# 690IV: Project Report
# Fast Neural Style Transfer and Artist Identification

Shruti Gullapuram Saranya Krishnakumar

## Abstract

*This project explores methods for artistic style transfer based on convolutional neural networks. The core idea proposed by Gatys et. al became very popular and with further research Johnson et. al overcame a major limitation to achieve style transfer in real-time. We implement two approaches which are capable of achieving multiple and mixed style transfer, building on top of Johnson's fast style transfer algorithm. Another problem which we tackle is artist identification of fine art paintings. It is a challenging problem primarily handled by art historians with extensive training and expertise, and useful for digitizing a vast variety of artworks. We train and compare a custom baseline CNN model and a ResNet-18 network fine-tuned with transfer learning for this task. Additionally, we also try to perform an experiment by using our artist identification models on stylized images to understand what representation the artist model has captured.*

## 1. Introduction

As a technique that can combine both artistic aspects and recognition (content) aspects of images, style transfer has always become an interesting topic for researchers in the field of computer vision. Style transfer is essentially combining the style of one image into the content of another. The problem used to be difficult because it was hard to extract texture information using hand-crafted features, but with the advent of CNNs the problem could be tackled. Gatys [5] first posed this as an optimization problem, where the neural network was not actually required to *do* something but rather the backpropagation and optimization was used to slowly alter the image to incorporate style characteristics. This process was slow to stylize new images. Thus, Johnson [9] came up with the idea of training an *image transformation network* which was capable of real-time stylization during inference. However, both these methods are capable of performing only single style transfer i.e. one network is trained per style. Building on top of Johnson's work [9], we focus on implementing the ideas of Dumoulin et. al [3] and Yanai et al. [15], which are capable of performing multiple and mixed style transfer, while also exploring other effects such as spatial transfer and color preservation, and the use of dilated CNNs.

Another part of our project is artist identification of paintings given no extra information. Such a task can be useful for labelling vast number of paintings to be digitized. It is also challenging because artists can paint a variety of content with multiple styles (which can change over time). We were inspired to do this along with style transfer to see, in one way, how powerful style transfer can be in capturing artistic style and how well the same artistic style can be recognized.

## 2. Related Work

### 2.1. Style Transfer

Many older approaches such as [14] at- tempted to produce textures by simply scanning the sam- ple across and down to produce a larger image which still looks natural. This method works very well, but gener- ally only for homogeneous textures. Gatys [5] was the first to introduce a deep neural net- work approach that extracts representations to sep- arate and recombine the content and style of arbitrary im- ages. Although [9] showed that the style and content of an image can be disentangled and applied independently, the method is computationally expensive. Johnson's work [9] was able to speed up style transfer by training a feed-forward network to replace the optimization-based method of Gatys, which also allowed the transformation of video input in real-time.

Yanai et al. [15] propose that the network can take an additional conditional vector input indicating the style and the styles can be mixed at test time. Dumoulin [3] proposes that Johnson's [9] network can be modified to add *conditional instance normalization* layers which are capable of learning a set of parameters unique to each style. Huang [7] proposes a slightly different network which can instead learn a set of *adaptive instance normalization* parameters for each style, also permitting arbitrary style transfer. Ghiasi et al. [6] expand on this idea and use Inception V3 architecture in their Style Prediction Network to *predict* the normalization parameters instead of learning them, thus also achieving ar-

bitrary style transfer.

### 2.2. Artist Identification

Traditionally, image features like scale-invariant feature transforms (SIFT), histogram of oriented graidents (HOG), and others were used along with classifiers like SVMs, and k-nearest neighbors to identify artists and style [10] [12]. CNNs were used to extract features and with these extracted features SVMs were used for classification purpose [2].

## 3. Methods

### 3.1. Style Transfer

To train our style networks with different contents we use 40,000 images taken from the MS COCO 2014 [11] dataset. We chose a set of 12 different styles, some taken from the WikiArt paintings dataset [1] and some arbitrary styles.

Gatys et. al [5] propose to use the response layers of a pretrained VGG-16 network. Their main idea is:

1. Images similar in Content: High-level features extracted by pre-trained network are close in Euclidean distance
2. Images similar in Style: Low-level features extracted by pre-trained network share spatial statistics, and are quantified by Frobenius norm of Gram Matrices

$$\mathcal{L}_c(p) = \sum_{j \in \mathcal{C}} \frac{1}{U_j} \parallel \phi_j(p) - \phi_j(c) \parallel_2^2$$

Content Loss

$$\mathcal{L}_s(p) = \sum_{i \in \mathcal{S}} \frac{1}{U_i} \parallel G(\phi_i(p)) - G(\phi_i(s)) \parallel_F^2$$

Style Loss

$$\mathcal{L}(s, c, p) = \lambda_s \mathcal{L}_s(p) + \lambda_c \mathcal{L}_c(p)$$

Total Loss

Figure 1. Equations from [5]

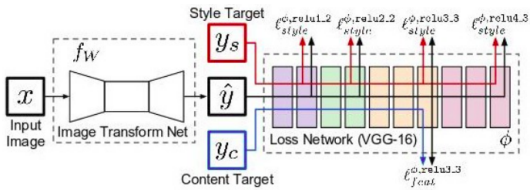



Figure 2. Network architecture from [9]

#### 3.1.1 Conditional Instance Normalization

Dumoulin et. al [3] introduced a new *conditional instance normalization* layer which is capable of learning a set of parameters unique to each style. The core idea was that performing an affine transformation on image features, conditioned by style features, is sufficient as a generic representation to achieve style transfer. This conditioning is performed after every *instance normalization* layer which was proved to be a better alternative to batch normalization as stated in [13]. We implemented mixed style transfer by giving the gamma and beta parameters weights at inference time as an input, and similarly these parameters can be applied to certain regions of the image to achieve spatial transfer.

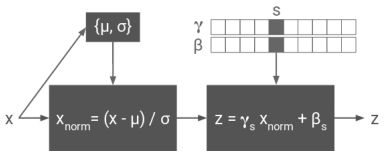



Figure 3. Style normalization parameters [3]

#### 3.1.2 Conditional Input Vector

Yanai et. al [15] proposed that multiple and mixed style transfer with a single feed forward network is possible without introducing a CIN layer. The idea, inspired from [8], is to provide a one-hot vector indicating which style image is shown during training, along with the style image, called the conditional input vector. The input vector is then duplicated and concatenated with the output of the third conv layer, before passing through a newly introduced 1x1 conv layer to combine them. During test time, the input vector can be given with weights for each style to achive mix style transfer.

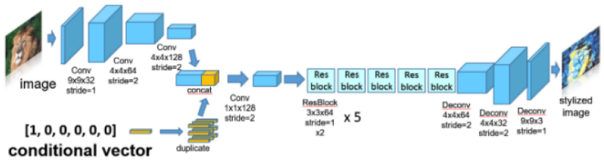


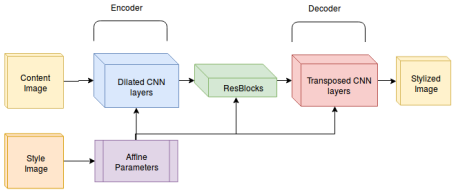

Figure 4. Network architecture from [15]

#### 3.1.3 Color Preservation

Gatys [4] also expanded his original algorithm to transfer style of one image while preserving the color of the original content image. The idea is take the stylized image only in the luminance channel (Y), while keeping the Cr and Cb channels the same as the content image. The motivation is that visual perception is far more sensitive to changes in luminance than in color.

#### 3.1.4 Using Dilated CNNs

The core network, *Image Transformation Net*, is essentially an encoder-decoder architecture. Hence, a relevant experiment we tried was to swap the conv layers with dilated convolutions and the upsampling layers with transposed convolutions. Dilated convolutions can increase receptive field size using fewer parameters, and with stride they can perform downsampling. We found that the model performed about 1.5 times faster than the CIN model, at a slight loss in output quality.




### 3.2. Artist Identification

For identifying artist from given painting, we train three neural networks on WikiArt dataset [1]. This dataset consists of around 100,000 paintings. To have a balanced dataset for training, we select 300 paintings each for 57 artists. For train, validation and test sets, we split the dataset in the ratio of 80:10:10. The training set consists of 240 paintings per artist and validation and test sets each consists of 30 paintings per artist.

The paintings in WikiArt dataset are of different shapes and sizes. To have uniform input image size, we take 224x224 crop of each painting. We also do random horizontal flip and normalize input images. This randomness add variety to train set and avoids over-fitting. For the validation and test sets, we do centre crop and normalize the images.

#### 3.2.1 Baseline CNN

We built two CNN architectures from scratch and trained it on WikiArt dataset. For both the architectures we used softmax classifier with cross entropy loss:

$$L_i = -\log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}})$$

Figure 5. Cross entropy loss

The low level image features are not sufficiently explored, they are quickly aggregated with shallow network architectures.

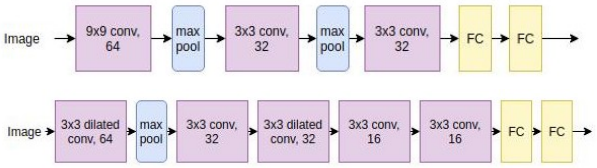



Figure 6. Baseline CNN architectures

#### 3.2.2 ResNet-18 Transfer Learning

The ResNet-18 architecture is known to work well for image recognition tasks. Hence we started with ResNet-18 network initialized with pretrained network weights, trained on the ImageNet dataset, trained for at least 20 epochs with Adam Optimizer. With increase in network depth, the gradients calculated in upper layers slowly degrade before reaching lower layers, hence the accuracy gets saturated. Residual blocks in ResNets make sure that upstream gradients are propagated to lower layers as well. Softmax classifier with cross entropy loss was used for ResNet-18 architecture as well. We tried two ways of transfer learning,

1. Retraining the entire network
2. Replacing last layer and training weights for only last layer

## 4. Results

### 4.1. Style Transfer

We evaluate our implementations taking both speed and quality into account. However deciding whether or not a particular style transfer was done well is mostly a subjective process and as such we can only rely on human evaluation.

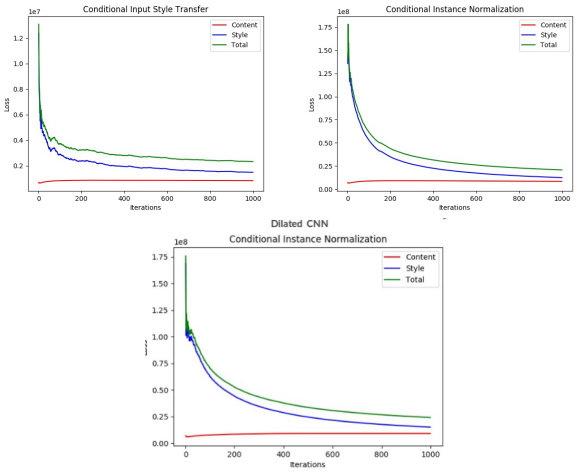



Figure 7. i. Conditional Input Vector (left), ii. Conditional Instance Normalization (middle), iii. Dilated CNNs (right)

#### 4.1.1 Single Style

Figure 8. Comparing models: i. Style (left), ii. Conditional Input Vector (middle), iii. Conditional Instance Normalization (right)

Figure 9. Comparing models: i. Style (left), ii. Conditional Input Vector (middle), iii. Conditional Instance Normalization (right)

#### 4.1.2 Mixed Style

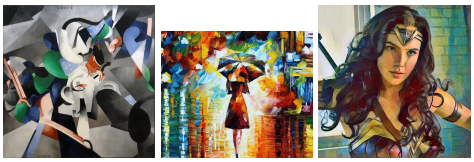


Figure 10. i. Style-1, ii. Style-2, iii. Mixed style result

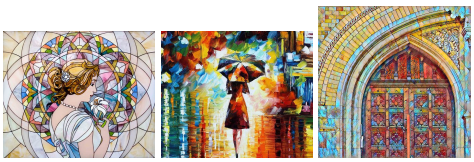


Figure 11. i. Style-1, ii. Style-2, iii. Mixed style result

Figure 12. i. Style-1, ii. Style-2, iii. Mixed style result

#### 4.1.3 Spatial Style Transfer and Color Preservation

Figure 13. i. Style (left), ii. Normal stylization (middle), iii. Color preservation (right)

Figure 14. i. Vertical transfer of 3 styles (left), ii. Horizontal transfer (right), with color preservation

#### 4.1.4 Dilated CNNs

Figure 15. i. Style, ii. Dilated CNN, iii. Normal stylization

| Training time (s) for 1000 iterations | | |
|---|---|---|
| Condition Input Vector | Conditional IN | Dilated CNNs |
| 7027.73 | 10149.36 | 7294.6 |

### 4.2. Artist Identification

| Baseline CNN | |
|---|---|
| Hyperparameters | Val Accuracy |
| lr = 0.1 , $\lambda = 0.01$ | 12 |
| lr = $e^{-2}$, $\lambda = 0.01$ | 27 |
| lr = $e^{-3}$, $\lambda = 0.01$ | 35 |
| lr = $e^{-4}$, $\lambda = 0.001$ | **47** |
| lr = $e^{-5}$, $\lambda = 0.0001$ | 27 |

| Finetuned ResNet | |
|---|---|
| lr = $e^{-3}$, $\lambda = e^{-4}$ | Train Accuracy = **91** |
| lr = $e^{-4}$, $\lambda = e^{-4}$ | Val Accuracy = **77** |

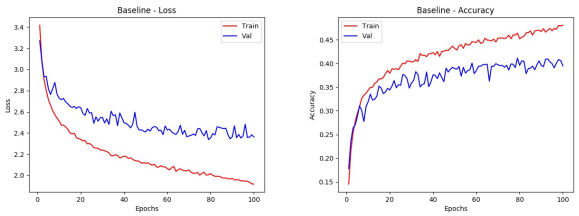



Figure 16. i. Baseline Loss (left), ii. Baseline Accuracy (right)

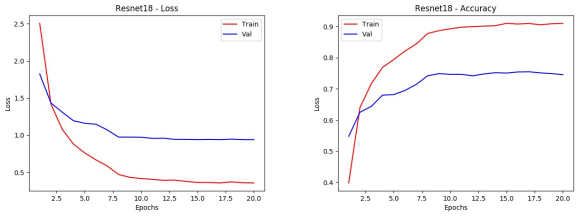



Figure 17. i. Resnet18 Finetuned entire network - Loss (left), ii. Accuracy (right)

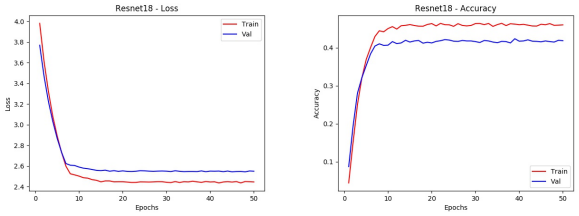



Figure 18. i. Resnet18 Finetuned last layer - Loss (left), ii. Accuracy (right)

#### 4.2.1 Recognizing Artists of Stylized Images

We found that the most frequently predicted artists were Erte, Maurice, and M.C. Escher. We trained a painting style by artist Boris, who painted many different styles and contents, and found that the artist predicted was M.C. Escher was in fact very similar in style to majority of Escher's works. Due to the fact that we can only train our style models with limited styles, it is difficult to do a proper analysis of the effect of stylized images. Thus this is one area we would really like to explore by also expanding our models to arbitrary style transfer.

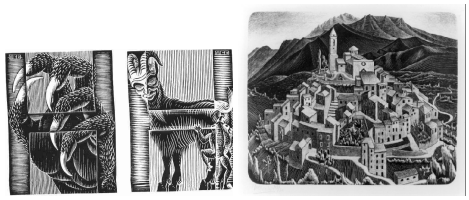


Figure 19. i. Boris painting, ii. Escher painting

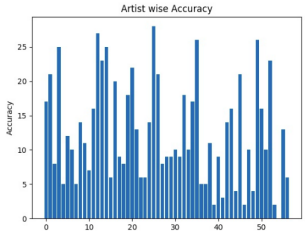



Figure 20. i. Accuracy per artist

## 5. Conclusion and Future Work

In this project we implemented multiple and mixed style transfer as in [3] and [15]. We also experimented with dilated CNNs to observe their effect. Further, we were also able to achieve spatial transfer and preserve color of content. In addition to style transfer, we also built classification models to predict artists from a given artwork.

For future work, we would like to achieve arbitrary style transfer based on [6]. Arbitrary style transfer would also allow us to do a thorough analysis of how artist identification models are affected by stylized images. Another improvement would be to modify [15] to produce the conditional input vector on-the-fly by passing the style image through a small CNN.

## References

[1] Kaggle wikiart dataset. https://www.kaggle.com/c/painter-by-numbers.

[2] Y. Bar, N. Levy, and L. Wolf. Classification of artistic styles using binarized features derived from a deep neural network. In *ECCV Workshops*, 2014.

[3] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *CoRR, abs/1610.07629*, 2(4):5, 2016.

[4] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*, 2016.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016.

[6] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017.

[7] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR, abs/1703.06868*, 2017.

[8] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016.

[9] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[10] J. D. Jou and S. Agrawal. Artist identification for renaissance paintings. 2011.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[12] T. E. Lombardi, M. J. Lang, and A. E. Campbell. The classification of style in fine-art painting. 2005.

[13] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

[14] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[15] K. Yanai and R. Tanno. Conditional fast style transfer network. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 434–437. ACM, 2017.