Neural Style Transfer Artist Aggregation Technique

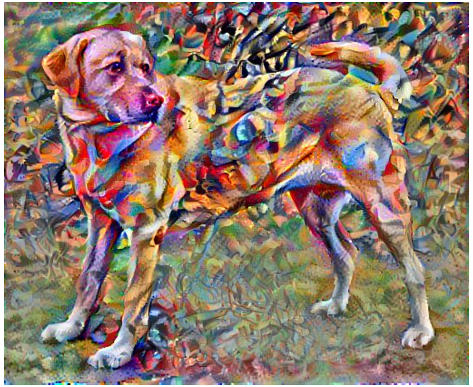
Seth Singson-Robbins
MS of Data Science
Fordham University
Deep Learning

What we are going to cover

- What is Neural Style Transfer?
- Project Overview
- Hyperparameters
- Results Analysis
- Conclusion

Neural Style Transfer - Overview

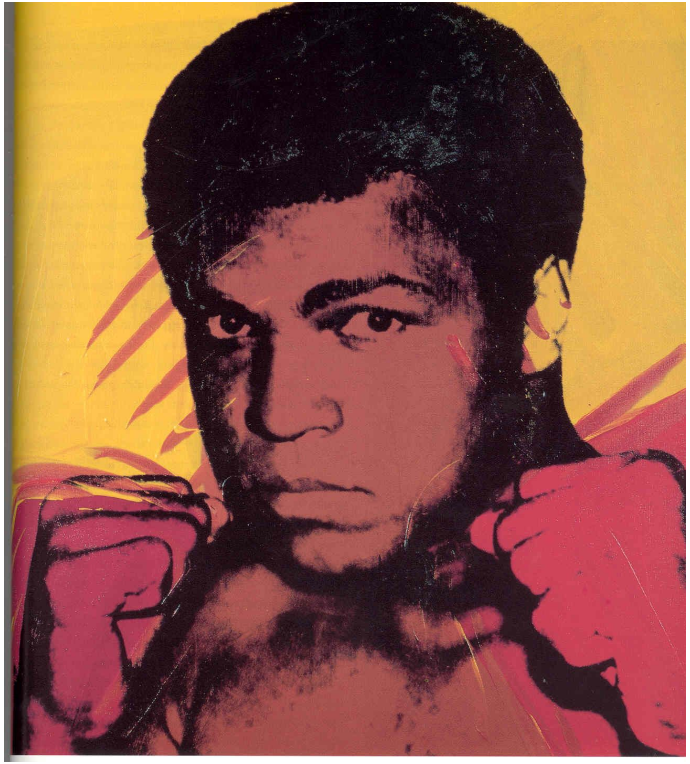
- Neural Style Transfer
  - Based on paper ‘A Neural Algorithm of Artistic Style’ by Leon Gatyz
- The model
  - Convolutional Neural Networks
  - Iterate changes to the content image and backpropagating the losses with the image and loss to get the final image.
- Uses the style of the style image to change the content image to match the style of the new image
  - Combination of style loss and content loss



Project Overview

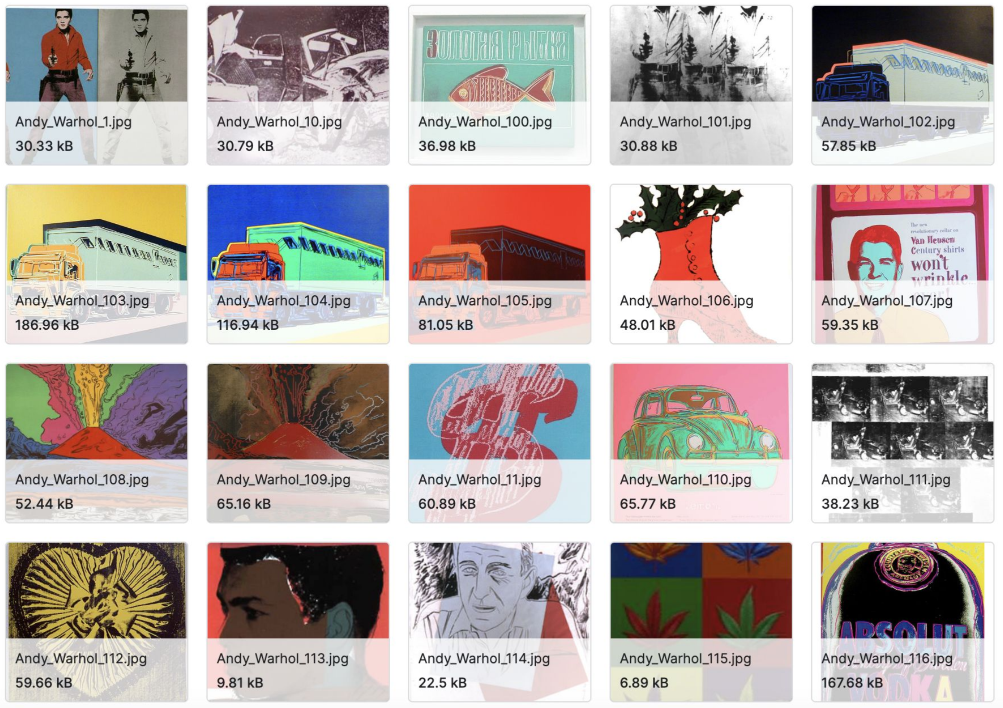
The Dataset

- Best Artworks of All Time (Kaggle)
  - 50 most influential artists of all time
  - Varying time periods and art styles
- Information of the Artists in a separate table
- 8,446 images in total



Project Iteration

- Original Model
  - Only looks at one style image to compare the generated image to calculate a style loss
- This model
  - Uses an array of images to derive the style loss
  - Aggregation of the loss
- This paper will use 30 images from several artists for the loss function of training and testing



Training and Test Image Selection

- Style Images
  - From 6 Artists dataset and the first 30 images of these datasets
  - Marc Chagall, Salvador Dali, Claude Monet, Pablo Picasso, Diego Rivera, Andy Warhol
- Content Images
  - 10 randomly selected images for training
    - 60 training examples
    - Analysis will be done by taking an average of the total loss (style + content) to pick quantitative hyperparameters
  - 5 randomly selected images for testing
    - 30 test examples for model analysis

Image Model Preprocessing

- Image resizing
    - All images are resized to 224 x 224 to fit into models
    - For test, they are resized again to be the same dimensions as original image
- RGB Transformation
    - Turns them into a readable format for the models
    - RGB - Turns each pixel into equivalent mapping of 3 colors
        - Red, Green, Blue from 1 to 255 to represent color of the pixel
        - Divided by 255 so normalized between 0 and 1 for the pixel colors
- Tensorflow
    - Turned into tensorflow model so all images in one dataset
    - Final dataset size (x, 1, 224, 224, 3) with x being the number of images

Hyperparameters

Hyperparameters Overview

- Difficulties of this study -> Unsupervised Models
    - Individualistic and varies greatly by different hyperparameters
- Quantitative Models
    - Optimizers
        - Which has the lowest and most efficient loss values
    - Epochs
        - EarlyStopping when average loss change between epochs less than 0.1% improvement
        - Or 100 epochs, whichever happens earlier
- Qualitative Models
    - Pretrained Models - VGG, ResNet, Inception, MobileNet
        - Each pretrained model do not have comparable content and style losses due to different layers
    - Loss Weights - Style Aggregation - Simple, Squared, Logged
    - 12 Qualitative Models to analyze (4 Pre Trained Models x 3 Style Aggregations)

Pretrained Models

- All models derived from the tensorflow package and are the main image CNN models
- Layers for Style and Content were picked based on location in the model and being CNN layers
- VGG
  - Model used in original model
  - Industry standard for image
- Other Models
  - ResNet
  - Inception
  - MobileNet

Optimization Models

- Array of standard optimization models for CNN
- Optimization Hyperparameters
  - Adam
    - What was used in the tensorflow example
  - Stochastic Gradient Descent
  - RMSProp

Loss Function

- Old Model - Style + Content Loss
    - Content Loss
        - MSE of Feature Maps between generated and original content images
        - Usually on the later layers of the CNN/pretrained model

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{l} - P_{ij}^{l}\right)^2$$

    - Style Loss
        - Gram Matrix - mathematical representation looking at correlations using feature maps and vectorization of the datasets
        - Usually looks at an array (5 for this paper) of early and middle layers of the model

$$J(S,G) = \frac{1}{(2 H^l W^l C^l)^2} \sum_k \sum_{k'} \left(Gkk'^{[l][S]} - Gkk'^{[l][G]}\right)$$

Loss Function

- Style Loss Aggregation
    - Style loss needs to be aggregated for each photo
    - 3 Options
        - Simple - Add up all the style losses, divided by number of images for the style loss
        - Squared - Square of each style loss, add up the images, square root for the style loss
        - Logged - Log of each image, add up the images, and exponential for the style loss

Simple Style Loss

$$\frac{\sum_1^n L_s^i}{n}$$

Squared Style Loss

$$\sqrt{\frac{\sum_1^n (L_s^i)^2}{n}}$$

Logged Style Loss

$$exp(\frac{\sum_1^n log(L_s^i)}{n})$$

Results Analysis

Quantitative Hyperparameters

- Optimizers
  - Most optimizers had clear delineations in their average performance overtime
  - Most had Adam as the best optimizer, for 25% of models RMSProp is the best
- Epochs
  - Varied between 26 to 100 epochs
  - VGG and ResNet most likely are better at optimizing towards best loss performance more efficiently

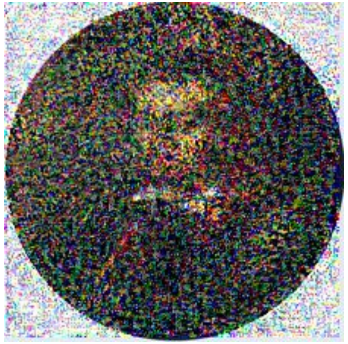
Model	Optimizer	Epochs
VGG Simple	RMSProp	36
VGG Squared	RMSProp	63
VGG Logged	Adam	100
ResNet Simple	Adam	100
ResNet Squared	Adam	26
ResNet Logged	Adam	62
Inception Simple	Adam	59
Inception Squared	RMSProp	100
Inception Logged	Adam	100
MobileNet Simple	Adam	100
MobileNet Squared	Adam	100
MobileNet Logged	Adam	100

Qualitative Hyperparameters

- Pretrained Models
  - ResNet did not change the images at all
  - Inception and MobileNet were very pixel-y
  - Only VGG will be used moving forward
- Loss Aggregation
  - VGG Logged was very grainy so will not be used
- Final Models to Analyze
  - VGG Simple and VGG Squared
- Examples: Image on Diego Rivera

ResNet (Like Original)

VGG Squared



Inception

VGG Logged

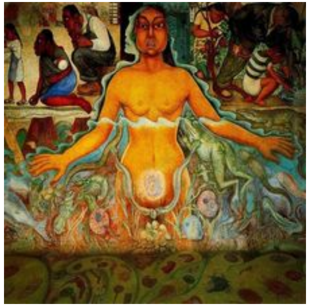
Squared vs Simple Model

- Simple and squared models deviate towards the art style of Diego Rivera
- Example shows that squared model gravitated towards a model that centered around a person like the content image
    - Versus simple model was more random
    - Not expected, thought to be opposite and seen with logged model
- Different line styles between pictures
    - Straighter lines for simple model

VGG Simple Model

Simple Lowest Loss

VGG Squared Model



Squared Lowest Loss

Conclusion

Conclusion

- Model hard to measure
  - Would be good to develop a more quantitative way to compare photos style and content between models
- Additional methods to iterate on
  - More epochs
  - Narrow - Time periods/eras of a specific artist
  - Broad - Overall art styles instead of a specific artist
- Use cases
  - Get general sense of style of the artist
  - Help create new images that are more of the artist than of a specific painting