# CS 229 Final Project: Neural Style Transfer

Jixun Ding, Fangze Liu, Wen Wang

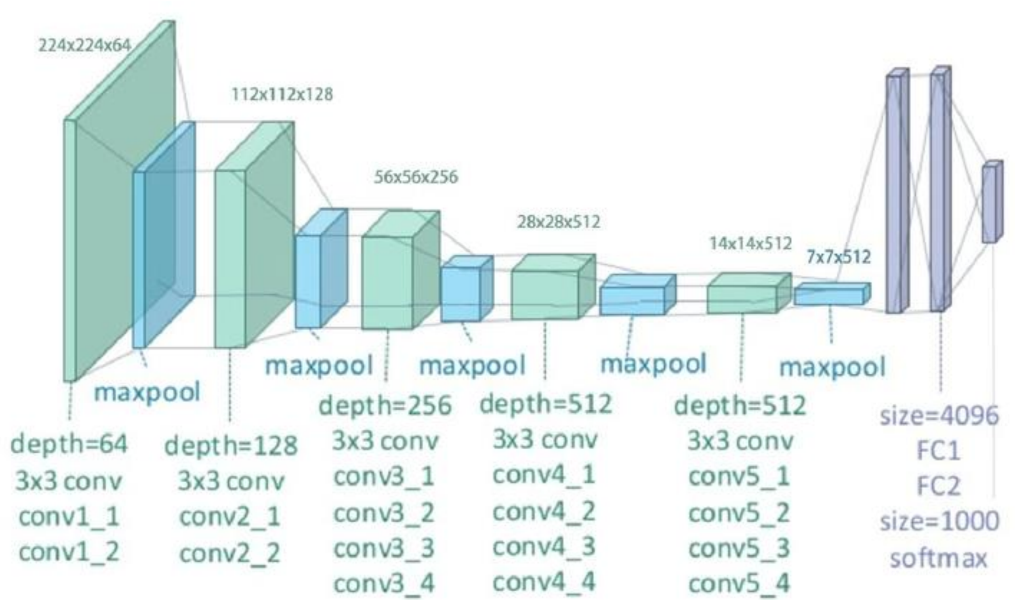
{xunger08,fangzel,wenwang}@stanford.edu

## Introduction

Given a pair of images, the process of combining the "style" of one image with the "content" of the other image to create a piece of synthetic artwork is known as **style transfer.** Gatys et al. (2015) demonstrated a generalized style transfer technique by exploiting feature responses from a pre-trained convolutional neural network, opening up the field of **neural style transfer**. In this work, we implement Gatys' algorithm to produce some synthetic artwork. We also implement a spatial control extension to Gatys' algorithm and check how the spatial control extension improves output image quality.

## Dataset and Features

In our implementation of Gatys' method we use weights of VGG-19 (Simonyan & Zisserman, 2014), a 19-layer deep convolutional neural network which has been pre-trained on the ImageNet dataset.




As content images, we use personal photos, stock images, as well as a few images in the benchmark dataset *NPRgeneral* proposed by Mould & Rosin (2016). As style images, we use well known artworks, all of which are in the public domain.

Each pair of style and content images are cropped and resized to be of matching aspect ratios and sizes, and normalized by subtracting the RGB average of the ImageNet dataset.

## Method

The style representation of an image $x$ at layer $l$ is represented by the Gram matrix

$$\mathcal{G}\left(F^{[l]}(\vec{x})\right) = \left[F^{[l]}(\vec{x})\right]\left[F^{[l]}(\vec{x})\right]^T$$

To transfer the style of an artwork $a$ onto a photograph $p$ and produce a synthesized image $x$, we initialize a random image $x$, and minimize the total loss function defined as a linear combination of content loss, style loss, and total variational loss

$$\boxed{\mathcal{L}_{\text{total}}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{\text{c}}(\vec{p},\vec{x}) + \beta\mathcal{L}_{\text{s}}(\vec{a},\vec{x}) + \gamma\mathcal{L}_{\text{v}}(\vec{x})}$$

$$\boxed{\mathcal{L}_{\text{c}} = \sum_{l\in\mathcal{C}} w_{\text{c}}^{[l]} \left\| F^{[l]}(\vec{p}) - F^{[l]}(\vec{x}) \right\|_2^2}$$

$$\boxed{\mathcal{L}_{\text{s}} = \sum_{l\in\mathcal{S}} w_{\text{s}}^{[l]} \left\| \mathcal{G}\left(F^{[l]}(\vec{a})\right) - \mathcal{G}\left(F^{[l]}(\vec{x})\right) \right\|_2^2}$$

$$\boxed{\mathcal{L}_{\text{v}} = \sum_{i,j} \left[ \left|\vec{x}_{(i,j)} - \vec{x}_{(i+1,j)}\right| + \left|\vec{x}_{(i,j)} - \vec{x}_{(i,j+1)}\right| \right]}$$

Our addition

If spatial control is applied, both style and content images are divided into $K$ regions and the style loss above is replaced with normalized style loss:

$$\boxed{\mathcal{L}_{\text{s}} = \sum_{l\in\mathcal{S}} \sum_{k=1}^{K} \frac{w_{\text{s}}^{[l]}}{K} \left\| \frac{1}{N_{a_k^{[l]}}} \mathcal{G}\left(F^{[l]}(\vec{a_k})\right) - \frac{1}{N_{x_k^{[l]}}} \mathcal{G}\left(F^{[l]}(\vec{x_k})\right) \right\|_2^2}$$

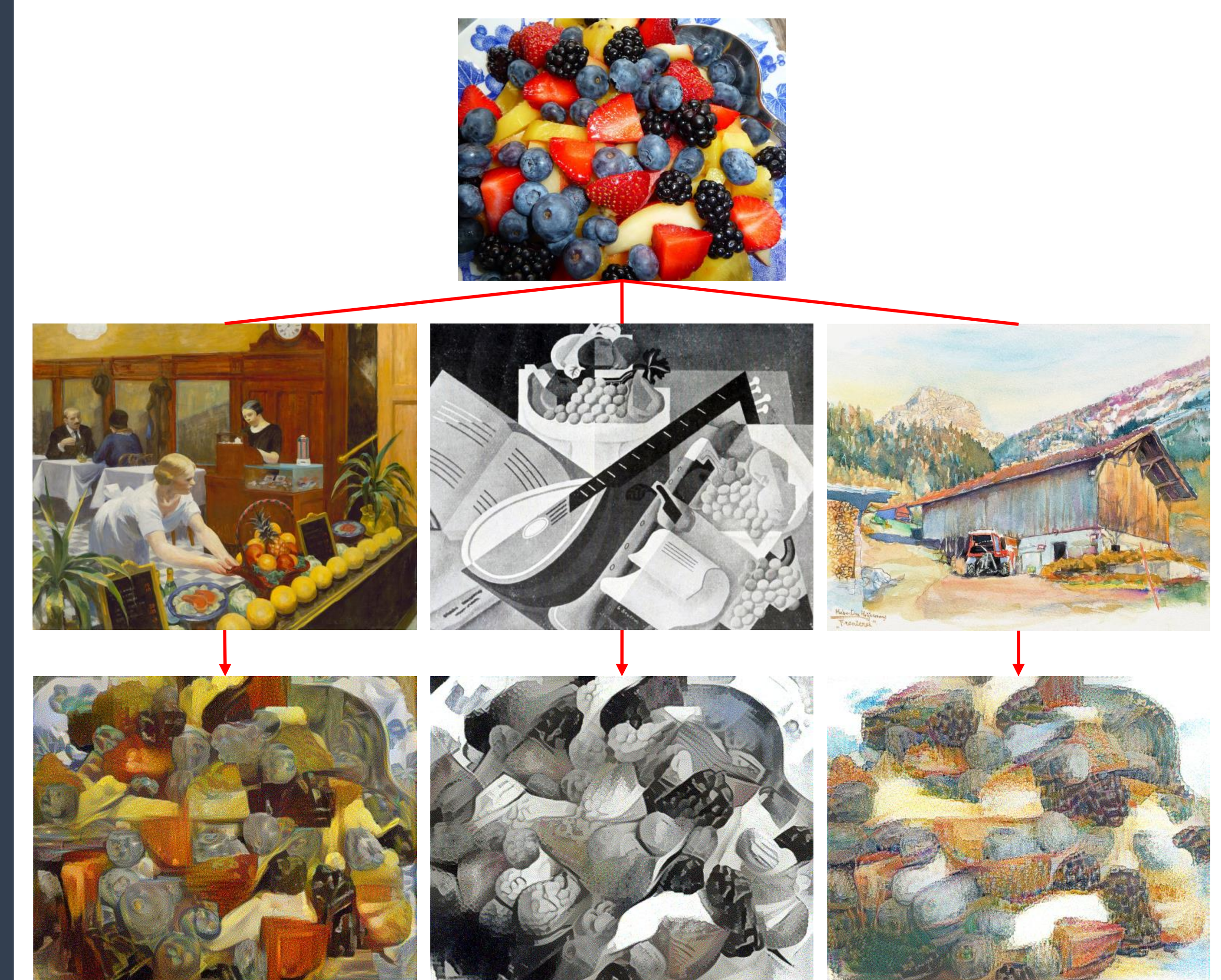
TensorFlow and Adam optimizer is used to optimize the loss function defined above.

## Results

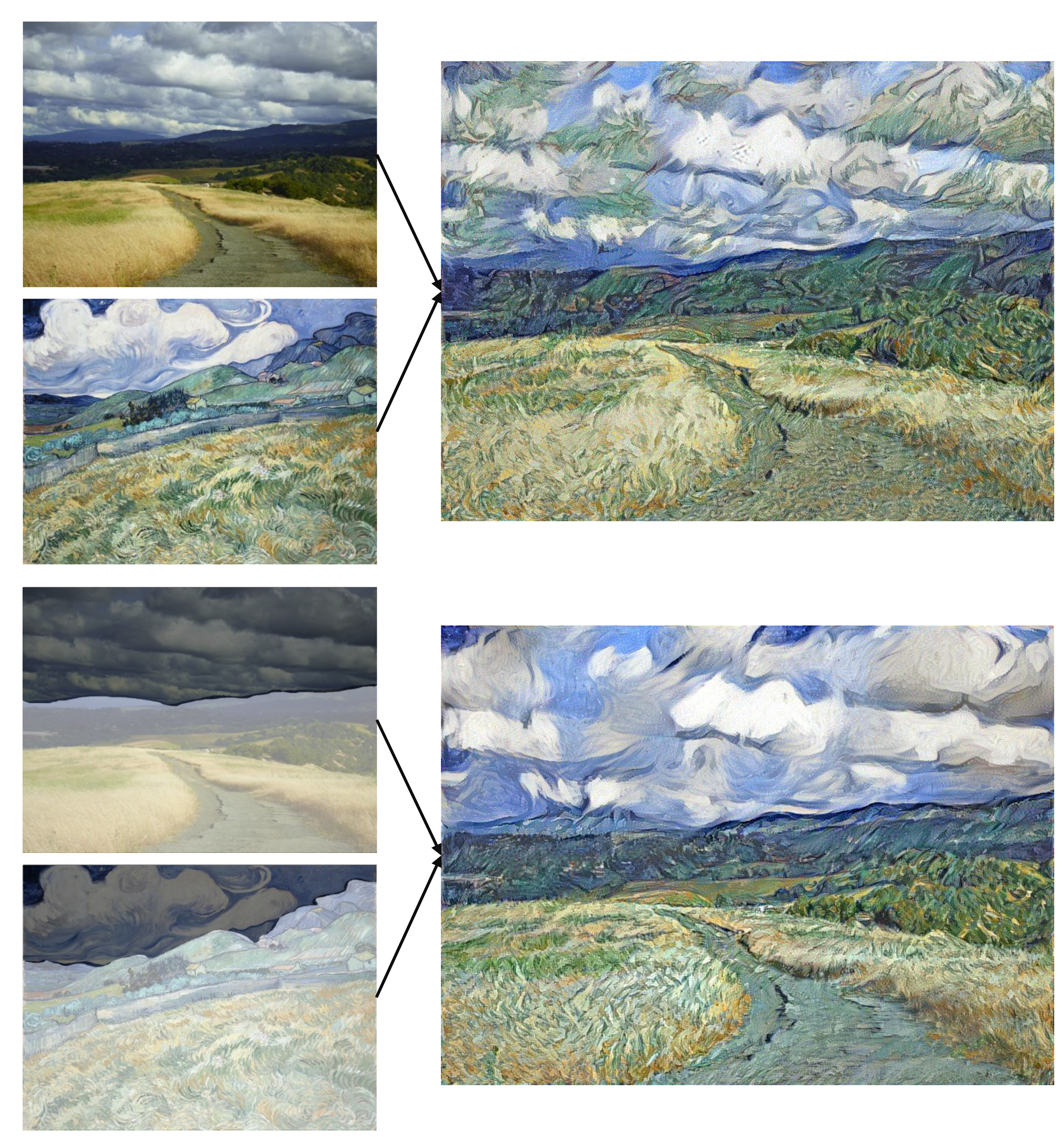
**Difficult case**

## Results, cont.

**One content image, different styles**:



**Spatial control:**



## Discussion

- We implemented Gatys' neural style transfer algorithm, which is capable of arbitrary style transfer and produces high quality output, but has high computational cost (>1 hour CPU runtime for 500x500 image, 300 iterations). Our output images are limited in resolution and contains residual noise from random image initialization. Additionally, each pair of {style, content} images requires custom hyperparameter tuning in order to obtain the most visually impressive result.
- Our spatial control method requires the user to generate two additional input masks. It enables us to obtain higher quality output for some examples, but introduces distortion at the mask boundaries, and fails if the art style is too photorealistic.

## Future Work

Potentially interesting future work of this project includes:

1. Implement additional control options, such as color control, feature size control, and blending multiple styles.
2. Implement a real-time style transfer algorithm and compare its performance with Gatys method.
3. Explore options for automatic hyperparameter tuning.

## References

- Gatys, L. A., Ecker, A. S., and Bethge, M. *A neural algorithm of artistic style*. CoRR, abs/1508.06576, 2015.
- Champandard, A. J. Semantic style transfer and turning two-bit doodles into fine artworks. CoRR,abs/1603.01768, 2016.
- Gatys, L. A., Ecker, A. S., Bethge, M., Hertzmann, A., and Shechtman, E. *Controlling perceptual factors in neural style transfer*. CoRR, abs/1611.07865, 2016.
- Simonyan, K. and Zisserman, A. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.