Stereo vision neural algorithm to process large scale images to reconstruct artistic style images.

Thiwanka Jayasiri
COSC428
Master's in Applied Data Science
University of Canterbury

Supervisor: Prof. Richard Green
Department of Computer Science
University of Canterbury

Introduction

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 were recently demonstrated by a class of biologically inspired vision models called Deep Neural Networks. Here I've used a latest AI method introduce based on Deep Neural Network that creates artistic images of high perceptual quality using stereo vision camera.

The AI method uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images. Moreover, considering the striking similarities between performance-optimized artificial neural networks and biological vision, in this research I intend to carry out a research to improve the algorithmic understanding of how humans' creativity and how the they perceive an artistic imagery in a real-world scenario.

Problem statement

Although many research being done in the area of the very large image recognition , there are only few publicly available research articles are there to understand the creativity aspects of image recognition and visual context building in human mind being published, hence I intend to carry out a research to understand the real time large scale image recognition using a stereo depth camera and bring it back the same image to artistic paradigm to explore how the neural networks excel in the semantic creative aspects.

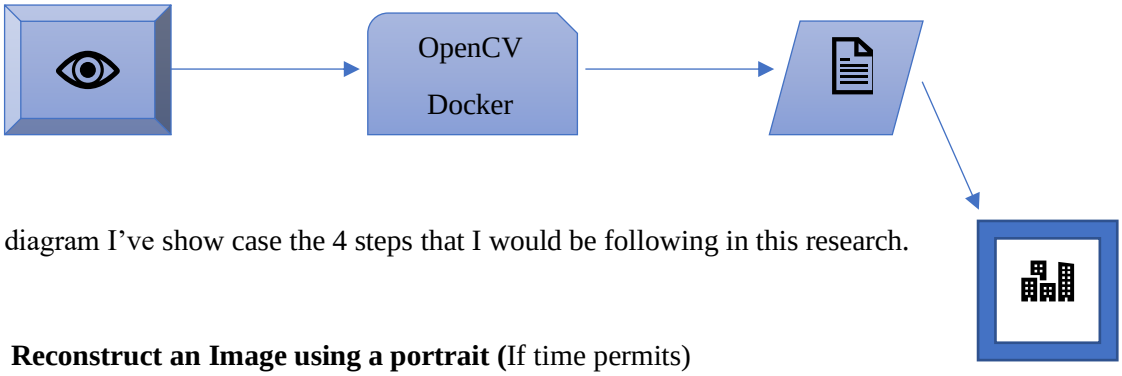
I've referred to an earlier research paper in this area, according to that the results presented in the main text were generated based on the VGG-Network, a Convolutional Neural Network that rivals human performance on a common visual object recognition benchmark task23 and was introduced and extensively described in the paper ***'Very deep convolutional networks for large scale image recognition '*** (Arge & Mage, 2015).

I intend to use the feature space provided by the 16 convolutional and 5 pooling layers of the 19-layer VGG Network. There is proposed method already which could deployed using the Google Colab under a publicly available repository and can explore the code base using Caffe and Caffe 2 frame work (Gatys, Ecker, Bethge, & Sep, 2015)or alternatively we can use the fast.ai frame work built on Pytorch to work on the same. In previous studies the images being collected from different sources and performed the synchronization. Here I'm trying to represent is that live image capturing using depth camera; using a depth camera since I'm focusing on three elements: baseline, resolution and focal length. By using a long-range depth camera, I believe I would be able to produce quality semantic artistic images.

Methodology

1) Stereo Vision Camera
2) OpenCV docker for the stereo vision camera to capture real time images

3) Python Script for the neural networks in this case VGG-16 could change accordingly.
    a. Google Colab Code base, intend to use GPU/TPU
    b. Several neural networks to optimize the quality output
4) Artistic Image Save



In this diagram I've show case the 4 steps that I would be following in this research.

5) **Reconstruct an Image using a portrait (**If time permits)

The results presented in the main text were generated based on the VGG-Network, a Convolutional Neural Network that rivals human performance on a common visual object recognition benchmark task23 and was introduced and extensively described in.22 We used the feature space provided by the 16 convolutional and 5 pooling layers of the 19-layer VGG Network. We do not use any of the fully connected layers. The model is publicly available and can be explored in the caffe-framework.24 For image synthesis we found that replacing the max-pooling operation by average pooling improves the gradient flow and one obtains slightly more appealing results, which is why the images shown were generated with average pooling.

Generally, each layer in the network defines a non-linear filter bank whose complexity increases with the position of the layer in the network. Hence a given input image ~x is encoded in each layer of the CNN by the filter responses to that image.

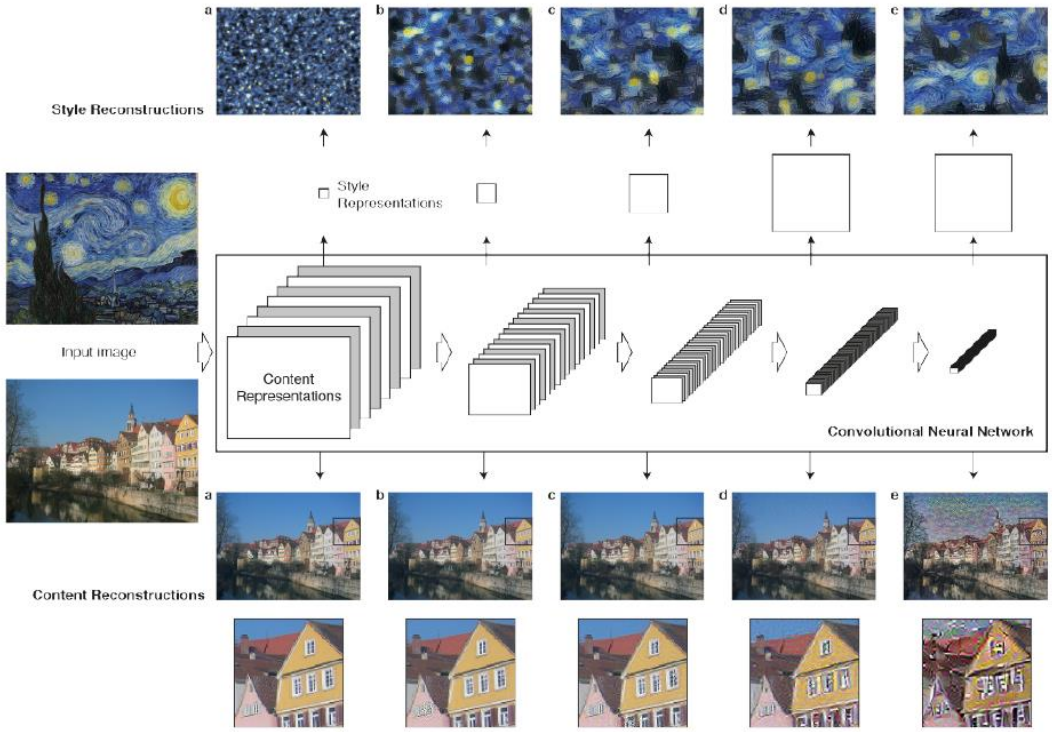
The class of Deep Neural Networks that are most powerful in image processing tasks are called Convolutional Neural Networks. Convolutional Neural Networks consist of layers of small computational units that process visual information hierarchically in a feed-forward manner (Fig 1). Each layer of units can be understood as a collection of image filters, each of which extracts a certain feature from the input image. Thus, the output of a given layer consists of so-called feature maps: differently filtered versions of the input image.

When Convolutional Neural Networks are trained on object recognition, they develop a representation of the image that makes object information increasingly explicit along the processing hierarchy(Gatys & Ecker, 2015). Therefore, along the processing hierarchy of the network, the input image is transformed into representations that increasingly care about the actual content of the image compared to its detailed pixel values.

(Fig 1, content reconstructions, see the diagram for details on how to reconstruct the image). Higher layers in the network capture the high-level content in terms of objects and their arrangement in the input image but do not constrain the exact pixel values of the reconstruction. (Fig 1, content reconstructions d,e). In contrast, reconstructions from the lower layers simply reproduce the exact pixel values of the original image (Fig 1, content reconstructions a,b,c). We therefore refer to the feature responses in higher layers of the network as the content representation, this was taken from the research paper :(Gatys et al., 2015).

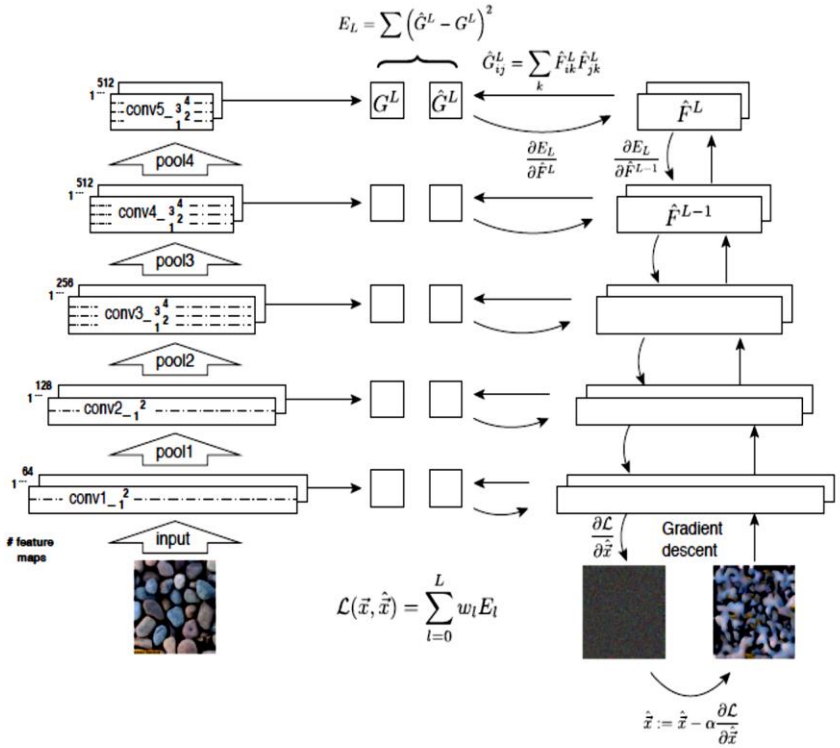
.

Figure 1



1/The Content representation and loss:

Figure 2



Above figure self-explanatory of the image processing through CNN and it's loss.

I followed the ideas described in the paper by defining two loss functions described in detail below, I will try to formulate the idea implemented and then show the results achieved right after in each of the following sections:

They have built a style representation quantity that computes the correlations between the different filter responses, where the expectation is taken over the spatial extend of the input image. These feature correlations are given by the Gram matrix:

$$\mathbf{Gram}_{\mathbf{ij}}^{\mathbf{l}} = \sum_{\mathbf{k}} \mathbf{F}_{\mathbf{ik}}^{\mathbf{l}} \mathbf{F}_{\mathbf{jk}}^{\mathbf{l}}$$

Let:
- a be the style image.
- x the generated image.
- Al style representations of a in layer l
- Gl style representations of x in layer l

As a loss function we use the mean-squared distance between the entries of the Gram matrix from the original image and the Gram matrix of the image to be generated,

$$\mathbf{E}_{\mathbf{l}} = \frac{\mathbf{1}}{\mathbf{4N}_{\mathbf{l}}^{\mathbf{2}}\mathbf{M}_{\mathbf{l}}^{\mathbf{2}}} \sum_{\mathbf{i,j}} \left(\mathbf{G}_{\mathbf{i,j}}^{\mathbf{l}} - \mathbf{A}_{\mathbf{i,j}}^{\mathbf{l}}\right)^{\mathbf{2}}$$

and the total loss is where *wl* are weighting factors of the contribution of each layer to the total loss is,

$$\mathbf{L}_{\mathbf{style}}^{\mathbf{a,x}} = \sum_{\mathbf{l}} \mathbf{w}_{\mathbf{l}} \mathbf{E}_{\mathbf{l}}$$

I intend to apply 5 weight combinations each time I take the weight as uniform across the layers and zero in non-selected layers: (this could change as I will continue to perform the work).

Proposed:

Configuration of the neural network,

Simple neural network architecture would look like as follows,

Convolution -> ReLU -> Pooling -> Convolution -> ReLU -> Pooling

Or

Convolution -> ReLU -> Convolution -> ReLU -> Pooling -> Convolution -> ReLU -> Convolution -> ReLU -> Pooling

VGG 16 Network

The neural style research paper (Arge & Mage, 2015)recommends the use of a pre-trained, very competent CNN called VGG 16. VGG 16 includes a classification section, but only its convolutional components are used in this result and probably would go ahead and adjust the network architecture as follows and it would perform accordingly with the test and train statistics.

Note: After every convolution listed, the ReLU activation function is applied. All filters in the network are of size 3x3. Pooling in the network is of size 2x2.

Convolutional block 1:
Conv1_1: 64 filters, each with depth 3.
Conv1_2: 64 filters, each with depth 64.
Max Pooling 1

Convolutional block 2:
Conv2_1: 128 filters, each with depth 64.
Conv2_2: 128 filters, each with depth 128.
Max Pooling 2

Convolutional block 3:
Conv3_1: 256 filters, each with depth 128.
Conv3_2: 256 filters, each with depth 256.
Conv3_3: 256 filters, each with depth 256.
Conv3_4: 256 filters, each with depth 256.
Max Pooling 3

Convolutional block 4:
Conv4_1: 512 filters, each with depth 256.
Conv4_2: 512 filters, each with depth 512.
Conv4_3: 512 filters, each with depth 512.
Conv4_4: 512 filters, each with depth 512.
Max Pooling 4

Convolutional block 5:
Conv5_1: 512 filters, each with depth 512.
Conv5_2: 512 filters, each with depth 512.
Conv5_3: 512 filters, each with depth 512.
Conv5_4: 512 filters, each with depth 512.
Max Pooling 5

2/The Style representation and loss:

Which I will look at when the scene being selected compared with the previous image styles and the ones being generated. My think is to get this work done using the original image and its loss function as mentioned above.

3/Mixing up the style and content representation to minimize the loss function

To generate the images that mix the content of a photograph with the style of a painting we jointly minimize the distance of a white noise image from the content representation of the photograph in one layer of the network and the style representation of the painting in several layers of the CNN

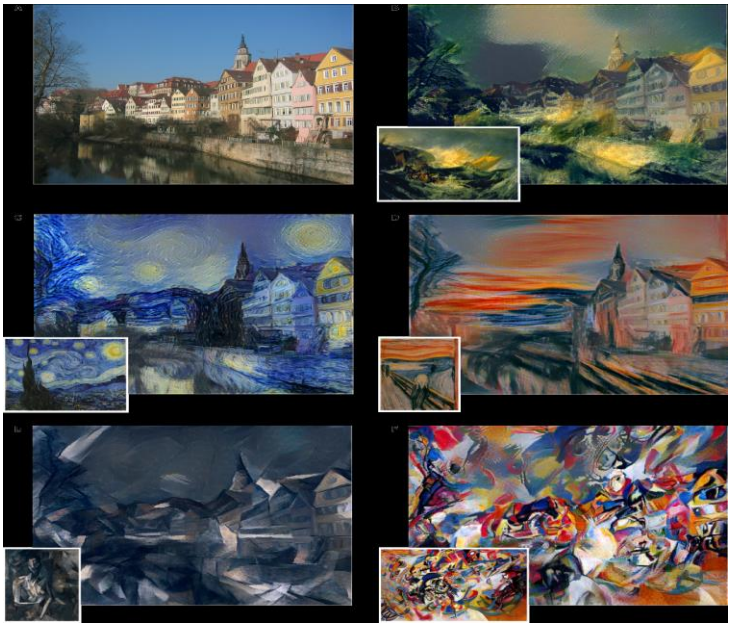
Let p be the content image and a be the painting. The loss function we minimize is

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

Synthesis method. Texture analysis (left). The original texture is passed through the CNN and the Gram matrices on the feature responses of several layers are computed. Texture synthesis (right). A white noise image is passed through the CNN and a loss function El is computed on every layer included in the texture model. The total loss function L is a weighted sum of the contributions from each layer. Using gradient descent on the total loss with respect to the pixel values, a new image is found that produces the same Gram matrices as the original texture.(Gatys & Ecker, 2015).

To gain more accuracy I require to make changes in the convolutional neural network; configuration.

Figure 02:



Images that combine the photography with the style of several well-known artworks. The images were created by finding an image that parallelly matches the content representation of the photograph and the style representation of the artwork.

We could adjust the number of layers and lead that in to a different visual experience although the past studies been done on converting a given image based on an exhibiting artwork, I would like to explore given an artwork can we build the real-world image. As it implies this goes in both ways first, we try and reproduce the real time image/video stream captured using a stereo vision camera (depth camera) and then apply then select some of the images of the drawings and convert those in to an artistic image or artistic video stream with the time duration. Then I will explore the events of recreating the original drawings to real world scenario. I would like to apply these to portraits moving forward given that the model works and try and reconstruct the famous image of Monalisa by DaVinci in to real world.

We can visualize the information at different processing levels in VGG by reconstructing the input image from only knowing the network's responses in a layer using the following loss function:

Let:

- p be the content image.

- x the generated image.
- Pl feature representations of p in layer l
- Fl feature representations of x in layer l

In the research papers, (Gatys et al., 2015) and(Arge & Mage, 2015) They used the Euclidean distance as loss function to measure the distance between both representations, the equation follows as ,

$$\mathbf{L_{content}(p, x, l) = \frac{1}{2}\Sigma\left(F_{ij}^{l} - P_{ij}^{l}\right)^{2}}$$

Final deliverables

Throughout this project I intend to understand more deeply feature representation in convolutional neural networks using the image /video streams captured through the stereo vision camera and how they can be exploited to produce very interesting results mentioned in the research article.

References

Arge, F. O. R. L., & Mage, C. I. (2015). VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION Karen, (Published as a conference paper at ICLR 2015 VERY), 1–14. Retrieved from https://arxiv.org/abs/1409.1556v6

Gatys, L. A., & Ecker, A. S. (2015). Texture Synthesis Using Convolutional Neural Networks Leon, 1–10. Retrieved from https://arxiv.org/pdf/1505.07376.pdf

Gatys, L. A., Ecker, A. S., Bethge, M., & Sep, C. V. (2015). A Neural Algorithm of Artistic Style, 3–7. Retrieved from https://arxiv.org/pdf/1508.06576.pdf