IJIRMET
ISSN (Online): 2456-0448
International Journal Of Innovative Research
In Management, Engineering And Technology
Available online

IMAXE

[1]Aiswarya Suresh, [2]Ann Mary Abraham, [3]AparnaRajendran, [4]Anand C,[5]Jeba Sonia J
Department of Computer Science and Engineering, AmalJyothi College of EngineeringKanjirappally
[1]meaiswaryasuresh@gmail.com,[2]annmary9901@gmail.com,[3]aparnarajendranar@gmail.com,[4]anandchandran92@gmail.com, [5]jsonia.sony@gmail.com

ABSTRACT

This paper brings a deep learning methodology to neural style transfer that deals with lots of image content and transferring it to reference style. Neural Style Transfer [1] is an interesting technique that represent the capabilities of neural networks. It is an optimization technique using a content image andmultiple style image. The input image is transformed to look like the content image but "painted" in the style of style images. Prisma is an example of neural style transfer application and it is a very popular Android app for styling. It takes input images from phone and process it in a server and sends it back to us. The main disadvantage of Prisma is, the user can use only one style image. In our work, an input of content image and the corresponding style images are given which are combined to form a final image that has the features and semantic details of the content image and the representation of style of style image(s). This allows for generating unique images by combining various style images and content images. The proposed system uses one content image and multiple style reference images for processing. To convert an image into the styles of an artist, neural artistic style transfer technique is used. For a given content image, we match the corresponding styles and content representations at intermediate layers of convolutional neural network which is used for image classification [4]. VGG16 model is used in the proposed system to perform style transfer at various layers in the network. It performs normalization and as requirement by VGG16 we convertsfrom RGB (red, green, blue) toBGR (red, green, blue). The system uses multiple style images which are transferred onto the specified content image and we get the corresponding output by using Keras functional API and Tensorflow as the backend. The comparison results show that this system overcomes the drawback of Prisma by using multiple style images. TheTensorflow on Android [12] as the frontend of the proposedsystem makes the style transfer easy and interactive.

KEYWORD: NeuraArtisticStyleTransfer,Convolution Neural-Network (CNN), VGG16

I. INTRODUCTION :

Artistic neural style transfer is anoptimization problem (long standing) that transfer the content image to look like the style images. Our proposal builds upon the recent work on style transfer using CNN byGatys [1[. Gatys used a pre-trained network i.e., pre-trained VGG-19 for image classification as feature extractor to drive texture synthesis and style transfer, yielding impressive performance benefiting from the rich feature representation. Later, Prisma Labs launched its application Prisma which draw ten million users in five weeks and Google Deep Style Ostagram and picsart also attracted a large number of users. People can see their transferred photo that done using various filters by uploading their photo. Thus, style transfer is becoming a new fashion and people show great enthusiasm for style transfer. These style transfers have limitations: one content image and one style image.

In this paper, we purpose a method for artistic neural style transfer that can use the style of the style images and content of the content image as much as possible. We're going to follow Gaty's idea, and use the feature spaces provided by one such model to independently work with content and style of images. We choose a 16 layer model (VGG16) [2]. Our proposed algorithm on style transfer can obtain a proper trade-off between conservingthe structureof content image and style of the reference image. The Tensorflow on Android is used to make our project more interactive and user-friendly.

II. RELATED WORK :

Style transfer by neural network is a field in deep learning. This technique has high popularity. Deepart.io and prisma are two such popular apps.
Inspired from CNN,Gatys et al. [1] first made the use of CNN to create style of painting on images. He proposed a model to extract content of a photo as feature responses from a CNN model and the artwork style as the summary feature statistics. The results from their results demonstrated that CNN is capable of extracting content information from an arbitrary photograph and style information from a well-known artwork. He used the CNN feature activations to combine the style of artworks and the content of given photo. The key idea behind his algorithm is to optimize an image using photo's content information and artwork style information. His proposed work produces images with the appearance of a given photo and style of the artwork.

Video Style Transfer: Video style transfer uses NST algorithms.It came afterGatys et al. [1] still images.NST algorithms. The video style transfer algorithm needs to consider the smooth transition between adjacent video frames. For this transitions, they divided the algorithms into two, video style transfer based on Image-Optimisationand Model-Optimisation.

Deep Photo Style Transfer [14]: It uses photographic style transfer using deep learning approach. It uses photos as content and style reference images. Here the output is the content of one photo and style of other photo that is the content photo appears in the style of style reference photo.

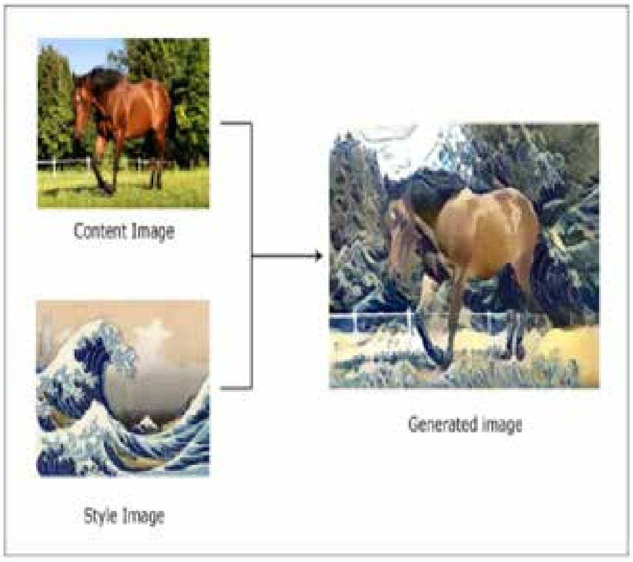


Figure 1: Style Transfer using a single reference image

III. METHOD :

In our work, CNN is used and it performs the feature extraction part. Every layer in CNN uses a matrix called filter matrix. This filter matrix is used over an array containing image pixels and perform a set of convolution operation to produce convolved feature map.

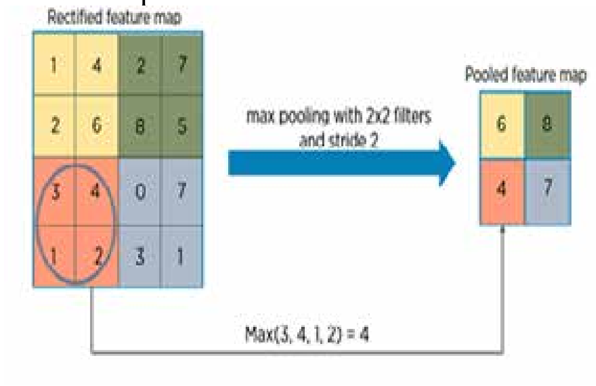


Figure 2: Feature Map

CONTENT REPRESENTATION AND LOSS

In our work, we choose every image to have the same content but not necessarily the same style and texture. We propose a convolved feature map for a given layer that matches with corresponding feature maps of chosen content image.
Content loss is defined as the Mean Squared Error between the feature map F of our content image C and the feature map P of our generated image Y, for a chosen content layer L.

$$L_{content} = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2 \quad (1)$$

Content loss minimization means the mixed-image has feature activation in the chosen layers that are almost similar to the activation of the content image. For the given layers, we select the things that will transfer the localfeatures from the content image to the combination image.

STYLE REPRESENTATION AND LOSS

In proposed system, we need to measure the style layers features and require to activate simultaneously the style image features, and to the concatenated image duplicate this activation pattern. The Gram-matrix (a matrix of dot-products for the vectors of the feature activations of a style-layer) of intermediate output layers is compared, instead of choosing raw outputs from the convolutional network.

If a value in our Gram-matrix is approximately equal to zero means two features do not activate same time in a layer. If values are large in a Gram-matrix which means the two of the features do activate at same time for the given reference image. We then use this pattern of activation of the style-images to create a concatenation of image. For creating combination image we use style images activation pattern.
For every entry in Gram matrix Gcan be given by the equation, if the feature map is a matrix F, then:

$$G_{ij} = \sum_K F_{ik} F_{jk} \quad (2)$$

The loss function for style can be calculated by the Mean Squared Error .It is calculated for the Gram-matrices.

$$L_{style} = \frac{1}{2}\sum_{l=0}^{L}(G^l_{ij} - A^l_{ij})^2 \quad (3)$$

The combination of shallow and deep layers transformation using the above equations gives the best form of style representation.

IV. IMPLEMENTATION :

The process of style transferring of image on laptop consumes some time and difficult without an interactive interface. We want to download the image we want to classify and enter a lot of code in the terminal just to style transfer the image. In our work anyone can transfer image live on a camera-equipped phone. We simply need to point at what we are trying to transfer and Tensorflow will do what we want to perform. We can also use Tensorflow on iOS and Raspberry Pi but for our work we use it on Android.

Here we transfer our input image into feel and look of style images. We perform this on an Android phone. We have two repositories in our work, android SDK and NDK (SDK is standard development kit).SDK is built in Java and NDK in C++.We use SDK as an interface to code everything we want in Java high level, that of the thing native to android (activities and fragments and so on).Once after downloading dependencies, train the model on desktop laptop in Python.

The first step in our work is to load the content and style images. We then convert it into something suitable for numerical processing. Then add another dimension to combine these representations of imageslater into a common structure of data. Next we need to massage this datataken as input to match with what was performed by ZissermananandSimonyan (2015) [12], the paper that introduces the VGG model which is the model that we are using in our work. For this, the next step that we need to do is two

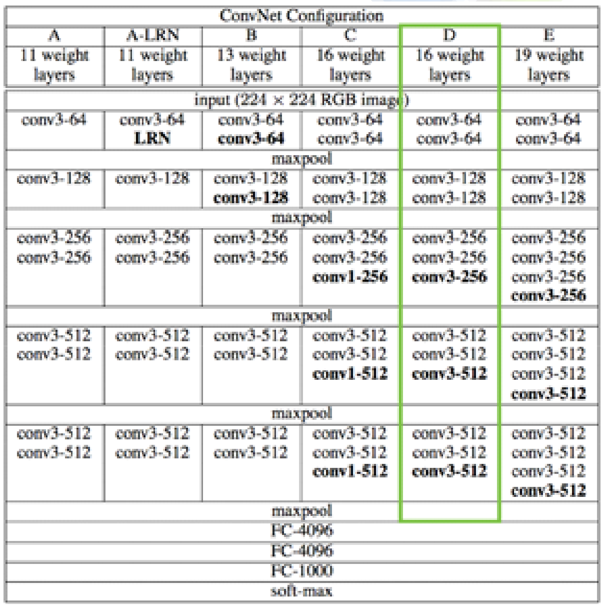
TRANSFORMATIONS

The mean RGB value (computed previously on a training set called ImageNet and easily obtainable from Google searches) should be subtracted from each pixel

we flip the ordering of the multi-dimensional array from RGB to BGR

We can use these arrays to define variables in Keras backend (the TensorFlow graph). There is a variable called placeholder variable is introduced to store the concatenated image. It retains the structure of our content image.

Then combine all this image data (content image, style images and concatenated images) into a single combination image that is suitable for VGG16 processing. The CNN is pre-trained for image classification already know how to encode perceptual and semantic information about images. This is the core idea introduced by Gatys [1].



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: Convolutional Network Layer

We use the feature spaces provided by one such model followed in their idea to independently work with content and style of images. We are going to use the 16 layer model (VGG16) by following Johnson et al. (2016). We gain a tiny bit in speed by making this choice. There is no requirement of fully connected layers (or the final softmaxclassifier) since there is no interest in the classification problem. The part of the model marked in green in figure 3 is needed in our work.

Keras includes a set of models which are pre-trained (includes the VGG16 model) that we are using in our work. We don't require any of the fully connected layers. The model we're working with has a lot of layers and Keras has its own names for these layers.

The style transfer problem can be posed as an optimization problem, where the loss function that we want to minimize can be decomposedinto three distinct parts (total variation loss, contentloss, and style loss).A set of scalar weights determines relative importance of these losses.

CONTENT LOSS

We follow Johnson paper [2] for this loss and draw the features of content from block2_conv2 because the original choice in Gatys et al. (2015) is block4_conv2 loses too much structural detail of the content.
The Euclidean distance (scaled, squared) between representation of feature of the concatenated images and the content image is the content loss.

STYLE LOSS

For this loss, we first define a Gram matrix. The terms of the Gram matrix G are proportionate to the co-variance of corresponding sets of features, and thus captures information about which features need to be activated. We then capture these aggregate statistics in the image, which are independent to the arrangements of objects in the image. This is what allows them to capture information about style independent of content.

By reshaping the feature spaces suitably and taking an outer product and then calculate the Gram matrix.This loss is theFrobenius norm (scaled, squared) of difference of the style images Gram matrices and concatenated images.

TOTAL VARIATION LOSS

The output is quite noisy if we were to solve the optimization problem with only the two loss terms, style and content. To encourage spatial smoothness we add another term, called the total variation loss.

DEFINE NEEDED GRADIENTS AND SOLVE THE PROBLEM OF OPTIMIZATION

The next step is to define gradients of the total loss relative to the combined image, and use iteratively these gradients to improve upon our combination image to minimize the loss.
Solving our problem of optimization is the final step. The L-BFGS algorithm is used to repeatedly toimprove upon this concatenated image .Its begins its life as a collection of random pixels (valid).

We stop after a number of iterations because the output looks what it required to us and the loss has optimized significantly. After creation of an app on Android and then made it to use Tensorflow libraries, we invoke the CNN model inside this Android app.

V. CONCLUSION :

We proposed a novel framework for artistic neural style transfer in this paper. The insight of the framework is that we create a new style representation of the input photo by applying multiple artistic styles. In our paper, we have applied optimization-based techniques for image generation. The image is generated with training the CNN with loss functions. We can input multiple styles to the content image and obtained output image with effects. The proposed model can be used in any Android phone connected to a server. The Tensorflow on Android [11] made it more user friendly and interactive.

Imaxe is neural style transfer app that's going to create novel painting in style of an artist or in traditional painting style like mural paintings with deep learning techniques. The goal is to give more visibility to local art and traditions from non-western cultures as general neural style transfer apps tend to focus on the classic western paintings.

VI. REFERENCE :

1. L.A. Gatys, A.S. Ecker, and. Bethge. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2414–2423, 2016. 1, 2, 3, 5
2. Perceptual Losses for Real-Time Style Transfer and Super-Resolution: Justin Johnson, AlexandreAlahi, Li Fei-Fei, {jcjohns, alahi, feifeili}@cs.stanford.edu, Department of Computer Science, Stanford University
3. J. Johnson. Neural-style. https://github.com/jcjohnson/neural-style, 2015.
4. DeepikaJaswal, Sowmya V, K.P.Soman: Image Classification Using Convolutional Neural Networks. International Journal of Advancements in Research & Technology, Volume3, June-2014 1661 ISSN 2278-7763
5. Keiron O'Shea and Ryan Nash: An Introduction to Convolutional Neural Networks. arXiv:1511.08458v2 [cs.NE] 2 Dec 2015
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems.pp. 1097–1105 (2012)
7. Zeiler, M.D., Fergus, R: Visualizing and understanding convolutional networks. In:Computer Vision–ECCV 2014, pp. 818–833. Springer (2014)
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition.doi:10.1109/cvpr.2009.5206848
9. M. Elgharib, and L. Doyle. Painting style transfer for head portraits using convolutional neural networks. ACM Transactions on Graphics (TOG), 35(4):129, 2016.
10. E. W. Weisstein. Gram matrix. MathWorld-A

Wolfram Web Resource. http://mathworld.wolfram.com/GramMatrix.html.

11. Creating an image classifier on Android using TensorFlow:https://medium.com/@daj/creating-an-image-classifier-on-android-using-tensorflow-part-1-513d9c10fa6a
12. Dr. S. Rabiyathul Basariya, and Dr. Ramyar Rzgar Ahmed, 2019. "The Influence of 'Adventure Tourism Activities' in promoting tourism business in mountain stations", African Journal of Hospitality, Tourism and Leisure, Volume 8 (2).
13. Dr. S. Rabiyathul Basariya, and Dr. Ramyar Rzgar Ahmed, Nov 2018. "A Study On consumer satisfaction and preference of colour TV brands in Chennai city", International Research Journal of Management and Commerce, Volume4, Issue 10.
14. Dr. S. Rabiyathul Basariya, and Dr. Ramyar Rzgar Ahmed, "A Study on Attrition: Turnover intentions of employees", Jan 2019. International Journal of Civil Engineering and Technology (IJCIET), Volume 10, Issue 9.
15. Dr. S. Rabiyathul Basariya, and Dr. Nabaz Nawzad Abdullah, Dec 2018. "A STUDY ON CUSTOMER'S SATISFACTION TOWARDS E-BANKING", International Research Journal of Management and Commerce, Volume 5, Issue 12,
16. S. Vasanthi and Dr. S. Rabiyathul Basariya, "EMPLOYEE CROSS TRAINING AND ITS IMPACT ON EMPLOYEE PERFORMANCE". International Journal of Civil Engineering and Technology (IJCIET),Volume 9, Issue 6, June 2018, pp. 800–806.
17. Very Deep Convolutional Networks For Large-Scale Image Recognition Karen Simonyan & Andrew Zisserman + Visual Geometry Group, Department of Engineering Science, University of Oxford {karen,az}@robots.ox.ac.uk
18. Vgg16 implementation: https://engmrk.com/vgg16-implementation-using-keras/
19. Deep learning: https://www.ibm.com/in-en/cloud/deep-learning
20. DeepPhotoStyleTransfer: FujunLuan,Sylvain ,Eli Shechtman ,KavitaBala