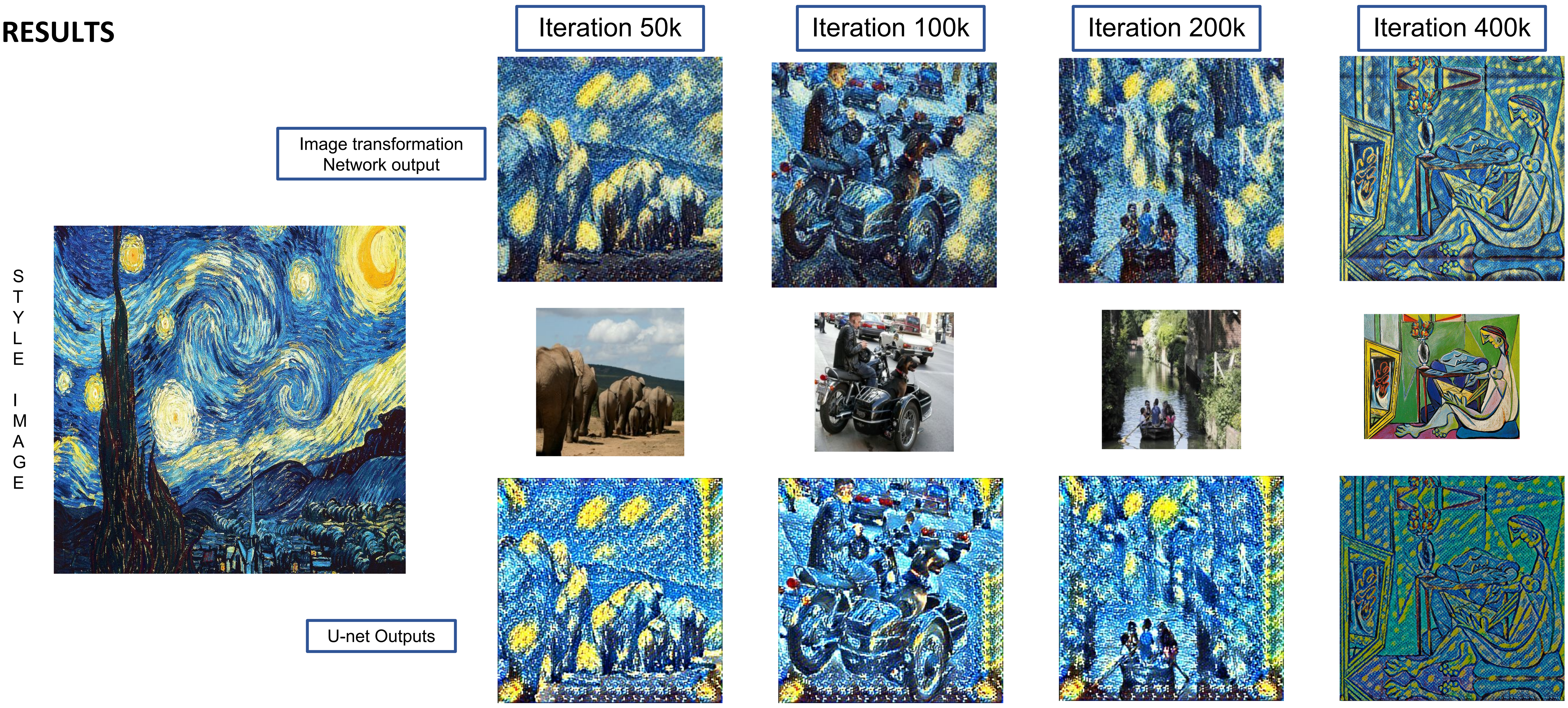
MODEL FORMULATION



TRAINING U-Net

We used the same parameters as Johnson et al. and only changed the Transform network in order to see the effects of a U-Net on Style transfer. We trained each network 5 epochs with over 80000 images form the coco dataset.

RESULTS



LOSS FUNCTION

$$\mathcal{L}_{content} = \frac{1}{2}\sum_{i,j}(F_{ij}^{l} - P_{ij}^{l})^2$$

Content loss between given content image and output image at layer L.

$$\mathcal{L}_{style} = \frac{1}{2}\sum_{l=0}^{L}(G_{ij}^{l} - A_{ij}^{l})^2$$

Style loss between styled image and output image at layer L.

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$$

Total loss is the linear combination of the content loss and style loss. Weights α and β can be modified to penalize content or style loss.

The output styled image is determined by obtaining a styled image that minimizes the total loss given the parameters content image, style image and the output image itself.

$$\hat{y} = \underset{y}{arg\,min}\, Total\ loss(y_c, y_s, y)$$

The model's loss functions are based on the model provided by Gatys et al. , which calculates the loss of the higher level features extracted from intermediate layers of a pretrained network instead of per-pixel loss between the output styled image and the input images.

Conclusions

- Images from both models are producing similar results.
- The bottlenecking effect of U-Net makes it good for learning larger features but less sensitive to the finer details. We had hoped that having each convolutions concatenated onto the upward pass would mitigate loss of finer details but that is not the case.
- There no significant speed improvements for using U-net.
- Using U-Net has caused a insignificant change thus leading us to the conclusion that, loss function in VGG-16 has a significant impact on the content and style of the output image. Therefore, in future we could use a different image classifier network to extract content and style layers from an input image other than VGG-16 such as ResNeT or Inception. This can help us determine effectiveness of VGG-16 to train the image transfer network.
- We need a better metric to compare our output images to Johnson's output Images