MATH 462 FINAL REPORT
PERCEPTUAL LOSSES FOR REAL-TIME STYLE TRANSFER AND SUPER-RESOLUTION

SAINA KOUKPARI - MCGILL UNIVERSITY

1. Introduction

The loss function plays the most important role in machine learning problems - allowing for evaluation and improvements of the model. In the case of our paper, we study image transformation, where images are generated by minimizing a loss function. The paper finds that success in this task requires semantic reasoning about the input image. Using a pre-trained VGG network we can measure semantic similarity by comparing the activations of specific layers with the generated and ground truth results; capturing more broad features in early layers and digging into more refined details in the later layers. The paper discusses the strategy of perceptual loss for training feed-forward networks and compares the results to those of previous loss methods.

2. Motivation

The motivation lies in the advancements of image transformations, specifically that of style transfer and single-image super-resolution, where a system receives some input image and transforms it into an output image. We discuss and compare two loss functions, the per-pixel loss used in many recent methods, and the perceptual loss introduced in this paper. As we know, the loss function plays a large role in training a good model. Other aspects of machine learning problems, i.e. data, model and interpretation, are also considered in training, however the use of convolutions are known to do an excellent job at recognizing features of images, thus the improvement of our model relies on working towards finding improved loss functions.

In previous work, per-pixel loss between the output and ground-truth images is used to train feed-forward convolutional neural networks. However, this loss finds faults when we offset two identical images by one pixel. Shifting by a pixel would cause the loss to increase but the image itself would be perceptually the same, leading to false adjustments to the model. In order to avoid this, we wish to find different ways of enforcing similarities between two images, hence we introduce perceptual loss functions.

Defining and optimizing the perceptual loss function allows for the generation of high-quality images for image transformations. This is based on high-level features extracted from pre-trained networks, specifically the VGG-16 neural network which encodes information about the content of our image, providing perceptive capabilities and giving the loss function we want. Perceptual loss is able to better reconstruct fine details in image generation, as well as output similar qualitative results as previous works, with improvements in run time. Finding success in newly defined losses often mark advancements for future works, hence we find significant importance in constructing and understanding loss functions.

3. Main Ideas and Concepts

The ideas introduced in the paper provide a conceptual understanding of the problem of image transformation, while presenting the mathematical insight needed for the loss function. The algorithms, mainly that of the VGG-16 model, discussed are pulled from previous work; see further readings section. The paper focuses on the comparison of previous work in the search for improvements, specifically that of Gatys et al for style transfer and the SRCNN model for single-image super-resolution with PSNR and SSIM used as assessment of the output (see further readings)

3.1. **Style Transfer.** The first discussed task is that of style transfer where we combine the content of an input image with the style of another by jointly minimizing a feature reconstruction loss and a style reconstruction loss; these losses will be discussed in further length in the later sections. Based on features extracted from the pre-trained convolutional network, feed-forward networks are trained to solve the optimization problem. We will see how style transfer generates an image $\hat{y}$ that combines the content of a target content image y_c with the the style of a target style image y_s, in the methods and mathematics sections of this report.

3.2. **Single-Image Super-Resolution.** The second discussed task is that of single-image super-resolution where high-resolution images are generated from lower-resolution inputs. There are many possible magnitudes of solutions to an input image as fine details must be inferred where the content can be visually ambiguous. Previous work has succeeded using a three-layer convolutional neural network trained with a per-pixel Euclidean loss, however replacing the per-pixel loss with a perceptual loss gives rise to visually rewarding results. The paper provides insight on the advantages of perceptual loss on $\times 4$ and $\times 8$ super-resolution trained with the feature reconstruction loss to allow transfer of semantic knowledge from the pretrained loss network to the super-resolution network. The details of this procedure are discussed further in the methods and mathematics sections.

3.3. **Methods.** The system passes an input image through an image transformation network f_W, specifically a convolutional neural network that transforms the input to the output image $\hat{y}$ through the mapping $\hat{y} = f_W(x)$. The output is then pushed through the VGG-16 loss network, ϕ, which is pre-trained for image classification. This defines the perceptual loss functions that measures the differences in content and style between images. The system defines several loss functions $\ell_1, \ldots \ell_k$, more specifically, the image transformation network computes the difference between output $\hat{y}$ and target image y with the loss $\ell(\hat{y}, y)$, and the loss network defines a feature reconstruction loss $\left(\ell^{\phi}_{feat}\right)$ and a style reconstruction loss $\left(\ell^{\phi}_{style}\right)$ that measures the differences in content and style between images. For each input image x we have a content target y_c and a style target y_s. In the case of style transfer both losses are used; we train one network per style target where the input image $x = y_c$ and the output image $\hat{y}$ aims to combine the content with the style of y_s. For the case of single-image super-resolution it suffices to use the feature reconstruction loss; we train one network per super-resolution factor where the input image x is a low-resolution input and the content target y_c is the ground-truth high resolution image.

3.4. **System Architecture.** Following previous works architectures (see further readings) with minor modifications, the network uses strided and fractionally strided convolutions for downsampling and upsampling. This process divides the network into two parts where the downsampling network uses CNN structures to reduce the dimension of the data by abstracting the representations of the input, and the upsampling network takes the abstracted image representations and makes their spatial dimensions equal to that of the input image. Computationally, this network saves major convolutional costs for large sized networks due to downsampling. Downsampling provides further advantages for changing large parts of the image in a systematic manner.

Composing the network are five residual blocks, i.e. stacks of layers set so that the output of each layer is added to a deeper layer in the block, and non-residual layers that connect to a spatial batch normalization and ReLU nonlinearities. Residual connections allow the network to easily learn the identify function which helps determine whether the output image shares its structure with the input image. Finally, the output layer ensures image pixels in the range [0, 255] through scaled tanh.

When tackling style transfer, we have color images of shape 3 $\times$ 256 $\times$ 256 for both the input and output. The network uses two stride-2 convolutions to downsample the input followed by several residual blocks and two convolutional layers with stride $1/2$ to upsample. For single-image super-resolution, the input is a low-resolution patch of shape $3 \times 288/f \times 288/f$ where f is the upsampling factor, and the output is a high-resolution image patch of shape 3 $\times$ 288 $\times$ 288. The network uses several residual blocks which are then followed by $\log_2 f$ convolutional layers with stride $1/2$. The use of fractionally-strided convolution in both tasks allows for upsampling to be learned along with the rest of the network.

3.5. **Results.** Given the task of style transfer, the paper compares their results with the results of Gatys et al as a baseline. Performing feature and style reconstructions at layers j and J, an image $\hat{y}$ was generated by solving the following equation using the optimization algorithm L-BFGS, with scalars λ and y initialized as white noise,

$$\hat{y} = argmin_y \lambda_c \ell^{\phi,j}_{feat}(y, y_c) + \lambda_s \ell^{\phi,J}_{style}(y, y_s) + \lambda TV \ell_{TV}(y)$$

Training consists of resizing each of the training images to a size of 256 $\times$ 256 and training the networks with a batch size of 4 for 40,000 iterations. The feature reconstruction loss is computed at layer relu2_2 and the style reconstruction loss at layers relu1_2, relu2_2, relu3_3, and relu4_3 of the VGG-16 loss network. We find positive results, both qualitatively and quantitatively. Comparing the perceptual loss method to the baseline results, the hyper-parameters λ_c, λ_s and λ_{TV} are identical. It was further found that the trained style transfer network is aware of the semantic contents of images, which derives from the fact that the VGG-16 loss network has features that are selective for people and animals; features which are transferred into the discussed network where theses objects are more distinguishable after style conversion. Comparing the minimized equation as above to the baseline minimization of the same equation found that the model using perceptual loss achieves a loss comparable to 50 to 100 iterations of the baseline method and has a run time much faster than that of previous work results.

Let us now compare the results for single-image super-resolution. In this case we wish to show the qualitative difference between models trained with per-pixel loss and the perceptual feature reconstruction loss. Regarding per-pixel loss, there are problems around the ambiguity of increasing the resolution of an image. Using feature reconstruction loss allows for the transfer of semantic knowledge from the pre-trained loss network to the super-resolution network which resolves this issue. The model is trained to perform on $\times 4$ and $\times 8$ super-resolution with 288×288 patches from a training set images where we wish to minimize the loss at layer relu2_ 2 in the VGG-16 network. The low-resolution inputs are obtained by blurring images with a Gaussian kernel of $\sigma = 1.0$ width and downsampling with bicubic interpolation.

The results are compared with the results of the SRCNN model - a three-layer convolutional network trained to minimize per-pixel loss - as a baseline. The feature reconstruction loss model proposed in the paper does a very good job at reconstructing sharp edges and fine details, suggesting that the model is more prone to understanding image semantics. Contrarily, the per-pixel loss gives fewer visual details.

4. Mathematical Analysis

We will now look at the above information through a mathematical lens. For more details on the specifics not mentioned in the following section, see further references. The following apply to the model construction as stated previously.

We consider the deep residual convolutional neural network parameterized by weights, transforming the input as $f_W(x) = \hat{y}$. The network is trained using stochastic gradient descent to minimize a weighted combination (W^*) of loss functions where we compute the difference between the output image and a target image by the scalar value for $\ell_i(\hat{y}, y)$.

$$W^* = argmin_W \mathbb{E}_{x,\{y_i\}} \left(\sum_{i=1} \lambda_i \ell_i(f_W(x), y_i) \right)$$

The paper defines two types of perceptual loss functions, the feature reconstruction loss, and the style reconstruction loss. Since the loss functions make use of the pre-trained VGG-16 loss network for classification, these loss functions are also deep convolutional neural networks.

4.1. **Feature Reconstruction Loss.**

This loss encourages the pixels of the output image $\hat{y}$ to have similar feature representations as computed by the VGG-16 loss network. Recall that we labeled the loss network with ϕ. Then we let $\phi_j(x)$ define the activations of the jth layer of the network when processing the image x. When such a layer j is convolutional, $\phi_j(x)$ will be a feature map of shape $C_j \times H_j \times W_j$. The loss is then defined by the euclidean distance between feature representations,

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

An output image $\hat{y}$ that minimizes the above loss for early layers tends to produce images that are visually identical to y. We then reconstruct from higher layers to preserve the image content and overall spatial structure but allow for style change by not preserving the color, texture, and exact shape. This reinforces the output image to be perceptually similar to the target image y while allowing for stylistic modifications where the images won't match exactly. The loss then penalizes the output image when it deviates in content from the target.

4.2. **Style Reconstruction Loss.** This loss defines $\phi_j(x)$ as above and further defines the $C_j \times C_j$ gram matrix $G_j^\phi(x)$ which captures information about which features tend to activate together.

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

The loss is then defined by the squared Frobenius norm between the Gram matrices of the output and target images,

$$\ell_{style}^{\phi,j}(\hat{y}, y) = \|G_j^\phi(\hat{y}) - G_j^\phi(y)\|_F^2$$

An output image that minimizes the style reconstruction loss, preserves stylistic features from the target image, but doesn't preserve the spatial structures of the image. As before, we reconstruct from higher layers to transfer larger-scale structures from the target image. The loss then, unlike the previous loss, penalizes differences in style, i.e. colors, textures, common patterns, etc. We allow for style reconstruction on a set of layers J by taking the loss to be the sum of losses for layers $j \in J$.

5. Discussion and Limitations

This paper introduces a new loss in the hopes to improve current work on image transformation and provides insight on the importance of a good loss function for machine learning problems. Training a feed-forward transformation network with the perceptual loss functions applied to the specific tasks of style transfer and single-image super-resolution was proven to have success. The model worked on style transfer by achieving comparable performance to that of previous work while significantly improving the speed of the network compared to existing methods. In the latter task, the perceptual loss allowed the model to better reconstruct fine details and edges.

While relatively new, perceptual loss has already led to innovations in new works such as that of the Generative adversarial networks (GANs) which is used for image and video generation. The authors of the paper wish to find even more use for perceptual losses as well as considering other possible loss networks that examine different types of semantic knowledge which give rise to advancements in image transformation networks.

6. Further Readings

The paper compares perceptual loss for style transformation with the per-pixel loss results of Gatys et el:

https://arxiv.org/pdf/1508.06576.pdf

The results for single-image super-resolution is compared to that of the SRCNN model:

https://arxiv.org/pdf/1501.00092.pdf

We discusses the concepts of PSNR and SSIM for super-resolution:

https://www.cns.nyu.edu/pub/lcv/wang03-preprint.pdf

The network makes direct use of the VGG-16 network: https://neurohive.io/en/popular-networks/vgg16/ and pulls from the architectures in the following literature:

https://arxiv.org/pdf/1511.06434.pdf
https://arxiv.org/pdf/1512.03385.pdf
http://torch.ch/blog/2016/02/04/resnets.html

More mathematics surrounding the regularization of output can be found in the following link:

https://ieeexplore.ieee.org/document/1510697