DLLAB WEEK 9 Pre-Report : Network become Artist

Hyunsoo Cha[1], Kwanghoon Sohn[2]
[1]Yonsei University. [2]Yonsei University.

1 Image Style Transfer Using Convolutional Neural Networks

1.1 Introduction

To sum up this paper, researchers are introducing the idea of extracting style and content from the original image into feature vector and creating new images through neural networks. It is the process of extracting the style of the content image and changing the style of the painting. That is, the skeleton of the image follows the content, but the style is changed through style image.

First, I explain the content presentation. Throughout CNN, content representation is reduced in size by downsample layers, such as max-pooling layers, which can be visualized at each stage. The style transfer used a VGG network, with convolution layers ranging from 1 to 5. Most of the lowest layers look perfect, but the higher the layer, the more detailed pixel information is crushed. Next, they describe style reconstruction. At the top of the CNN, feature space is extracted from the input image only for texture information. They express it as extracting correlations between different features of different layers on CNN.

In this paper, they show a high-performance feature presentation in which CNN independently modifies and processes natural images of content and style. These algorithms are called the A Neural Algorithm of Artistic Style. It applies state-of-the-art CNN technology to constrain texture synthesis through feature representation.

1.2 Deep Image Representations

First, the researchers implemented it over the VGG network[6]. It created a feature space with 16 convolutional layers and 5 pooling layers, and normalized the network so that the mean activation of the convolutional filter was 1. The fully connected layer was not used by the researchers.

Content Representation The terms are as follows: A layer of N_l distinct filters is a feature map with each size of M_l. The height of the feature map multiplied by the width is M_l. The response of the layer can be represented by a $F^l \in^{N_l \times M_l}$ matrix. F^l_{ij} stands for activation of the ith filter of the position j of layer l. To visualize this image information, they use gradient descent[4], which combines white noise images. Let $\vec{p}$ and $\vec{x}$ be the original image and the generated image. And let P^l, F^l be the feature representation of layer l. The loss of content is available next.

$$L_{\textbf{content}}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2 \tag{1}$$

Partial differentiation of the above expression for activation yields the following results:

$$\frac{\partial L_{\textbf{content}}}{\partial F^l_{ij}} = \begin{cases} (F^l - P^l)_{ij}, & \text{if } F^l_{ij}0 \\ 0, & \text{if } F^l_{ij}0 \end{cases} \tag{2}$$

The expression used standard error back-propagation. As it passes through the layer, it becomes distant from the original and the content of the image appears. After reconstruction, the original pixel value is not available.

Style Representation Extracting text information and making it a feature space is the core of style extraction in input image. There is a correlation between the responses of different filters. It consists of Gram matrix[2], written in the formula as follows. $G^l \in^{N_l \times N_l}$, G^l_{ij} is internal to the feature map of the layer. $G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$. Researchers were able to obtain a stationary with feature correction of multiple layers. But this is not a global arrangement. Total loss is obtained by the following formula.

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G^l_{ij} - A^l_{ij})^2 \tag{3}$$

Total style loss is as follows. Also, the style loss function is as follows: $L_{\textbf{style}}(\vec{a},\vec{x}) = \sum_{l=0}^{L} \omega_l E_l$. ω_l is the weight factor. Partial differentiation is as follows.

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2}((F^l)^T(G^l - A^l))_{ji}, & \text{if } F^l_{ij} > 0 \\ 0, & \text{if } F^l_{ij} < 0 \end{cases} \tag{4}$$

It is a loss function and backpropagation that is very similar to content.

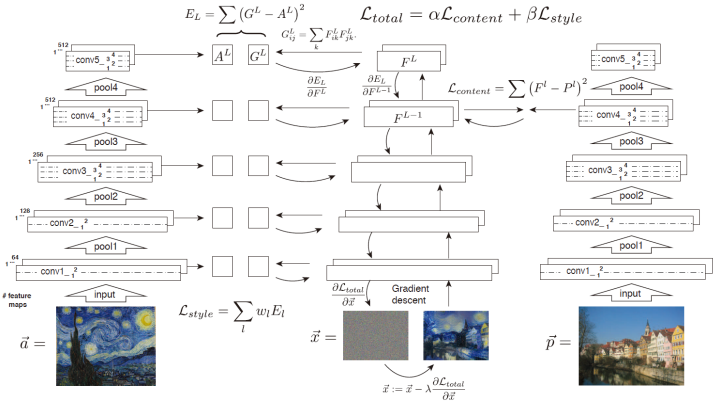

Figure 1: The above picture depicts a style transfer algorithm. The order is as follows: First, $\vec{a}$ passes through the CNN, resulting in style representation A^L. Similarly, $\vec{p}$ passes through CNN, resulting in content representation P^L.

A random white noise $\vec{x}$ enters the CNN, resulting in style features, content features $G^l, and F^l$. This allows them to compute $L_{\textbf{style}}(left)$, and $L_{\textbf{content}}$. Total Loss is a linear combination of the two losses. They differentiate this and proceed with optimization in a way that minimizes loss. This is then updated by $\vec{x}$ to obtain the final style transfer value.

Style Transfer Total Loss is as follows.

$$L_{\textbf{total}}(\vec{p},\vec{a},\vec{x}) = \alpha L_{\textbf{content}}(\vec{p},\vec{x}) + \beta L_{\textbf{style}}(\vec{a},\vec{x}) \tag{5}$$

α, β, means weighting factor for content and style reconstruction. Gradient values for partial differentiation of this to $\vec{x}$ are a process for optimization.

1.3 Results

The value of this paper is demonstrated by passing CNN differently for style and content to obtain feature space. A good adjustment of the weight value α, β allows them to regulate the styling of images.

As they can see, the larger the value of the ratio, the closer the content, and the smaller the value of the ratio, the closer the style. The style transfer of the photo style is somewhat obvious.

1.4 Discussion

The researchers suggested how to use feature representation. As a limiting factor for this network, it can be the low resolution of the synthesized image. As much as the dimension of the optimization problem, the increase in the number of units in CNN linearly increases the number of pixels. Thus, the speed at which it takes to synthesize images is strictly determined by their resolution. Another problem is that the composite image becomes low-level noise. Noise is characterized and appears as a resemble in the filter unit of the network.

2 Perceptual Losses for Real-Time Style Transfer and Super-Resolution

2.1 Introduction

A supervised method uses the difference between ground-truth images per pixel to compute the loss function. This method was used in segmentation papers[3] and also in super-resolution[1]. However, there is a downside to this approach: it is poor at capturing the perfect difference between output and ground-truth images.

In style transfer, the output must be observed to be a significant change in color and texture than the input image. For super-resolution, they implement 4, 8x super-resolution, replacing per-pixel-loss. Simultaneously, they solve the optimization problem for style transfer. The feed-forward problem has been solved by deep CNN with a per-pixel loss function.

2.2 Methodology

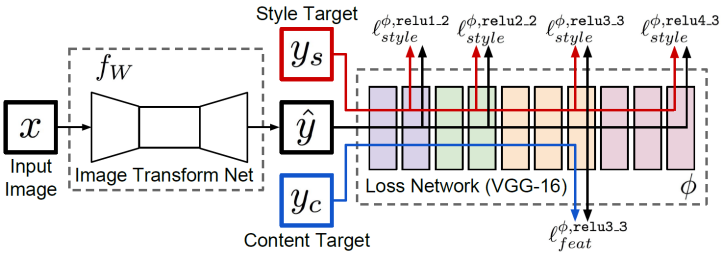

Figure 2: Researchers train an image transformation network to convert input images into output images.

It optimizes image classification by defining a perfect loss function. The above network is configured with image transformation network f_W, loss network ϕ with loss functions $l_1, \cdots, l_k$. The image conversion network consists of a deep-residual convolutional neural network with a weight of W. The method of mapping from input images to output images is $\hat{y} = f_W(x)$. The scale value, which measures the difference between target image y_i and output image $\hat{y}$, is $l_i(\hat{y}, y_i)$. Training proceeds with a stochastic gradient design and organizes it into the following loss function.

$$W^* = \text{argmin}_W E_{x,y_i}[\sum_{i=1} \lambda_i l_i(f_W(x), y_i)] \tag{6}$$

Pre-training semantic information and perceptual information by encoding and training CNN with image classification to measure the loss function implemented by the researchers is essential for better performance improvement of per-pixel loss[4]. Loss network is $l^{\phi}_{\textbf{feat}}$ which is defined feature reconstruction loss. Also, style reconstruction loss is $l^{\phi}_{\textbf{style}}$. For x input images, the content target is defined as y_c and the style target is called y_s.

Image Transformation Networks The image transformation network follows the structure of Ranford[5]. Instead, the pooling layer is not used; it uses the network's own downsampling and upsampling functions to partially create a convolution layer using strides. In addition, the network implements five separate blocks. For non-residual blocks, the output layer is derived via the special batch norm and ReLU. The input image for style transfer is $3 \times 256 \times 256$, and the input image for super-resolution is $3 \times 288 \times 288$.

The upsampling factor for super-resolution is set to f. The logical block sets the convolutional layer of $\log_2 f$ to strand 1/2. The upsampling function is set to a convolution that is partially strided rather than fixed.

In style transfer, two stride-2 convolution are used as downsamples. The input image is followed by a manual block, and the two convolutional layer is upsampled to stride 1/2. Even though the input and output images are the same size, the network benefits from downsample and upsample. The first advantage is that it is easy to calculate. If they make 3×3convolution C filter the size of the input image $C \times H \times W$, then $9HWC^2$. This is the same computation as $DC \times H/D \times W/D$ for the DC filter. After downsample, a larger network can be used for the same amount of operations. The second advantage is that it can effectively get the size of the receptive field. It is the

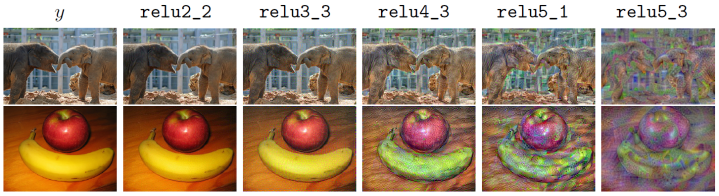

Figure 3: The figure above shows feature reconstruction loss $l^{\phi,j}_{\textbf{feat}}(\hat{y}, y)$ which is a plot of each layer to find the output image. The network used a pre-trained VGG-16. The more layers passed, the more the content and spatial information of the image seems to be preserved, but the color, texture, and accurate form are crushed.

high-quality style transfer that needs to change a large portion of the image in a consistent manner; if downsampling is done by D times, the receptive field will be $2D$ for each 3×3 convolution. In other words, it will have a larger receptive field.

Perceptual Loss Functions For loss network ϕ, feature reconstruction loss is defined as follows.

$$l^{\phi,j}_{\textbf{feat}}(\hat{y}, y) = \frac{1}{C_j H_j W_j} ||\phi_j(\hat{y} - \phi_j(y))||^2_2 \tag{7}$$

They compute over the loss network ϕ without matching $\hat{y} = f_W(x)$ for pixels of target image y. The style reconstruction loss is calculated as follows.

$$G^{\phi}_j(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \tag{8}$$

The above formula has the following. $\phi_j(x)$ is the activation of the jth layer of the network ϕ for the input image x. In addition, the feature map's form consists of $C_j \times H_j \times W_j$. $G^{\phi}_j(x)$ is Gram-matrix. For $\phi_j(x)$ given in C_j dimensions at each point in the $H_j \times W_j$ grid, Gram-matrix is proportional to covariance, not to the center of the feature in the C_j dimension. The grid location is assumed to be an independent sample. Style reconstruction loss is defined by the following formula.

$$l^{\phi,j}_{\textbf{style}}(\hat{y}, y) = ||G^{\phi}_j(\hat{y}) - G^{\phi}_j(y)||^2_F \tag{9}$$

2.3 Experiments

Style transfer consists of a combination of target content image y_c and target style image y_s. The output image is calculated by the following formula.

$$\hat{y} = \textbf{argmin}_y \lambda_c l^{\phi,j}_{\text{feat}}(y, y_c) + \lambda_s l^{\phi,J}_{\text{style}}(y, y_s) + \lambda_{TV} l_{\text{TV}}(y) \tag{10}$$

The value y means white noise. They proceed with optimization with L-BFGS. Dataset consists of COCO. Using Adam, the learning rate consists of 1×10^{-3}. For single-image super-resolution, they implemented feature reconstruction loss to relu2-2 on the VGG-16 network. The network used SRCNN, minimizing 33×33 patches to per-pixel loss. A total of 10^9 were trained over and over again and again.

2.4 Conclusion

In conclusion, the researchers combined the advantages of feed-forward image transformation tasks with an optimization approach for how to generate images due to training of feed-forward transformation networks in a perceptual loss function. They were able to directly numericalize and compare how much performance has improved over existing models of style transfer, and also have clear advantages in terms of computational speed. In addition, for single-image super-resolution, perceptual loss allows the boundary to produce a clear output of the model.

3 Reference

[1] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *arXiv preprint arXiv:1505.07376*, 2015.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[4] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[5] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.