Image Style Transfer Using CNN

Team 17
Erdong Liao, Jiadao Zou, Tianqi Zhu
{exl170630, jxz172230, txz160930}@utdallas.edu
Dec 03, 2018

Introduction

- Rendering the semantic content of an image in different styles is a difficult image processing task.

- Ideally, a style transfer algorithm should be able to extract the semantic image content from the target image and then inform a texture transfer procedure to render the semantic content of the target image in the style of the source image.

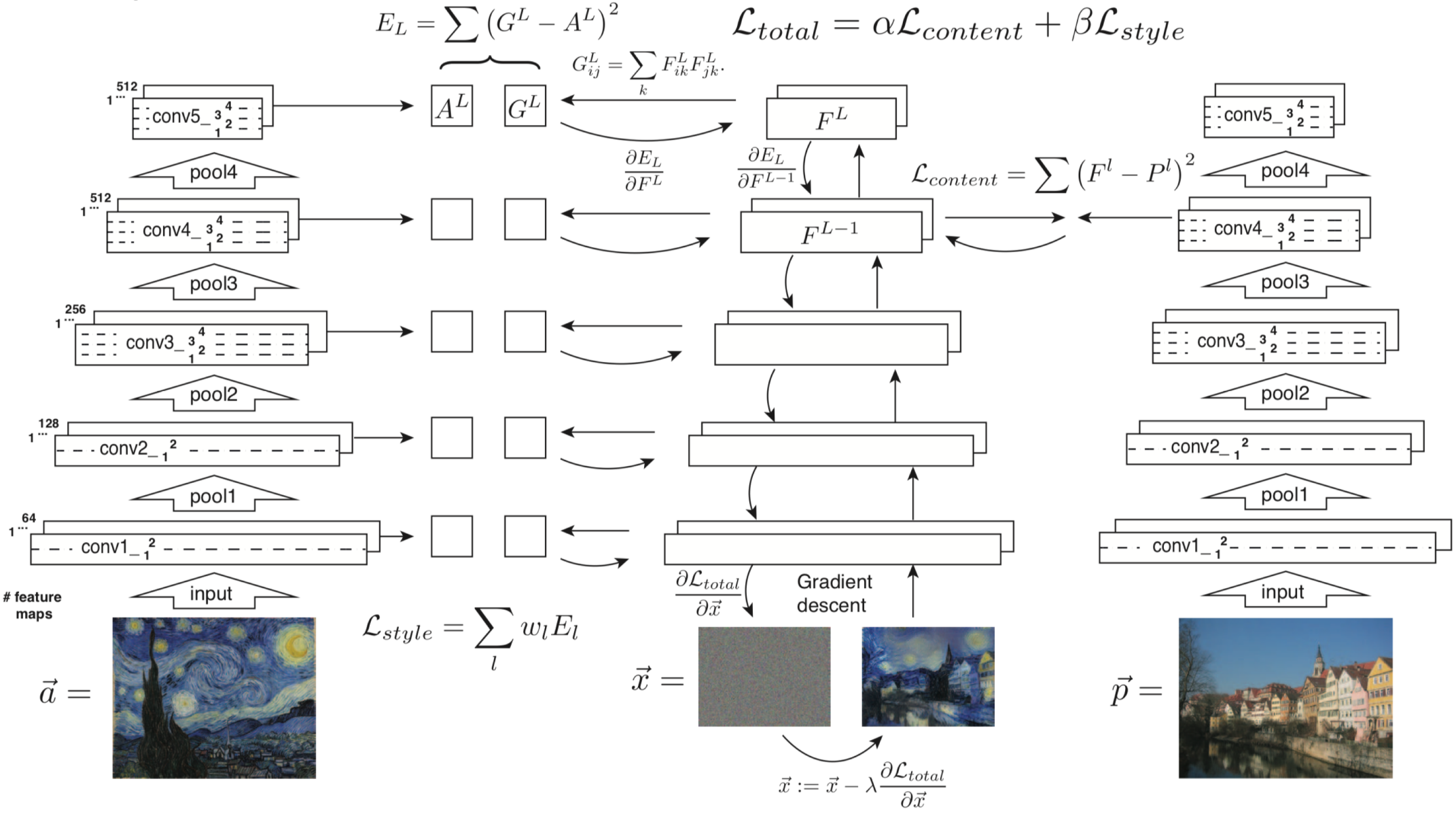
Related Work

- Deep Art: Image Style Transfer Using Convolutional Neural Networks
  - Using CNN approach to separately extract semantic content from original photograph and style of artwork.
  - Very computationally inefficient. The stylized image is, in fact, obtained by iterative optimization until it matches the desired statistics.
- Instance Normalization: The Missing Ingredient for Fast Stylization
  - The change is limited to swapping batch normalization with instance normalization, and to apply the latter both at training and testing times
- ...
- Include
  - CNN extracts features
  - Different Normalization

Implementation

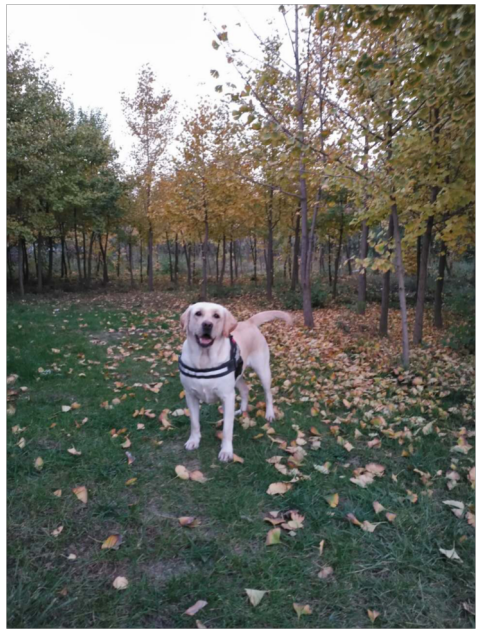
- Firstly we build the network structure as *Image Style Transfer Using Convolutional Neural Networks* paper implies using VGG
  - Using Tensorflow.
  - Lower Layer simply reproduce the exact pixel values of the original image.
  - Higher Layers focus on object and its arrangement but do not constrain the exact pixel values.
  - Start from a random white noise image, iterate on loss function and dynamically assign weights on different layers to reconstruct the image containing features from both photograph and artwork.

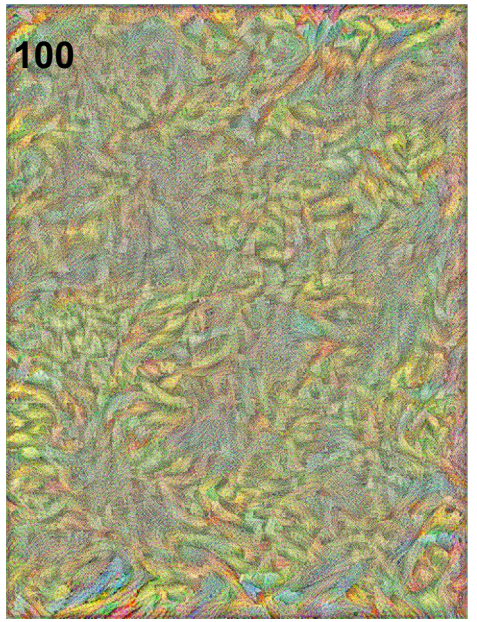
Deep Art Structure

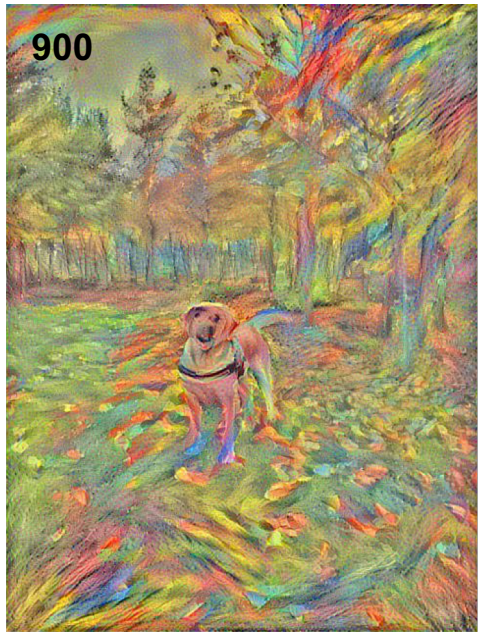


Results

- 2 hours, 1800 iterations
- Include
  - Original Photograph
  - Style-guide Artwork







Improvement

- Really Computationally Inefficient.
- Why? iterative optimization until it matches the desired statistics.
- What we want to accomplish
  - Speed: faster
  - Performance: reconstruct same quality images in term of visually appealing
- Ideas on how to reach this purpose
  - Hardware: CPU, GPU, trivial
  - Optimize the network structure and algorithm
  - Have a large image database, can quickly look up the image similar to input image, use the features have been extracted
  - image preprocessing + different masks + feature scaling

Task

Photograph

Artwork

Contributions

- Use Otsu's method to find mask, then use mask to determine border and edges.
  - First convert image to grayscale image.
  - *Y = 0.299 R + 0.587 G + 0.114 B*
  - Find the mask with pixel values larger than Otsu threshold.
  - minimizes the intra-class variance (the variance within the class)

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

  - Remove small object(noise).
  - Define convolution core to filter image as different purpose
    - Small core is used to preserve the image's basic structure.
    - Medium core is user to preserve the content inside of the image.
    - Large core is used to remove outliers.

Contributions

- Use Harris Corner to detect image's corners and scale it up as mask, can reduce computation effectively.
  - Harris Corner Detector was first introduced by Chris Harris and Mike Stephens in 1988 upon the improvement of Moravec's corner detector

$$f(x,y) \approx \sum_{(x,y)\in W} (I_x(x,y)\Delta x + I_y(x,y)\Delta y)^2,$$

$$\lambda_{min} \approx \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)} = \frac{det(M)}{trace(M)}$$

  - Dilate the features with different scales so that the original photograph can combine with style-guide artwork in different extent.

Contributions

- Use statistics method(mean, variance)
  - Produce two masks based on pixel values' mean and standard variance.

  - The masks represent the typical pixel value of original photograph.
    - mask1 = I_img > mean_img + (std_img * std_factor)
    - mask2 = I_img < mean_img - (std_img * std_factor)
  - Combine two masks together and clip to RGB scale.

Results

Otsu's Method(4.2s)

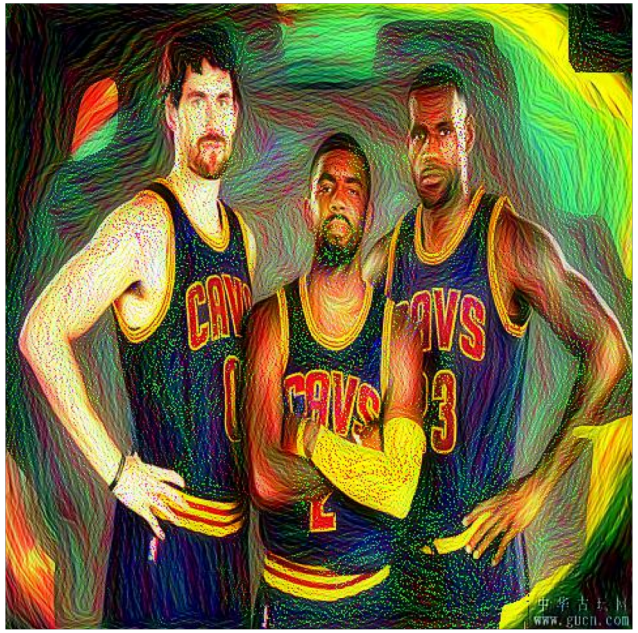
Harris Corner Method(7.3s)

Statistic Method(4.8s)

Results

Statistic Method

Contrast Enhanced



Sharpness Enhanced

Brightness Enhanced

Summary

- Pros:
  - Improve from hours-level style transfer network to seconds-level style transfer network.
  - Maintain a relatively good quality with respect to visually appealing.
  - Preprocessing step(sharpness, contrast) can enhance the feature extracted by different masking method.

- Cons:
  - The universality need to be improved, for some image's objects pixel values similar to background pixel values, it may result in bad output.
  - Needs further work to pipeline all processes without manually tuning too many parameters(GUI?).
  - Network structure is worth improving.

Future Work

- Make result more visually appealing.
- Object(s) Transfer instead of transferring whole image.
- What about applying Style Transfer to videos?
- Combine instance normalization with our work.
- etc...
-

References

- Image Style Transfer Using Convolutional Neural Networks
  - https://arxiv.org/pdf/1508.06576.pdf
- Instance Normalization: The Missing Ingredient for Fast Stylization
  - https://arxiv.org/pdf/1607.08022.pdf
- Perceptual Losses for Real-Time Style Transfer and Super-Resolution
  - https://arxiv.org/pdf/1603.08155.pdf

Backup

- Something we kindly want you know

- Include
  - Set up Caffe environment taking four whole days (module version compatibility, lib dependencies, Mysticism, etc,.)
  - Want to develop a GUI for playing around different parameters with respect to convolution kernel size/threshold, feature scaling factor. Not finished because of time limit.