International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

PAPER PRESENTATION ON APPLICATION OF CARTOONIFYING OF AN IMAGE

K.SANDHYA , K.VAMSHIDHAR REDDY , P.MANIDEEP REDDY
Mr.Meeravali shaik (Guide)
Sreenidhi Institute of Science and Technology,Hyderabad

ABSTRACT:

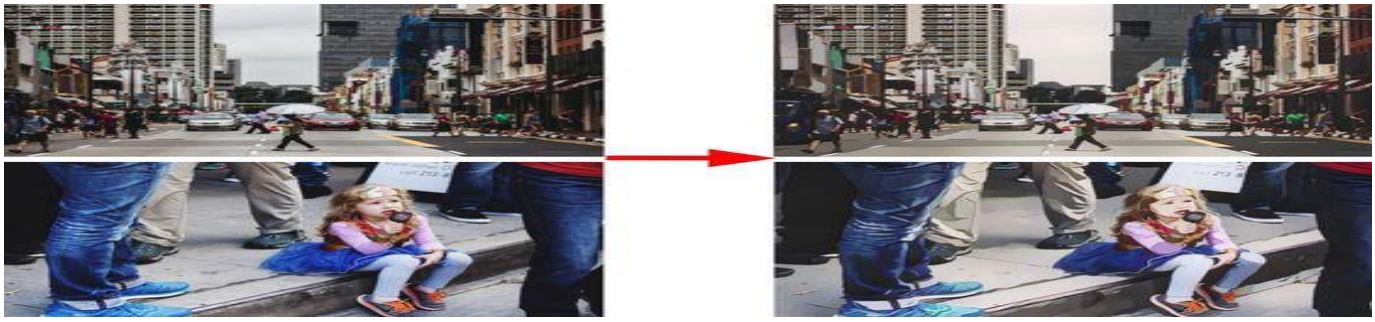
There are many techniques involved in converting an Image into Cartoon like Adobe,Photoshop,Windows MAC. We can convert any type of images into cartoon by the above methods but we have to choose an efficient approach. This paper is about conversion of an image into cartoon by using a different technique called Neural style transfer Algorithm.
This helps a lot for conversion in an efficient manner.

INTRODUCTION

We are using techology extensively these days and these technology makes us to communicate and connect with the people easily all over the world. There are so many social media platforms like Insta,Youtube, Twitter where we can post our photos,videos,blogs,tweets etc. .and we always seek attention of others to create influence and connect with them. So we will make use of posting a nice profile or a post.we provide a solution for this by cartoonifying the image which adds elegant look for our profile.By using of many libraries which is there on the online we can make the conversion of image to cartoon.

NEURAL STYLE TRANSFER

On the internet, we might have seen a lot of artworks created by neural networks so that method of transeferring a style from picture to another or in real time videos is known as neural style transfer because neural networks or their properties to transfer.The style of one image to another image that gains the style is called the content image and the image whose style is transferred is called style image The previous existing systems, transitions, and gradients makes a lot more smoother and better.And know we also have really fast neural style transfer which can transfer style in real time.



The class of deep neural networks that are more powerful in image processing tasks are called convolutional neural networks

Convolutional neural networks consists of layers of small computational units that process visual inforamation hierachically in a feed forward manner

Each layer of units can be understood as a collection of image filter,each of which extracts a certain feature from the input image thus,the output of given layer consists of so-called feature maps differently filtered versions of the input image

Along the processing hierarchy of the network, the input image is transformed into representation that increasingly care about the actual content if the image compared to its detailed pixel values.

We therefore refer to the feature response in higher layers of the network as the content representation.

The 4_2 convolution layer of pre-trained VGG-19 network is used as content extractor.

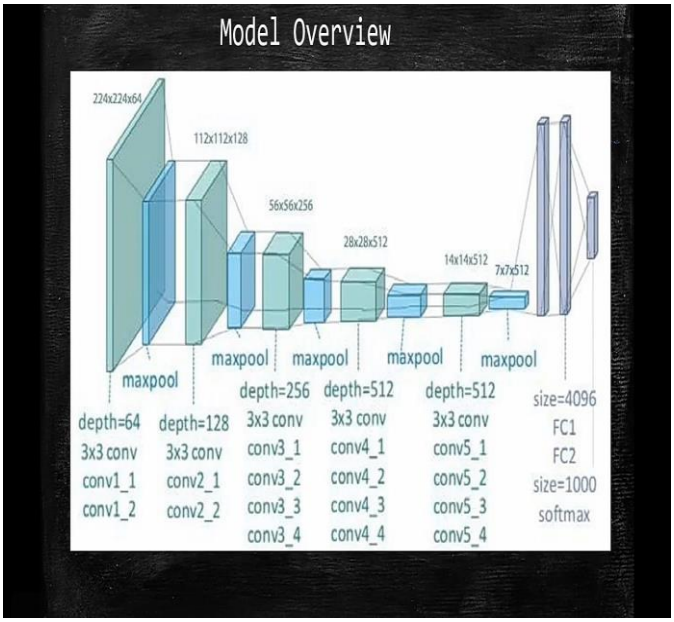
In order to style of the input image, we use correlations between different filter responses it ignores the actual arrangement of the object in the image. But it actually captures only the texture and colours.

Hence in order to extract the style we use correlations.

These correlations between feature maps have been given as a special name known as **GRAM MATRIX.**

The layer useful for calculating the gram matrix are the first part of the first layer. Same goes for the second, third, fourth, fifth layers with varied style constant for each layer.

So, if we want to use more of the first layer, we multiply with the constant that is, higher in value compared to the others.



For style, we are using lower-level layers as well as styles captured from lower layers and content is captured only from the higher-level layers.

In standard Implementation, second part of fourth layer is used.

The layers used for calculation of GRAM MATRIX are 1_1,2_1,3_1,4_1,5_1 with valid style weight constant for each layer.

The constant that we multiply to each of these GRAM MATRIX can be seen as hyperparameter used for changing style levels. So, if we increase that style from particular layer is captured more and not from the other layers.

Whenever a constant depth and we perform convolutional with constant depth so that the next convolutional won't be given a name of include like for example if depth is 64 and we perform another convolution and the depth remains the same so it will become the part of first convolution that's why we will refer one two convolutions here.

If we change the depth convolution changes.

It is a standard notation, hard and fast tool so once change the depth, we can come to next level of convolution so it is referred to as gone to know that is the first part of convolution to know.

Now if we change the depth, it becomes three and keep the depth same and keep on performing a lot more convolutions, the depth remains same .so the corner will be same but it's part will increase.

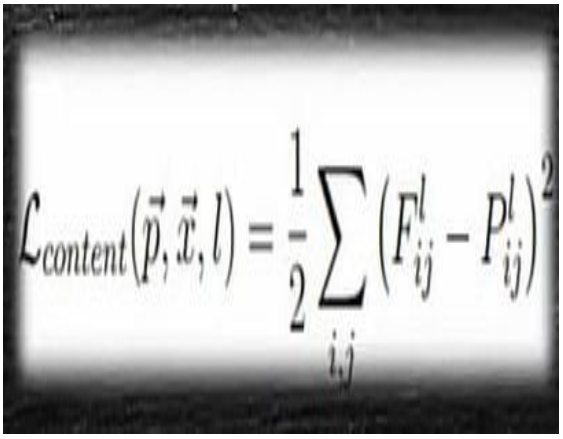
That's why we have multiple parts on 3_1 ,3_2, 3_3.

The key finding of this paper is that representation of content and style in convolution in your networks are separable.

It's helps to separate content and style in convolution neural networks.

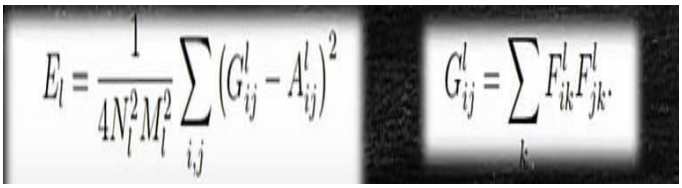
CONTENT LOSS AND STYLE LOSS

Let p and x be the original image and the image that is generated and P and F their respective feature representation in layer i . We define the squared -error loss between the two feature representations as the content loss.



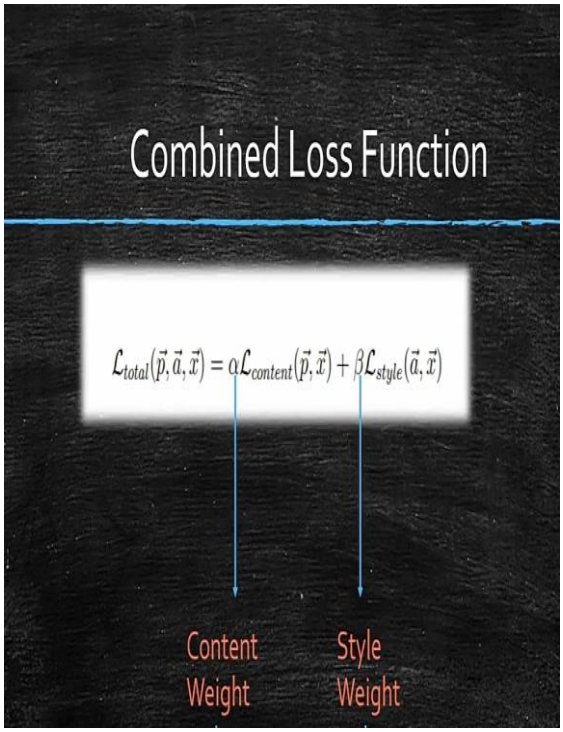
$$\mathcal{L}_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2$$

Let a and x be the original image and the image that is generated and Ai and Gi their respective style representation in layer i . The contribution of that layer to the total loss is then



$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}\left(G_{ij}^{l} - A_{ij}^{l}\right)^2 \qquad G_{ij}^{l} = \sum_{k} F_{ik}^{l}F_{jk}^{l}$$

We will combine these functions to know some constant times content loss plus some constant time style loss and these are called alpha and beta and so-called content weight and style with these parameters.



If we use hyper parameters, we can set hyper parameters to get very varied styles or varied levels of style transfers if content weight is kept very high it will be given more preference so less style will be applied. If style weight is more then, the content is kept very low.

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE Requirements:

- Any Processors(intel i5/i7/Xenon)
- Windows 10 64 bit
- 8 GB RAM,10 GB HDD Space

SOFTWARE Requirements:

- Anaconda Framework for Python IDE & Packages (Recommended),
- JupyterNotebook,Tensorflow machine Learning library
- Programming Language: Python

LIMITATIONS AND FRAMEWORK

It requires a lot of time and computation hardware to train different style images.

There will be slight difference in producing style images for different content images because of dependency of precision.

ACKNOWLEDMENTS

We would like to express our special thanks of gratitude to our mentor Mr. Meeravali Shaik who gave us golden opportunity to do this wonderful project on this topic which also helped us in doing lot of research and we came to know about so many new things .We are really thankful to them.

Secondly, We would also like to thank my friends who helped us a lot in finalizing this project within the limited time frame.

REFERENCES:

- A Neural Algorithm of Artistic Style, 2016 - Leon A. Gatys, Alexander S. Ecker, Matthias Bethge .
- Image Style Transfer Using Convolutional Neural Networks, 2016 - Leon A. Gatys, Alexander S. Ecker, Matthias Bethge
- Neural Style Transfer: A Review, 2018 - Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, Mingli Song
- Kumar, R.; Dey, A.; Smarandache, F.; Broumi, S. A study of neutrosophic shortest path problem. In Neutrosophic Graph Theory and Algorithms; Smarandache, F., Broumi, S., Eds. ; IGI-Global, 2019;Chapter 6, pp 144-175.[9]
- Kumar, R.; Edalatpanah, S. A.; Jha, S.; Singh, R. A novel approach to solve gaussian valued neutrosophicshortest path

problems. International Journal of Engineering and Advanced Technology2019,8, 347-353.[10]

- Kumar, R.; Edaltpanah, S. A.; Jha, S.; Broumi, S.; Dey, A. Neutrosophic shortest path problem. Neutrosophic Sets and Systems2018,23, 5-15.[11]
- Mohapatra, H.; Panda, S.; Rath, A. K.; Edalatpanah, S. A.; Kumar, R. A tutorial on powershell pipelineand its loopholes. International Journal of Emerging Trends in Engineering Research2020,8 (4).[12]