NEURAL IMAGE TRANSFER

A COURSE PROJECT REPORT

By

TIRUMANI SURAJ (RA2011026010266)
JERAMIUS SHANON (RA2011026010267)
KOLAPATI MANI DEEPAK CHANDU (RA2011026010279)
AJAY ABHINAV B (RA2011026010280)

Under the guidance of

Dr.M.Vimaladevi

In partial fulfilment for the Course

of

18CSCE388T - ARTIFICIAL NEURAL NETWORKS

in CINTEL DEPARTMENT

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chenpalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "**NEURAL IMAGE TRANSFER**" is the bonafide work of **Tirumani Suraj(266),Jeramius Shanon(267),Kolapati Mani Deepak Chandu(279),Ajay Abhinav B(280)** who carried out the project work under my supervision.

SIGNATURE

Dr.M.Vimaladevi
ASSISTANT PROFESSOR
CINTEL
SRM Institute of Science and Technology

ABSTRACT

The main goal of this project is to explore Neural-style-transfer through implementation. We'll Implement a NST model using Tensorflow and keras, and at the end of the project we'll deploy it as a web app so that anyone can create stunning digital art which they could even sell as NFT's.

Pings were used to check the connectivity and the reachability of the systems from all the network. Style transfer is a computer vision technique that takes two images — a "content image" and "style image" — and blends them together so that the resulting output image retains the core elements of the content image, but appears to be “painted” in the style of the style reference image. Training a style transfer model requires two networks,which follow a encoder-decoder architecture

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We are highly thankful to our my Course project Faculty **Dr.M.Vimaladevi, Assistant Professor,Department of Cintel for** her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

ABSTRACT

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.

we’ll take the base input image, a content image that we want to match, and the style image that we want to match. We’ll transform the base input image by minimizing the content and style distances (losses) with backpropagation, creating an image that matches the content of the content image and the style of the style image.

INTRODUCTION

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, Lstyle. 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.

LITERATURE SURVEY

You may be wondering why these intermediate outputs within our pretrained image classification network allow us to define style and content representations. At a high level, this phenomenon can be explained by the fact that in order for a network to perform image classification (which our network has been trained to do), it must understand the image. This involves taking the raw image as input pixels and building an internal representation through transformations that turn the raw image pixels into a complex understanding of the features present within the image. This is also partly why convolutional neural networks are able to generalize well: they’re able to capture the invariances and defining features within classes (e.g., cats vs. dogs) that are agnostic to background noise and other nuisances. Thus, somewhere between where the raw image is fed in and the classification label is output, the model serves as a complex feature extractor; hence by accessing intermediate layers, we’re able to describe the content and style of input images.

REQUIREMENT ANALYSIS

In fine art, especially painting, humans have mastered the skill to create unique visual experiences through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. However, in other key areas of visual perception such as object and face recognition near-human performance was recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks. Here we introduce an artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality.

It is necessary to say what the different layers of a CNN represent in order to understand the subsequent calculations.The shallower layers of a CNN tend to detect lower-level features such as edges and simple textures.The deeper layers tend to detect higher-level features such as more complex textures as well as object classes.As the generated image shall have similar content as the input image. It is advisable to use a layer in the middle, representing content to a high degree.

ARCHITECTURE & DESIGN

The neural style transfer paper uses feature maps generated by intermediate layers of VGG-19 network to generate the output image. This architecture takes style and content images as input and stores the features extracted by convolution layers of VGG network.

IMPLEMENTATION

Cost calculation

First, why cost/loss calculation? It is important to understand that in this context the cost is the mere difference between the original and the generated image. There are multiple ways on how to calculate it (MSE, euclidean distance, etc). By minimizing the differences of the images we are able to transfer styles. When we start out with big differences in the loss, we will see that the style transfer is not that good. We can see that styles are transferred, but it seems rough and unintuitive. With each cost minimization step, we go in the direction of a better merger of the style and content and ultimately a better resulting image. As we can see the central element for this process is the loss calculation. There are 3 costs that need to be caluclated:

Content cost

Style cost

Total (variation) cost

Those steps are in my opinion the hardest to understand, so let's dive into it one by one. Always keep in mind that we are comparing the original input with the generated image. Those differences are the cost. And this cost we want to minimize. It is so important to understand this because in the process other differences will also be calculated.

Content cost

As we found out before, we define the content of an image by its objects. Things that we as humans can recognize as things.Having understood the structure of a CNN, it now becomes apparent that at the end of the neural network we can access a layer, that represents the objects (the content) quite well. Going through the pooling layers we lose the stylistic parts of the image, but in terms of getting the content, this is desired.Now the feature maps in higher layers of the CNN are activated in the presence of different objects. So if two images have the same content, they should have similar activations in the higher layers.That is the premise for defining the cost function.

Style cost

Now it is getting sophisticated.

Make sure to understand the difference between the style of an image and style loss of an image. Both calculations are different. One is to detect the "style representation" (texture, colors, etc), the other is to compare the style of the original image with the style of the generated image.

The total style cost is calculated in two steps:

the style cost of all convolutional layers. Identifying the style of the style image a. Getting the feature vectors from a convolutional layer b. Comparing those vectors with feature vectors from another layer (finding its correlation) .the style cost between the original (the original style image!) and the generated image.

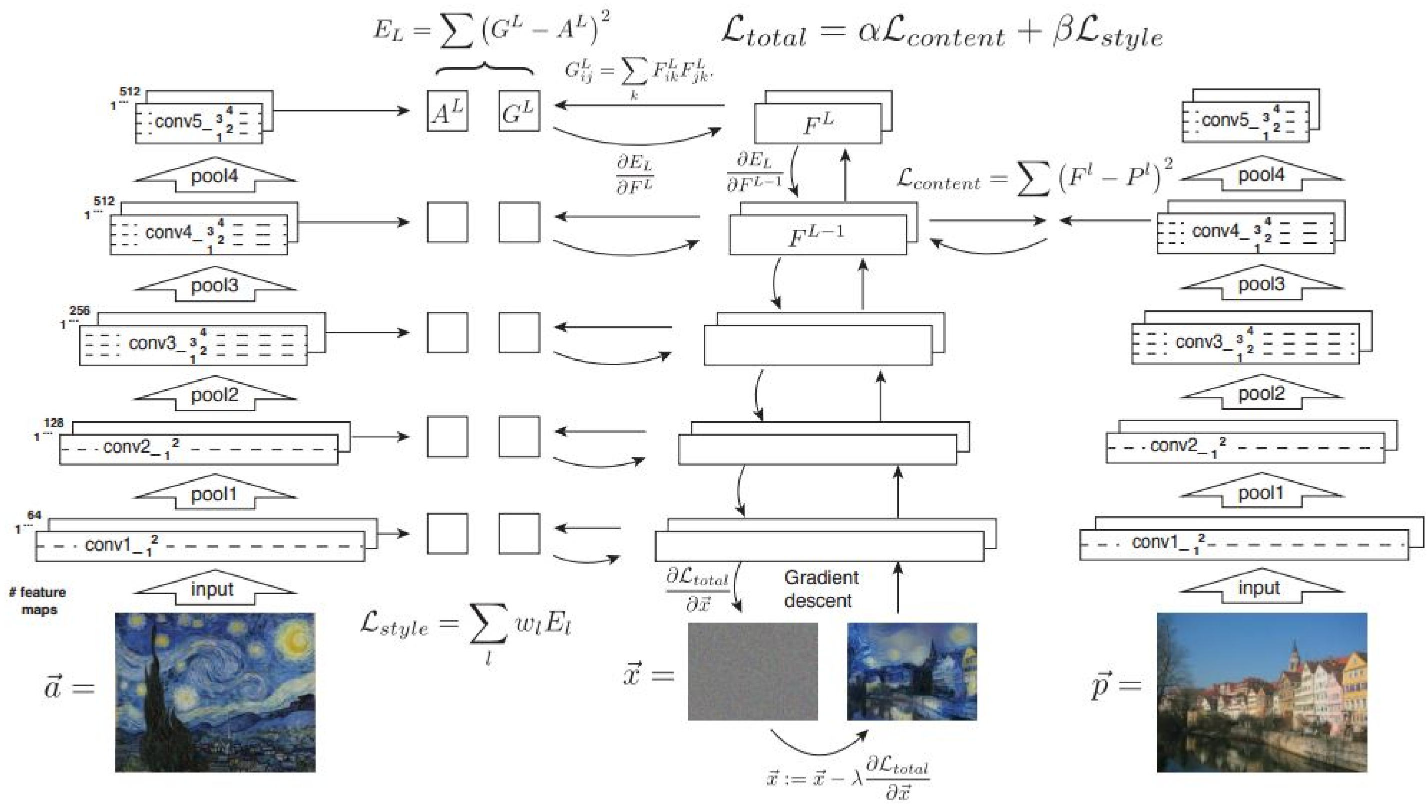
To find the style the correlation is captured by multiplying the feature map to its transpose, resulting in the gram matrix.

Total variation cost

It acts like a regularizer that improves smoothness in the generated image. This was not used in the original paper but sometimes improves the results. In essence we smooth out the differences between style and content transferal within the generated image.

EXPERIMENT RESULTS & ANALYSIS

RESULT:



RESULT ANALYSIS:

In this work, we have implemented Gatys' neural style transfer algorithm and explored the impact of hyperparameter setting on the output. We implemented a spatial control extension to Gatys' algorithm and showcased successful and unsuccessful examples of adding spatial control to neural style transfer. Gatys' method is capable of arbitrary style transfer, and produces high quality output, but has high computational cost (> 1 hour CPU runtime for 500x500 image, approximately 300 iterations). All 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 may fail spectacularly if the target style is too photorealistic, or if image resolution is too low

CONCLUSION & FUTURE ENHANCEMENT

Neural style transfer allows to blend two images (one containing content and one containing style) together to create new art. You first went through why you need neural style transfer and an overview of the architecture of the method. Then you defined the specifics of the neural style transfer network with TensorFlow. Specifically, you defined several functions to define the variables/inputs, compute the VGG output, compute the losses and perform the optimisation. You next understood the two losses that allow us to achieve what we want; the content loss and the style loss in detail, and saw how they come together to define the final loss. Finally you ran the model and saw artwork generated by the model.

Style transfer: beautiful results… but computationally expensive

When the original technique was created, there was a lot of excitement in the air: new papers and techniques were developed, and several apps were created to give the general public a taste of style transfer, like sites Deepart.io, Dreamscope.com, and app Pikazo. But there was a problem: style transfer, in its original conception technique, is very computationally expensive! Meaning, it takes huge GPU computations to actually produce moderately small images (and don't even think to run it on CPU, then it would literally take weeks!).

So, the technique clearly didn't scale to cater for free, massive apps, and so the afore-mentioned apps and sites collapsed under even small audiences. Hence there was a big incentive to invent faster techniques, which would work locally on the users' devices. And voila', a new technique, apropiately called fast neural style, appeared in 2016. Effectively it made the execution really fast… as in milliseconds fast. This enabled massive apps to actually cope with demand, and so apps like Prisma or Painnt (disclosure: developed by my company, Moonlighting Apps) appeared and became quite successful. To avoid naming clashes, the old, original technique went to be called optimization style transfer, as opposed to the new, feed-forward fast style transfer.

REFERENCES

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. CoRR, abs/1508.06576, 2015.

URL http://arxiv.org/abs/1508.06576.

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. URL

http://arxiv.org/abs/1611.07865.

Gu, S., Chen, C., Liao, J., and Yuan, L. Arbitrary style transfer with deep feature reshuffle. In The IEEE Conference
on Computer Vision and Pattern Recognition (CVPR), June 2018.

Jing, Y., Yang, Y., Feng, Z., Ye, J., and Song, M. Neural style transfer: A review. CoRR, abs/1705.04058, 2017. URL http://arxiv.org/abs/1705.04058.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016.

Li, X., Liu, S., Kautz, J., and Yang, M. Learning linear transformations for fast arbitrary style transfer. CoRR, abs/1808.04537, 2018. URL http://arxiv.org/abs/1808.04537.

Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M. Universal style transfer via feature transforms. CoRR, abs/1705.08086, 2017. URL http://arxiv.org/abs/1705.08086.