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

June 1, 2019

This article is written as I read "Perceptual Losses for Real-Time Style Transfer and Super-Resolution" by Johnson et al. [3].

## 1 Introduction

- An approach to solve the image transformation problem is to train a network in a supervised manner, using a per-pixel loss. Per-pixel loss does not capture perceptual differences between the output and the ground-truth images, but the approach is fast.
- Perceptual loss functions can be implemented by comparing differences between high-level features extracted from pretrained CNNs. High quality images can be generated by minimizing this function. The style transfer paper by Gatys et al. is an example that does this [1]. However, this approach is slow.
- The present paper trains a feed-forward network using perceptual loss functions. The network can transform images in real-time and achieve good quality outputs. The authors apply this approach to two problems: style transfer and super-resolution.

## 2 Method

- The system consists of two components:
  - An image transformation network $f_W$.
    - Parametermized by wegiths $W$.
    - Transforms input image $x$ into output image $\hat{y}$.
  - A loss network $\phi$.
    - Used to define several loss functions $\ell_1$, $\ell_2$, …, $\ell_k$.
    - Each function $\ell_i$ computes a scalar value $\ell_i(\hat{y}, y_i)$, which measures the difference between the output image $\hat{y}$ and a target image $y_i$.
- The network is trained to find the optimal weight $W^*$ that minimizes a weighted sum of the loss funcions:

$$W^* = \arg\min_W E_{x,\{y_i\}}\left[\sum_{i=1}^{k} \lambda_i \ell_i(f_W(x), y_i)\right].$$

- The loss network $\phi$ is used to define a *feature reconstruction loss* $\ell^{\phi}_{feat}$ and a *style reconstruction loss* $\ell^{\phi}_{style}$ that measures differences in content and style between images, respectively.

- Each input image has an associated *content target* $y_c$ and *style target* $y_s$.
  - For style transfer $y_c$ is $x$ itself, and $y_s$ is the image having the style that we want to transfer $x$ to.
  - For super-resolution, $x$ is a low-res image. $y_c$ is the high-res image. The style reconstruction loss is not used.

## 2.1 Image Transformation Networks

- The architecture follows the guidelines by the DCGAN paper [4].
  - No pooling layers.
  - Downsampling and upsampling are implemented by strided and fractionally strided convolution layers.
- The paper designed two network: the style transfer network and the super-resolution network. The style transfer network receives an input image and tranfer it to a fixed style, determine at training time.
- The following are the common features between the two networks.
  - The networks use 5 residual blocks [2].
  - All non-residual convolutional layers are followed by batch normalization and ReLU non-linearity.
  - The output layer uses scaled tanh to make sure the pixels are in the range $[0, 255]$.
  - The first and the last convolutional layers use kernels of size $9 \times 9$. Other convolutional layers use $3 \times 3$ kernels.
- The networks take the following inputs and outputs:
  - The style transfer network inputs and outputs are images of size $3 \times 256 \times 256$.
  - The super-resolution network outputs an image of size $3 \times 288 \times 288$. If the upsampling factor is $f$, then the input of of size $3 \times (288/f) \times (288/f)$.
- In the super-resolution network with upsampling factory $f$, the residual blocks are followed by $\log_2 f$ convolutional layers with stride $1/2$.
- In the style transfer network, two stride-2 convolutional layers downsample the input. The result is then passed to the residual blocks and then two fractionally strided convolutional layers to upsample it to the original resolution.
- Downsampling and then upsampling in the style transfer network has the following benefits:
  - Downsampling significantly reduces the cost of evaluating the residual blocks.
  - Downsampling increases the effective receptive field sizes of each pixel in the input.
- The paper argues that shortcut connections in residual blocks are beneficial to image transformation because, in most cases, the output image should share structure with the input image.
- The detailed architecture can be found in the supplementary material of the paper.[1]

[1] `https://cs.stanford.edu/people/jcjohns/papers/eccv16/JohnsonECCV16.pdf`

### 2.2 Perceptual Loss Function

- The paper uses a *loss network* $\phi$ to define two perceptual loss functions.
- Here, $\phi$ is VGG-16 pretrained for image classification.
- Let $\phi_j(x)$ be the activation of the $j$th layer of the network. We restrict ourselves to convolutional layers. Let $C_j \times H_j \times W_j$ be the shape of $\phi_j(x)$.
- The first loss function is the **feature reconstruction loss**:

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

  The paper demonstrated that finding $\hat{y}$ that minimizes the feature loss function tend to preserve the spetial features while color, texture, and exact shape degrade as we use deeper layers.
- The second loss fucntion is the **style reconstruction loss**. This is defined with the *Gram matrix* $G_j^\phi(x)$, which is $C_j \times C_j$ matrix whose elements are given by:

  $$G_j^\phi(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

  The style reconstruction loss the the squared Frobenius norm fo the difference between the Gram matrices of the output and target images:

  $$\ell_{style}^{\phi,j} = \|G_j^\phi(\hat{y}) - G_j^\phi(y)\|_F^2.$$
- If we view $\phi_j(x)$ as being a collection of vectors of dimension $C_j$, then $G_j^\phi(x)$ is proportional to the uncentered covariance of the $C_j$ dimensional vectors. So, it captures information about which features tend to activate together.
- The Gram matrix can be computed efficiently by viewing $\phi_j(x)$ as a matrix $\psi$ of size $C_j \times H_j W_j$. We then have that:

  $$G_j^\phi(x) = \frac{\psi \psi^T}{C_j H_j W_j}.$$
- The paper also defines two simple loss functions:
  - The **pixel loss** simply computes the average square difference between corresponding pixels:

    $$\ell_{pixel}(\hat{y}, y) = \frac{\|\hat{y} - y\|_2^2}{CHW}.$$
  - The **total variation regularizer** constrains the image to change smoothly over space:

    $$\ell_{TV}(\hat{y}) = \sum_{w,h} \left( (\hat{y}_{w,h+1} - \hat{y}_{w,h})^2 + (\hat{y}_{w+1,h} - \hat{y}_{w,h})^2 \right)$$

# 3 Implementation

## 3.1 Style Transfer

- Style transfer is defined as finding the image $\hat{y}$ that minimizes the following loss function:

$$\hat{y} = \arg\min_{y} \left( \lambda_c \ell_{feat}^{\phi,j}(y, y_c) + \lambda_s \ell_{style}^{\phi,j}(y, y_s) + \lambda_{TV} \ell_{TV}(y) \right)$$

  where $y_c$ is the target content image, $y_s$ is the target style image, and $\lambda_c$, $\lambda_s$, $\lambda_{TV}$ are scalar hyperparameters.

- For the feature reconstruction loss, the paper uses the output of `relu3_3` of VGG-16. For the style reconstruction loss, the paper uses the outputs of `relu1_2`, `relu2_2`, `relu3_3`, and `relu4_3`.

## 3.2 Super Resolution

- The author trained models to minimize the feature reconstruction loss at layer `relu2_2`.

# References


[1] Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *CoRR abs/1508.06576* (2015).

[2] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR abs/1512.03385* (2015).

[3] Johnson, J., Alahi, A., and Li, F. Perceptual losses for real-time style transfer and super-resolution. *CoRR abs/1603.08155* (2016).

[4] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR* (2016).