**Joint Bilateral Learning for Real-time Universal Photorealistic Style Transfer**



**Submitters:**

Eylon Ben Simon

Shiraz Karov

Gal Toledano

Saar Victor

**The process of the project** **and the project flow:**

We first read the article, understood bits of it. Then we went ahead and learned about neural network and convolutional neural network. We then read the article again, saw some YouTube videos explaining the general idea and began to translate it using our own words.

During this process, we found a code that implements this whole thing. We went over it and understood the key components about it. We ran the code, trained the model and tried this whole network ourselves.

**The problem:**

Transfer a styled photo and paste it onto an input image, so that the output image will be at a high quality which will be equal to the notion that the output photo was taken from a high-resolution camera.

**The main goals:**

The goals are- Universal photorealistic style transfer, Inference implementation that runs at 4K on a mobile phone and Inference implementation that runs in real time.

**The solution:**

To combine two other methods specializing in photos manipulation. The first one, Adain which is used to transfer an abstract style onto a photo by transferring feature statistics and generalizing unseen inputs. The second one, HDRNet, which detects edges to an impressive extent, and the network itself is quite small. In addition to both methods, we alter the loss function to be more punishing so the style loss will be minimal.

First, we want to learn a multi-scale model the joint distribution between style and content and style features. Once we achieve this distribution, we can predict an affine bilateral grid. Instead of using strided convolutional layers to directly learn from pixel data, we use a VGG-19 pretrained network from a recent work discovered, to extract low-level features from both images at four scales (conv1\_1, con2\_1, conv3\_1 and conv4\_1). With a sequence of splatting blocks inspired by StyleGAN architecture, we process these multi-resolution feature maps. Starting from the finest level, each splatting block applies a stride-2 weight-sharing convolutional layer to both content and style features, halving spatial resolution while doubling the number of channels. By using the shared-weight constraint, we allow the following Adain layer to learn the joint style / content distribution without correspondence supervision.

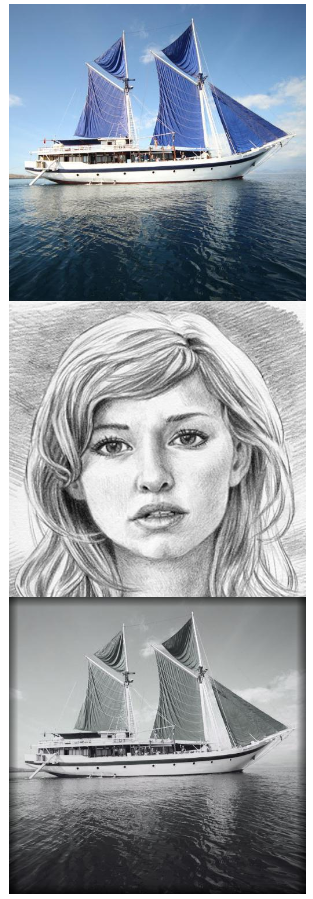
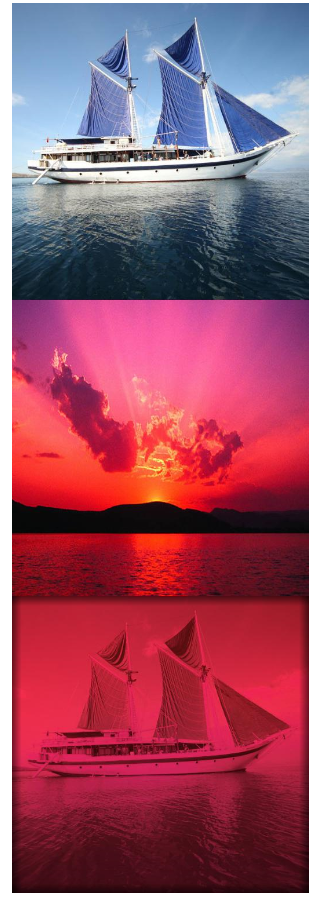
We append the content feature map to the similarly Adain – aligned feature maps from the pretrained VGG-19 layer of the same resolution once this specific content feature map is rescaled. Now the content feature map contains from channel, and we use a **stride-1 convolutional layer** to select only the relevant channels between learn-and-normalized vs. pretrained-and-normalized features. We use **three splatting blocks** in our architecture, corresponding to the finest resolution layers of the selected VGG features. Since this component of the network effectively learns the bilateral-space content features based on its corresponding style, it can be thought of as learned style-based splatting.

Joint Bilateral Learning. Because we have aligned-to-style content features in bilateral space, we aim to learn an affine bilateral grid that encodes transformation that locally captures style and is aware of scene semantics. Just like in HDRNet, we split the network into **two asymmetric paths: a fully convolutional local path** that learns local color transforms and thereby sets the grid resolution, and a **global path** which consists of both convolutional and fully connected layers, that learns a summary of the scene and helps spatially regulate the transforms. **The local path is made of two stride-1 convolutional layers**, keeping the number of features and spatial resolution constant. This provides enough depth to learn local affine transforms without letting its receptive field grow too large. We use a small network to learn a global notion of scene category because we want to perform universal style transfer without any explicit notion of semantics. The second path**, the fully connected one, consists of two stride-2 convolutional layers** in order to reduce the resolution even more, followed by **four fully-connected layers** to produce a 64-element vector “summary”. We then append this summary at each x, y spatial output from the local path and **use a 1x1 convolutional layer** to reduce the final output to 96 channels. These 96 channels can be reshaped into an 8 “luma bins” that separate edges, each sorting a 3x4 affine transform. We use the ReLU activation after each layer except the final 1x1 fusion layer, and zero-padding for all convolutional layers.

**The obtained results:**

The photos we received by using this architecture are exactly how we imagined them to be. The edited photos we got look as if they were taken from a professional DSLR camera.

Furthermore, after the network was trained, we got instant results.

**Conclusions:**

We saw the technique presented in this article is significantly faster than the state of the art, degrades gracefully even in extreme cases and runs in real-time on a smartphone. We also agree with the authors about the fact that it's fast runtime and robustness will lead to practical applications in mobile photography. This work can be even more improved by reducing network size and figure out how to relax the photorealism constraint to generate a continuum between photorealistic and abstract art.