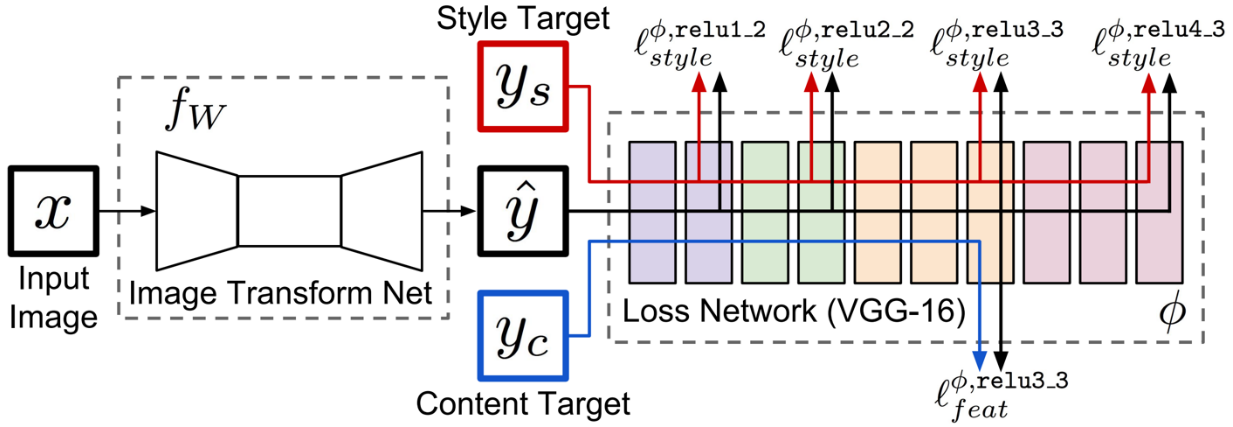
With the introduction of style transfer by Gatys et al and other papers inspired from it, two novel image transformation techniques based on a per-pixel loss optimization and perceptual similarity were created. However, these two techniques had major drawbacks that hindered usability. Per-pixel based optimization is fast but could not recreate high level features which represented perceived similarity, and perceptual losses are slow, but are able to learn high level features. Johnson et al provide a middle ground, combining the two techniques attempting to create a fast and perceptual image transformation network.



Style Target
y_s
$\ell_{style}^{\phi,\mathrm{relu1\_2}}$
$\ell_{style}^{\phi,\mathrm{relu2\_2}}$
$\ell_{style}^{\phi,\mathrm{relu3\_3}}$
$\ell_{style}^{\phi,\mathrm{relu4\_3}}$
f_W
x
Input Image
Image Transform Net
$\hat{y}$
Loss Network (VGG-16)
ϕ
y_c
Content Target
$\ell_{feat}^{\phi,\mathrm{relu3\_3}}$

Johnson et al put forward algorithm consisting of an image transfer net and a loss network which is a pretrained VGG-16 network. The image transformation network is a convolutional network meant to down sample the image and up sample the image with a specific style. The image transformation network's output is the stylized image. The stylized image, content image, and style image are passed into the VGG-16 and at specific activation layers, the values are compared for content loss and style loss. For style loss, layers 1, 2, 3 and 4 are pulled from the VGG-16 network from the style image and transformed image outputs. These outputs are used to form gram matrices which are then compared using mean squared error. Using the gram matrices with the activation function outputs allows for comparison of relationships between all features in the style image and transformed image. For the content loss, the mean squared error is taken between the transformed image's and content image's 3rd activation layer. This will allow for a per-feature content loss. This network will train on a single style image and multiple content images and each iteration of the training loop will cycle through a new content image and calculate the losses between the transformed image, the style image, and the content image. As the model is trained, the image transformation network will up sample any image with the requested style. This creates a fast style transfer inference as the model just needs to take a content image as an input and it will transfer the style image's style onto the content image by using the image transform net. However, whenever a new style is requested, a whole new training process must be done.