Use of Convolution Neural Network for style transfer

Aishwarya Panicker[1], Aarushi Gautam[2] , Oyshee Das[3] and Ajay Kumar[4]

Department of Computer Science,SRM Institute of Science and Technology,Vadapalani campus,Chennai

Abstract— Recognition of images in the styles of other images is a tedious task. Usually the image resolution is a major contributor to it.The omnipresent CNN makes it a bit easier to preserve the images content and resolution. CNN hierarchy from VGG net represent the images feature on a much larger scale and thereby providing a high level representation of the image content. Although there is a shorter window for escaping from the shortcomings such as trade off between the content and style matching and minimization of the loss function but gradient descent could be a way forward.

I. INTRODUCTION

The vision involves presenting an algorithm that creates artwork from several mediocre pictures by presenting the capabilities and internal representations of neural network.The technique leverages the adoption of Convolution Neural Network for creation of artistic looking images from base image and style images. Base image which generally constitutes a photograph is in a way visually superimposed on the style image which may involve a scenery ,painting or other forms of filter. CNN - a deep neural network algorithm, usually comes into use for imagery image processing ,classifica-tion and recognition of images. CNNs are extensively used in image and video recognition,natural language processing,medical image analysis,recommender systems etc.

A. Texture Synthesis and Transfer

Computer aided design has seen a flurry of activity in the past decade as researchers have delved deep into the idea of image based rendering using real world image as samples and recreating them. This in turn gave rise to various texture synthesis and transfer algorithms and paved way for the generation of image by mixing up style and content of different images. Texture synthesis makes use of the structural content of the sample image for the construction of digital images with same texture while texture transfer relies on transferring the style from one image onto another.

Previous establishments have relied on image quilting for texture synthesis and transfer. Image quilting is the process by which new image is produced by rendering together small patches of existing images and also used for texture transfer: applying the texture of an image to another. [1]Ashikhmin has done pioneering work in his paper on Fast Texture Transfer and has put forth an algorithm for texture transfer between images that is much faster than the current art tech-niques. [2]Exploring the idea further and incorporating the concept of convolution neural network for the processing of image which successfully captures the spatial and temporal arrangements of an image without flattening them.[3]

B. Convolution Neural Network

The purpose of CNN is extraction of high level features from the base and style images. The initial Convolution Layers are used for capturing the low level features such as edges, colour, gradient orientation etc.As the CNN is trained it learns to extract more complex features discarding irrelevant information and here's where max-pooling layers come into view. Pooling layers are used to reduce the dimensionality and computational cost by reducing the number of parameters. Max pooling has been taken into account rather than average pooling as it provides better results. It follows the strategy of down sampling an input image representation wherein for each of the areas that are represented by the filters,we create the feature map by taking maximum of each of those areas,thus forming an output matrix in which each element is the maximum of an area in the input matrix.

The emergence of Convolution Neural Networks for style transfer comes with genesis of processes, namely - Convolution,Pooling and fully connected layers. Feature map extracted from the images are processed to perform image recognisation and localization. For this, feature space of VGG 19[4] comes into use. VGG 19, that has 16 convolution layers, 5 pooling layer and 3 fully connected layers have been used . The images are normalized such a way that it is invariant from the geometrical distortions[5]. Normalisation is the technique which is applied for attributes with varying scales and this is required because neural networks rely on gradient calculation due to which the values may get exceedingly large which will eventually slow down the processing speed.To make the output layer more interpretable softmax function is used which will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.The formula computes the exponential (e-power) of the given input value and the sum of exponential values of all the values in the inputs. Then the ratio of the exponential of the input value and the sum of exponential values is the output of the softmax function and its main advantage is that output probabilities will be in the rang of 0-1.

C. Edge Detection

It is determined by the changes in density of an image from pixel to pixel. High frequency images are those in

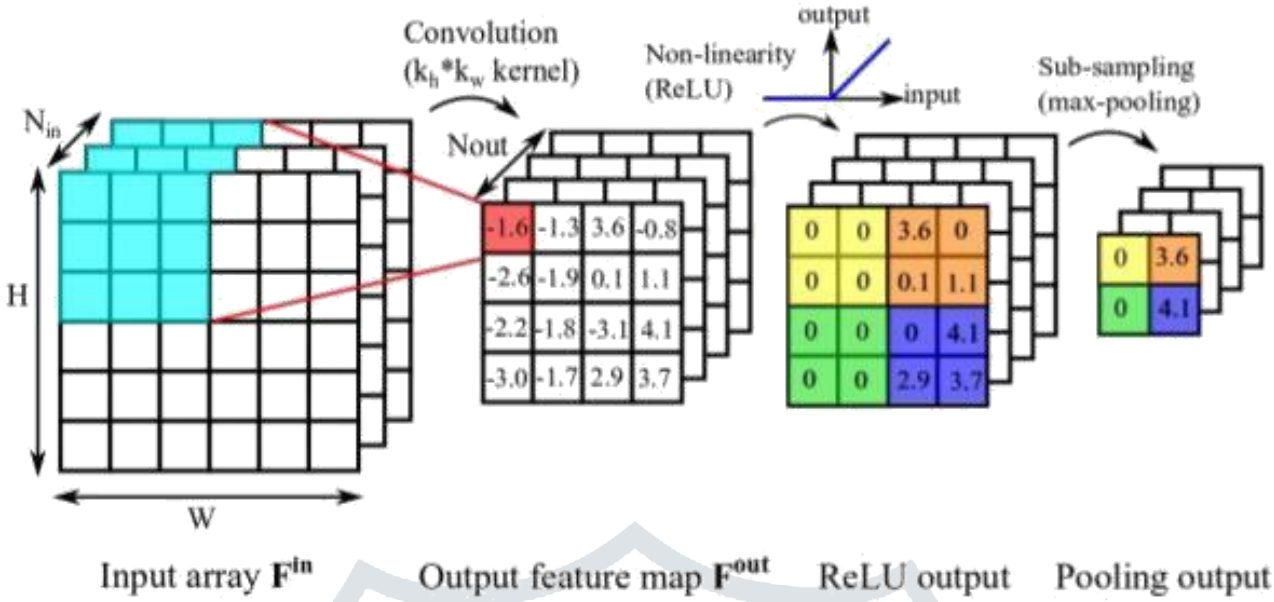

Fig. 1: The figure represents the Convolution Neural Network.Input array being the matrix of pixels. Output feature map is taken by extracting a single feature from the content image and trying to impose it over the styled image. The value of feature from content image is multiplied with every unit of the styled image and then it is then added to figure out the similarity. The output feature map is then proceeded for ReLU layer. ReLU or Rectified Linear Unit layer removes the negative value from the matrix by making it zero,rest all positive values are returned as same. Further,pooling takes place.

which intensity and level of brightness changes drastically. Low frequency images have uniform brightness.Thus high frequency components correspond to edges of objects in images which help to classify these objects and thus helps in detecting the edges[6].

Convolution kernels: Filters are in the form of mattresses called convolution kernels which are used for the filtering of irrelevant information and amplifying the necessary features,thus determining the edge boundaries. For application of this filter image f(a, b) is convolved with the kernel.

$$K \quad F(a; b) = \text{OutputImage}$$

II. MERGING

After removing the negative values in ReLU, layers get stacked. This is known as Deep Stacking where output of one layer becomes the input of one layer. Layers can be repeated several(or many) times in a deep stacking. Every time the layer gets iterated, it brings smaller feature map with itself. Fully connected layer brings a vote for every value. Vote depends on how strongly a value predicts X or O.Higher the value, stronger the weight.After calculating the average of close proximities to X or O ,average is calculated. Comparison of the values of average of X and O disclose the winner. These values can also be stacked[9]. All these stacking completely depends on the architecture being used.Here, Backpropogation plays a big role in the exploration of all the magic numbers, as in ,features in

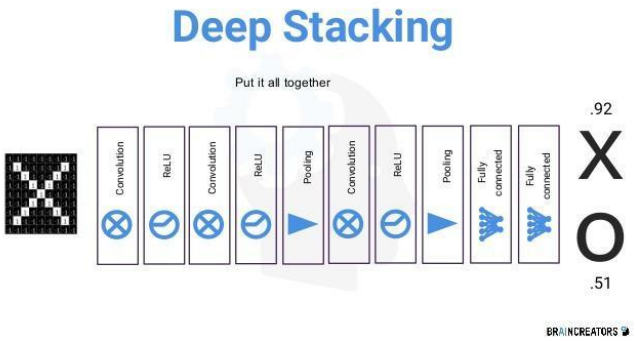

Fig. 2: The figure represents deep stacking where different architectural layers of convolution neural network are merged iteratively.

convolution layers and the voting weights in fully connected layers.

Gradient Descent lets one adjust the feature pixel and voting weight up and down to minimize and mitigate the error function. The iterative approach lets the convex loss function converge with the minimum i.e. zero slope. Hyperparameters (or knobs) used for convolution are: number of features and size of features, for pooling: window size and window stride and number of neurons for fully connected layers.

Flattening: The pooled feature map obtained after pooling is a 2D matrix that can not be operated on easily while using them for fully connected layer. Thus it has to be converted into an array of values from where votes value can easily be collected. After flattening, we end up having a vector of data

	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49
		Total	0.57

Fig. 3: The table here represents the error table.This error table is the prominent factor responsible for the efficient output as lesser the error rate higher the efficiency of the model. Also, it can be used to reveal whether the feature is closer to X or O. Here, X or O can be a formation in the content image.

that is passed through neural networks for further processing.

Losses: Losses are minimized by the use of backpropagation. The data that is sent for backpropagation is the stylized image we wished for. Our style image and content image are passed through the neural networks model which has been pretrained on thousands of images. The outputs of various layers of network are used to compute the content and style loss. For minimizing the loss, our output image is changed rigorously several times. After a few iterations, the output image contains the style of one image and content of another image.

Content loss: To compute the content loss, output of previous layer or previous iterations are taken and output of the current layer is taken. Then, their difference is squared. Here, we calculate the content loss in our previous output and the ongoing network layer output.

$$Loss_{content} = \sum_{i;j} (O_{i;j} - P_{i;j})^2$$

The formula here, represents the content loss computation. Summation here represents summing the result overall over i and j.

Gram matrix: Gram matrix that has non localized info about the image, is responsible for distributing the features. Distribution of features is done by multiplying a matrix with its transpose. Further, the euclidean distance between the two gram matrices is calculated for accuracy and efficiency.

$$G_{a;c} = \sum_{c} R_{a;c} R_{b;c}^{T}$$

The formula represents the gram matrix computation. G here represents the output for gram matrix computation. R represents the matrix and latter being its transpose.

Error = right actual

III. STYLE AND CONTENT LOSS

Content loss:Loss that is calculated as the difference between the Content Representation from the content image C_c and Content Representation from target image T_c.A mean square difference between the two representations is calculated.

$$L_{content} = \frac{1}{2} \sum_{b} X_a (T_c - C_c)^2$$

$L_{content}$ indicates how much variation these images have with respect to one another. Optimization for reducing the content loss is performed using backpropogation and by changing the target image such that it matches with the content image.

Style Loss:For calculating the style loss,mean square dis-tance between the gram matrix of target and style image is found in pairs and stored in lists.

$$L_{style} = a \sum_{i} w_i (T_{s;i} - S_{s;i})^2$$

These lists are called S_s and T_s and 'a' is an arbitrary constant that accounts for the value in each layer. These mean squared distances are multiplied with style weights and are summed up.

The optimization is performed by varying the style repre-sentation of target image in various iterations[7].Total Loss is calculated by summing up the content loss and the style loss which tells us how close the target image is to that of style and content image.

IV. HYPER PARAMETERS

Hyperparameters (or knobs) used for convolution are: number of features and size of features, for pooling: window size and window stride and zero padding. Number of features and size of the features play a major role in the volume of the output.Greater the number of features, higher the clarity but in bigger models like that of VGG 19, consumes features as many as 100,000,000. Anyhow, the aim later should be to reduce the number of features but reduction in features gives a setback to our clarity.This becomes a challenge but to overcome this Parameter Sharing comes into play.Parameter Sharing can be defined as the sharing of parameters by different layers. Number of features that comes into use for one particular location could be of use to some other cooardinate too[8]. Therefore same parameters are used and hence reducing our number of features by not disturbing the model's clarity.

Secondly, window size and stride size should be specified. Suppose our stride is 2 units. Then single computation takes place for 2 units and then our stride moves to next 2 units.Thirdly,zero padding is hyperparameter which is very essential to the model.It's credibility comes into use when we usually are left with units less than our stride size. So, the window is then padded with zeroes to avoid this dead end.

The output of the volume can be calculated through the following formula.

$$(V - C + 2Z)/S + 1$$

Here, V stands for the input volume, C stands for Convolution layer neurons' field size,Z stands for amount of zero padding and S stands for stride size. Suppose our input size,

Fig. 4: The figure depicts 3 style transfer outputs i)The style of an artwork ”Starry Night of Vincent van Gogh” is transferred to the picture of a dog ii)Style transfer of Pablo Picassos cubist style on the Mona Lis iii)Content image: The Tubingen Neckarfront by Andreas Praefcke, Style painting: Head of a Clown, by Georges Rouault

V is 4, field size for Convolution layer neurons’ C is 2, P is 1 and S is 1.So the output comes out to be 5.

V. Result

Through this paper we try to establish the use of Convolution Neural Network in styling the image,feature extraction, geometric normalisation and minimization of loss of content. The omnipresent CNN are the key to minimization of loss function which plays a bigger role today when we need high resolution images. The image shown in figure 4 left side was used for extracting the features and the image in figure 4 middle was used to extract the style.Later the extracted feature from content image and style extracted from the style image are combined using matrices and then the output from one layer is then filtered as per the architecture and moved for the next processing. The image in figure 4 right side represents the output for the model proposed.

VI. Future Scope

With the advancement of synthesizing novel view using batch processing techniques the computaional cost for performing style transfer has reduced significantly. In the current era, catering to the needs of general audiences that need faster techniques which would work locally on the user’s devices, style transfer fails to revolutionize the computer graphics and art world. But with few more optimisations like finding an alternative for GPU computations and exploiting faster and more flexible techniques for training,its future can be made as bright as originally envisioned.

References

[1] Alexei A. Efros,1University of California, Berkeley and William T. Freeman,Mitsubishi Electric Research Laboratories on A. Efros,University of California, Berkeley and W. Freeman,Mitsubishi Electric Research Laboratories on Image Quilting for Texture Synthe-sis and Transfer, Proc. Siggraph, ACM Press, 2001, pp. 341-346.

[2] N. Ashikhmin, ”Fast texture transfer,” in IEEE Computer Graphics and Applications, vol. 23, no. 4, pp. 38-43, July-Aug. 2003.

[3] Karen Simonyan and Andrew Zisserman,Visual Geometry Group, Department of Engineering Science, University of Oxford on ”Very Deep Convolutional Netwoks for Large Scale Image Recognition” published as a conference paper at ICLR 2015

[4] A. Aimar et al., ”NullHop: A Flexible Convolutional Neural Network Accelerator Based on Sparse Representations of Feature Maps,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 3, pp. 644-656, March 2019.

[5] M. S. Yasein and P. Agathoklis, ”An Image Normalization Technique based on Geometric Properties of Image Feature Points,” 2007 IEEE International Symposium on Signal Processing and Information Tech-nology, Giza, 2007, pp. 116-121.

[6] J. Sklansky, ”Image Segmentation and Feature Extraction,” in IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 4, pp. 237-247, April 1978.

[7] C. Xu et al., ”Multi-loss Regularized Deep Neural Network,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 12, pp. 2273-2283, Dec. 2016. doi: 10.1109/TCSVT.2015.2477937

[8] M. U. Yaseen, A. Anjum, O. Rana and N. Antonopoulos, ”Deep Learn-ing Hyper-Parameter Optimization for Video Analytics in Clouds,” in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 253-264, Jan. 2019.

[9] A. Ruiz-Garcia, M. Elshaw, A. Altahhan and V. Palade, ”Stacked deep convolutional auto-encoders for emotion recognition from facial expressions,” 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 1586-1593.