Image Stylization Using Artificial Intelligence

Tezuesh Varshney
Department of Computer Engineering
Aligarh Muslim University
Aligarh, 202001
tezueshvarshney@zhcet.ac.in

Mansi Agrawal
Department of Computer Engineering
Aligarh Muslim University
Aligarh, 202001
mansi.29ag@gmail.com

Abstract

In the era where computers could perform most sophisticated task with ease teaching computer to emulate art is a complex task. Although creating art using AI techniques is relatively new, artists have been using algorithms, automation and computation to create art for decades. This project lies at the intersection of image processing to create artistic images. Aim of the project is to take an initial step into using Artificial Intelligence algorithms to create art. The aim of this project was two-fold for us 1) To understand the basics of various deep learning concepts and get an idea how to apply them to real life problems 2) To recreate images using basic geometric primitives. We have used the concepts of deep learning which includes ConvNet architecture, Transfer Learning and Gradient Descent for style transfer. Also we have used hill-climbing algorithm to recreate images using geometric shapes.

1 Introduction

1.1 Motivation

The motivation of this project was to explore the role of Artificial Intelligence to create images. The group members are both interested in learning about how machines would learn to draw. The project is a preliminary experiment on how computer can create artworks, and to develop a simplistic understanding of it.

The motivation of understanding Artificial Intelligence algorithms through art is to foray into this field. With a background in learning about the algorithms used in machine learning as well as the mathematics behind it, the group members were motivated to build something interesting using AI techniques. AI is a field full of nuances and is used to solve some of the hardest problems in the outside world, and we therefore could not attempt exhaustive research about the subject, but were motivated to get started and to convert some of the theory that we had been studying to something practical and visualize some interesting results.

Finally, the greatest motivation was to learn new and interesting things about how machines can create artistic images, as well as get experience with a medium-term project, and therefore also develop our software development skills.

1.2 Objective and Scope

Objective of the project is to create artistic images from the given input images. The project is divided into two folds:

1. Using neural style transfer to transfer style of one image to another
2. Using Hill Climbing algorithm to recreate image using basic geometric primitives.

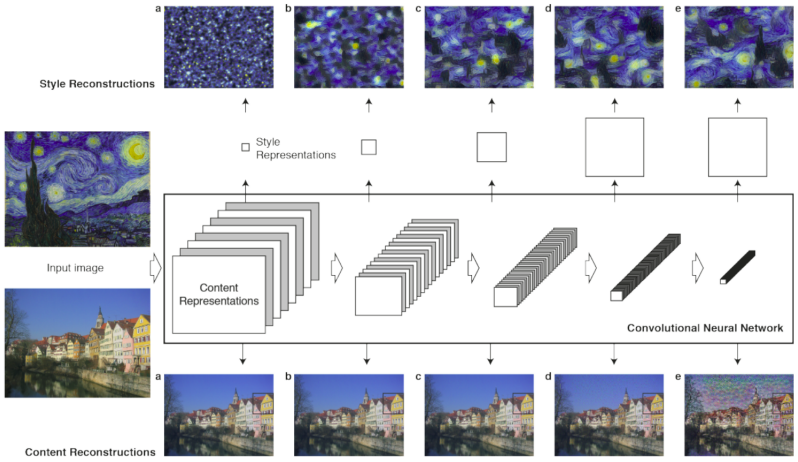

Figure 1: Style Transfer Description

The scope of the project is quite wide. These techniques are nowadays used in photo-editing apps such as Prisma in which the style of one image is added to other image. Users use such apps for clicking artistic images and use them as profile picture. There are other works on style transfer which are being applied on characters, doodles, faces etc.

There is a basic implementation of Convolutional Neural Networks (ConvNets) which acts as feature extractor for style and content image and is used for style transfer. Also, Hill Climb Algorithm is used to reproduce image using primitives.

2 Background

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

Once we have a pre-trained model we try to understand and visualize the functioning of Convolutional Neural Networks which provides an insight of how neural networks learn and complex properties.

3 Neural Style Transfer

Style transfer is the technique of recomposing images in the style of other images. Neural style transfer is an optimization technique used to take three images, a content image, a style reference image (such as an artwork by a famous painter), and the input image you want to style - and blend them together such that the input image is transformed to look like the content image, but "painted" in the style of the style image.

Style transfer, Figure 1, is a fun and interesting technique that showcases the capabilities and internal representations of neural networks. The principle of neural style transfer is to define two distance functions, one that describes how different the content of two images are, $L_{content}$, and one that describes the difference between the two images in terms of their style, L_{style}. Then, given three images, a desired style image, a desired content image, and the input image (initialized with the content image), we try to transform the input image to minimize the content distance with the content image and its style distance with the style image. Style transfer is divided into two process:

1. Selecting Model and Feature Selection
2. Running Machine learning Algorithm

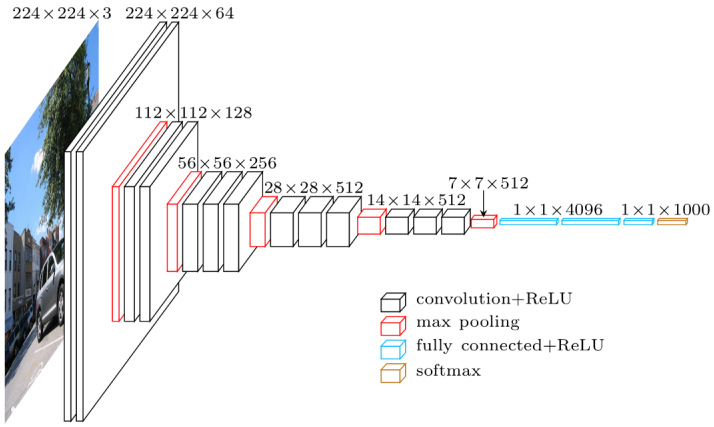

Figure 2: VGG 19

3.1 Selecting Model

We have selected VGG-19 model and thus select intermediate layers for feature extraction.

3.1.1 VGG-19 Model

VGG-19, Figure 2, is a 19 layer neural network created by Karen Simonyan and Andrew Zisserman (Very Deep Convolutional Networks for Large-Scale Image Recognition). The neural networks takes as an input an image of size 224 x 224 x 3 and outputs a classification vector that can classify the image as one of a thousand different objects.

So, if we feed an image A_i to this network then the output will be a classification of the image e.g. whether it is a house or a train and the activation at each of hidden layers are $L_1(A_i)$, $L_2(A_i)$. And $L_{19}(A_i)$. The activation at each layer represents a feature of the image that the network has learnt.

In order to get content image and style image representation we have used intermediate layers Intermediate layers represent feature maps that become increasingly higher ordered as you go deeper.

```
content_layers = ['block4_conv2']
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']
```

3.2 Loss

Generally each layer in the network defines a non-linear filter bank whose complexity in- creases with the position of the layer in the network. Hence a given input image $\vec{x}$ is encoded in each layer of the CNN by the filter responses to that image. A layer with N_l distinct filters has N_l feature maps each of size M_l , where M_l is the height times the width of the feature map. So the responses in a layer l can be stored in a matrix $F^l \in R^{N_l * M_l}$ where F^l_{ij} is the activation of the i^{th} filter at position j in layer l. To visualise the image information that is encoded at different layers of the hierarchy we perform gradient descent on a noise image to find another image that matches the feature responses of the original image.

3.2.1 Content Loss

let $\vec{p}$ and $\vec{x}$ be original image and image that is generated and P_l and F_l be their feature representation in the layer . Content Loss between two feature representations is given by:

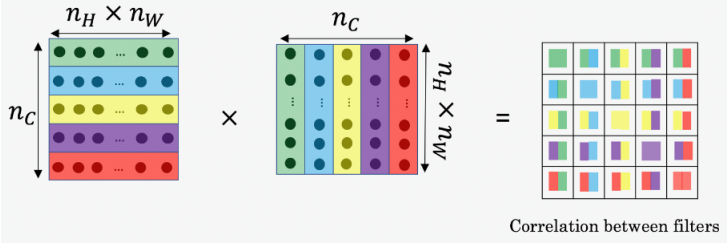

Figure 3: Representation of Gram Matrix

$$L_{content}(\vec{p}, \vec{x}, l) = \sum_{ij}(F_{ij}^l) - P_{ij})^2$$

The derivative of this loss with respect to the activations in layer equals

$$\frac{\partial L_{content}}{\partial (F_{ij}} = \begin{cases} (F^l - P^l)_{ij} & if F_{ij} \geq 0 \\ 0 & if F_{ij} \leq 0 \end{cases}$$

from which the gradient with respect to the image can be computed using standard error back-propagation.

3.2.2 Style Loss

Computing style loss is a bit more involved, but follows the same principle, this time feeding our network the base input image and the style image. We describe the style representation of an image as the correlation between different filter responses given by the Gram matrix G^l, Figure 3, where G_{ij}^l is the inner product between the vectorized feature map i and j in layer . We can see that G_{ij}^l generated over the feature map for a given image represents the correlation between feature maps i and j.

To generate a style for our base input image, we perform gradient descent from the content image to transform it into an image that matches the style representation of the original image. We do so by minimizing the mean squared distance between the feature correlation map of the style image and the input image. The contribution of each layer to the total style loss is described by

$$E_l = \frac{1}{4{M_l}^2{N_l}^2} \sum_{i,j}({G_{ij}}^l - {A_{ij}}^l)^2$$

Here G_{ij}^l and A_{ij}^l are the respective style representation in layer l of input image $\vec{x}$ and style image $\vec{a}$. Total loss across each layer is given by

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l \in L} w_l E_l$$

where we weight the contribution of each layer by some factor . By including the feature correlations of multiple layers, we obtain a stationary, multi-scale representation of the input image, which captures its texture information but not the global arrangement.The derivative of E_l in each layer l is calculated as

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2}((F^l)^T(G^l - A^l))_{ij}, & F_{ij} > 0 \\ 0, & F_{ij} < 0 \end{cases}$$

3.2.3 Total Variational Loss

Total Variational Denoising, is a process, most often used in digital image processing, that has applications in noise removal. It is based on the principle that signals with excessive and possibly spurious detail have high total variation, that is, the integral of the absolute gradient of the signal is high. According to this principle, reducing the total variation of the signal subject to it being a close match to the original signal, removes unwanted detail whilst preserving important details such as edges.

$$L_{variational}(x_{i,j}) = (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2)$$

3.2.4 Total Loss

The loss function we will try to minimize is i.e. Total loss.

$$L_{total} = \alpha * L_{content}(\vec{p}, \vec{x}) + \beta * L_{style}(\vec{a}, \vec{x}) + \gamma * L_{variational}(\vec{x})$$

Here:

$\vec{p}$ = Content Image
$\vec{a}$ = Style Image
$\vec{x}$ = White Noise
α = Weight of content loss in total loss
β = Weight of style loss in total loss
γ = Weight of variational loss in total loss

3.3 Algorithm

Aim of neural style transfer is to minimize the loss. Backward pass is implemented to modify the white noise image using Gradient descent Algorithm.

3.3.1 Gradient Descent Algorithm

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. We try to minimize total loss of the white noise image , content image and style image by tampering the white noise image on the basis of given gradients. $Image = Image - \alpha * Gradients$ Gradients for content image and style image are evaluated as explained in equations (1) and (2).

3.3.2 Adam Optimizer

Adam realizes the benefits of both AdaGrad and RMSProp. Instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam also makes use of the average of the second moments of the gradients (the uncentered variance). Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters $beta1$ and $beta2$ control the decay rates of these moving averages.

3.4 Result

Style Transfer: Dog and Starry Night

4 Creating Images using Primitives

4.1 Introduction

Hill Climbing is heuristic search used for mathematical optimization problems in the field of Artificial Intelligence. Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum. In the above definition, mathematical optimization problems implies that hill climbing solves the problems where we need to maximize or minimize a given real function by choosing values from the given inputs. Example- Travelling salesman problem where we need to minimize the distance traveled by salesman.'Heuristic search' means that this search algorithm may not find the optimal solution to the problem. However, it will give a good solution in reasonable time. A heuristic function is a function that will rank all the possible alternatives at any branching step in search algorithm based on the available information. It helps the algorithm to select the best route out of possible routes.

4.2 Algorithm Used

We used hill climbing technique to recreate images using basic primitives. The idea was to draw basic figures on the canvas of different sizes in order to get the final image similar to the input image.
Algorithm:

1. Initialise maximum and minimum size of the polygons.
   At each iteration, shapes are drawn on the canvas at random positions, picking colors from the input image. The shapes can have size ranging from minimum size to maximum size. The number of shapes at each iteration depends on the maximum size and width of the image.
2. The maximum size is reduced by 2 after 30 iterations.
3. Shapes are drawn on the canvas until maximum size becomes equal to minimum size.
4. The resultant image on the canvas is the output image.

4.3 Results

We used 3 types of primitives namely, ellipse, rectangle and triangles. We got the following output images.

Example 1: Scenery

Example 2: Dog

5 Conclusion

It is truly fascinating that a neural system, which is trained to perform one of the core computational tasks of biological vision, automatically learns image representations that allow the separation of image content from style. The explanation could be that when learning object recognition, the network has to become invariant to all image variation that preserves object identity.

6 Future Work

Potential future projects include different style transfers on a single image, such as transferring different style to main object and different style to background. One approach to accomplish this would involve separating the main object by object segmentation, and running our gradient descent algorithm on those two regions separately. The combination of the two would result in an image of two separate styles on separate regions. We would also like to explore real time style transfer.

Acknowledgments

We would like to express our deep and sincere gratitude to our project supervisor, **Prof. Sarosh Umar**, Ph.D., Professor and Chairman, Computer Engineering Department, Zakir Husain College of Engineering and Technology, AMU, Aligarh being an excellent mentor to us during the entire course of project. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the research and to present the project as clearly as possible. It was a great privilege and honor to work and study under his guidance.

References

[1]Leon A. Gatys & Alexander S. Ecker &Matthias Bethge *A Neural Algorithm of Artistic Style.*, arxiv.org, 26 Aug 2015

[2] Matthew D Zeiler& Rob Fergus, *Visualizing and Understanding Convolutional Networks*, arxiv.org, 12 Nov 2013 [3] Roman Novak & Yaroslav Nikulin, *Improving the Neural Algorithm of Artistic Style*, arxiv.org, 15 May 2016 [4] Diederik P. Kingma & Jimmy Ba, *Adam: A Method for Stochastic Optimization*, arxiv.org, 22 Dec 2014 [5] Chapter 9 - The Evolution of Code, from the Book , *The Nature of Code* [6] Karen Simonyan & Andrew Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition* , arxiv.org, 10 Apr 2015