# A Neural Algorithm for Artistic Style

E4040.2016Fall.ASVM.report
Aayush Mudgal am4590, Sheallika Singh ss5136, Vibhuti Mahajan vm2486
*Columbia University*

## Abstract

*We aim to transcribe the style of an artwork onto an image using biologically inspired Convolutional Neural networks. These models are of high perceptual quality demonstrating the generative power of neural networks trained in purely discriminative fashion. We leveraged the weights of two different pretrained convolutional neural networks to obtain an optimum 'content' optimisation during the reconstruction of an image constrained to the 'style' component of another image.*

## 1. Introduction

The idea of texture generation in images arises from 90's when mathematical models of textures were developed. The work done by Zhu, Wu and Mumford defined a probabilistic model for texture generation in their 1997 paper on Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling [2]. They used an idea, that two images are indeed two samples of a particular texture iff their statistics match. The statistics used were histograms of a particular texture filtered with a number of filters : $\{hist(F * I),\ i = 1, ..., k\}$ . And whatever image has the same statistics is thought as a sample of texture $I$ . The main drawback was that the Gibbs sampling was employed (which is very slow). The model that we use today starts from a random image and subsequent adjustment of statistics in an iterative fashion, so that they match the desired results, and this is very similar to the scheme suggested by J. Portilla and P. Simoncelli in their seminal work on A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients[3]. The only difference is that now we use neural-networks based nonlinear filters as opposed to carefully handcrafted filters and also improved statistics to work upon as shown by Gatys et. al. in their seminal work on Texture Synthesis Using Convolutional Neural Networks[4] This model provides a new tool to generate stimuli for neuroscience and offers insights into the deep representations learned by convolutional neural networks. Within the model, textures are represented by the correlations between feature maps in several layers of the network. Across layers the texture representations increasingly capture the statistical properties of natural images while making object information more and more explicit.

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 [14]. We can directly visualise the information each layer contains about the input image by reconstructing the image only from the feature maps in that layer [5]: 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 while the reconstructions from the lower layers simply reproduce the exact pixel values of the original image. We therefore refer to the feature responses in higher layers of the network as the content representation.

In a nutshell, the feature representation preserve spatial information. If we dismiss it, we will know what objects are there on the picture, but will not be able to reestablish their location. The style can be thought as something independent of content, something we are left with if we let the content off. L. Gatys suggests to dismiss spatial information by computing correlations between the feature maps.

This framework can be further augmented by replacing the bag-of-feature-like statistics of Gram-matrix-matching by an MRF regularizer that maintains local patterns of the "style" exemplar: MRFs and dCNNs are a canonical combination — both models crucially rely on the assumption of locally correlated information and translational invariance. This equips the encoding of features in a dCNN with approximate Markovian consistency properties: Local patches have characteristic arrangements of feature activations to describe objects, and higher-up encoding becomes more invariant under in-class variation. This method produces plausible results for both art to photo and photo to art transfers. In particular, we can obtain meso-structures in the synthesized images. The classic data-driven approach to generative image modeling is based on Markov random fields (MRFs): We assume that the most relevant statistical dependencies in an image are present at a local level and learn a distribution over the likelihood of local image patches by considering all local k × k pixel patches in the example image.

## 2. Summary of the Original Paper

### 2.1 Methodology of the Original Paper

The core idea of the paper A Neural Algorithm of Artistic Style[4] is constrained optimisation; the constraints are of two types: *content* and *style*. Given a content image C and style image S the paper aims to generate an image X with content from C and style from S.

To design a loss function for the optimization process; it is desired to make an intermediate representation of X close to C i.e. $L_{content} = min_x || F_x - F_c ||$ ; where $F_x$ and $F_c$ are the feature representation of the image X and C respectively and $F \in C \times W \times H$ . This can be achieved because an input image is easily invertible from the feature representations of intermediate layers as shown in (F retains spatial information)[5] , [6].

Style is considered independent of content, something we are left with if we let the content off. The paper suggests to dismiss spatial information by computing correlations between the feature maps where the expectation is taken over the spatial extend of the input image. These feature correlations are given by the Gram matrix $G \in C \times C$ (note the absence of spatial dimension). The loss function thus becomes: $L_{style} = min_x || G(F_x) - G(F_c) ||$ .

To generate the images that mix the content of a photograph with the style of a painting the paper jointly minimise 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 a number of layers of the CNN.So the loss function becomes: $L = \alpha L_{content} + \beta L_{style}$

### 2.2 Key Results of the Original Paper

The original paper has generated results on the basis of 19 Layer VGG-network using the pre-trained feature-space of 16 convolutional and 5 pooling layers (using average pooling instead of max-pooling for better visual representations). The paper has presented an artificial neural system that achieves a separation of image content from style, thus allowing to recast the content of one image in the style of any other image. The authors have demonstrated this by creating new, artistic images that combine the style of several well-known paintings with the content of an arbitrarily chosen photograph i.e. deriving the neural representations for the content and style of an image from the feature responses of high performing Deep Neural Networks trained on object recognition.

The images are synthesised by finding an image that simultaneously matches the content representation of the photograph and the style representation of the respective piece of art (see Methods for details). While the global arrangement of the original photograph is preserved, the colours and local structures that compose the global scenery are provided by the artwork. Effectively, this renders the photograph in the style of the artwork, such that the appearance of the synthesised image resembles the work of art, even though it shows the same content as the photograph.

When matching the style representations up to higher layers in the network, local images structures are matched on an increasingly large scale, leading to a smoother and more continuous visual experience. Thus, the visually most appealing images are usually created by matching the style representation up to the highest layers in the network

## 3. Methodology

Building upon the model of the original paper[7] , we based our model on the convolution and pooling layers of VGG-19 and tried different versions of style and component values by taking different combinations of convolution layers while calculating the gram matrix and feature maps respectively.

The key finding of the original paper [7] is that an image can be represented separately in terms of content and style. Following the paper we have used convolution layers to extract the style and content feature mappings of the image. As mentioned in the paper extracting style representation from upto the higher layers of the CNN gives smoother images. Therefore conv1_1, conv2_1, conv3_1, conv4_1 and conv5_1 layers are used for style representation of an image. conv4_2 layer is used for content representation. Now, a random image is generated with pixel densities distributed according to Normal (0,1) distribution. Using this random image and optimisation an image is generated to have style of one image (art_image) and content of another image ( photo).

The optimisation method used is either RMSprop or Adam, which tried to maximise the correlation of style features between the generated image and art and correlation between content of photo and generated image.

Inspired by the paper [7], we went on extracting feature maps from layers of AlexNet [13]. Here ConvPool layer 1, ConvPool layer 2, ConvPool layer 3, ConvPool layer 4, ConvPool layer 5 are used to extract style features from the image and ConvPool layer 1 is used to extract content. Following the same procedure as for VGG net, optimisation is used on the randomly produced image from standard normal distribution to generate image having style of art_image and content of photo.

In the techniques above we actually are trying to maximise the statistical dependencies for feature maps

between images. In [8] it is stated that the most relevant statistical dependencies is at the local level.The idea is to maximise the correlation between the images at local level which as a result gives smooth, coherent generated image. So building upon this idea we tried to define new loss function which divides the input into patches and the sum of losses over all the patches gives the final loss function. $L_{content}$ is same as before, but $L_{style}$ becomes :

$$L_{style-MRF} = \sum_{i=1}^{m} || \Phi_i(F_x) - \Phi_i(F_c)||^2$$

Where $\Phi_i()$ is $G(F_{x_i})$ mapping function to extract features of patch i of $F_x$ and similar for $F_c$ .

## 3.1. Objectives and Technical Challenges

The objective of this project was to present an algorithm for combining the content of one image with the style of another image using state of the art convolutional neural networks. The algorithm should allow the user to tradeoff the relative weight of the style and content reconstruction terms. We replicated the stylised image generation model of the original paper[7] and tried to apply the model on non-photorealistic (artwork) synthesis by extracting the image layout at an abstract level using Markov Random Fields(MRF). This scheme is provided by C. Li and M. Wand in their paper on Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis[8]. This model works on the fact that the MRF regularizer prevents over-excitation artifacts and reduces implausible feature mixtures common to previous dCNN inversion approaches, permitting synthesizing photographic content with increased visual plausibility. Unlike standard MRF-based texture synthesis, the combined system can both match and adapt local features with considerable variability, yielding results far out of reach of classic generative MRF methods.

The discriminative design poses a problem: The corresponding dCNNs compress image information progressively over multiple pooling layers to a very coarse representation, which encodes semantically relevant feature descriptors in a semi-distributed (not spatially localized) fashion. While an inverse process can be defined [5], [6], it remains difficult to control: simply maximizing the class-excitation of the network leads to hallucinatory patterns[8]. Rather than that, we need to reproduce the correct statistics for neural encoding in the synthesis images.[7] suggests the uses the filter pyramid of the VGG network as a higher-level representation of images, benefitting from the vast knowledge acquired through training the dCNN on millions of photographs. VGG-network, a Convolutional Neural Network that rivals human performance on a common visual object recognition benchmark task. Training of this network involved huge time, memory and monetary resources. To get comparable results we had to load weights of pre-trained VGG-network into theano expressions. The performance of the model is subjective in nature and requires repetitive experimentation with varying model complexities and parameters to get visually plausible results. The challenge lies in the fact that there are endless combinations to check with no clear-cut direction in which to proceed while parameter selection and complexity determination.

## 3.2. Problem Formulation and Design

The workflow is presented in Figure1.

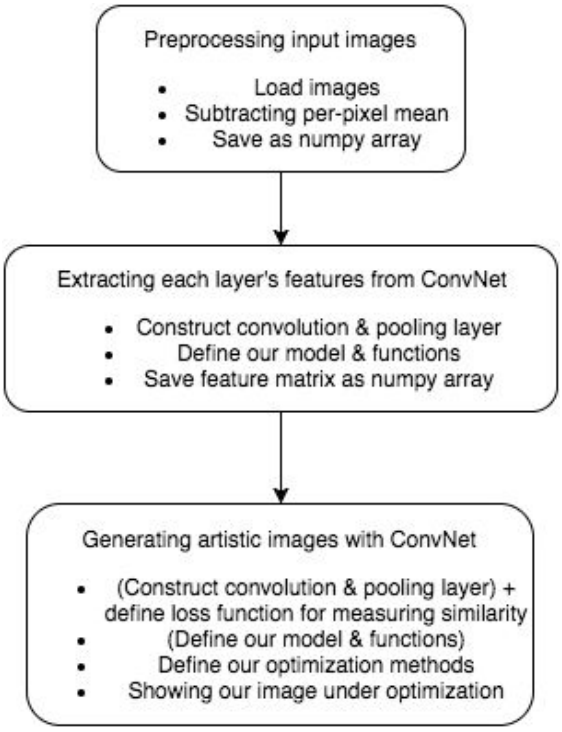



Figure1: System Overview

## 4. Implementation

This project can be divided into three major chunks: Obtaining feature maps for a pre-trained VGG-net, using a neural algorithm of artistic style to transcript the style of an image onto the content of another, improving upon the previous algorithms by taking local abstractions of style by replacing Gram Matrices by an MRF regularizer that maintains the local patterns of "style".

1. Obtaining weights of a pre-trained Convolutional network: VGG-network and ALEX-network.(complete this section)
2. Assigning the pre-trained weights to the respective layers

3. For the 2 images - art ( from which the style is to be generated) and photo ( for the content generation), outputs from layers are extracted and saved.
    a. VGG Net- Layers used for extracting style from art image are conv1_1, conv2_1, conv3_1, conv4_1 and conv5_1. Layer used for extracting content from photo is conv4_2.
    b. ALEX Net- Layers used for extracting style from art image are ConvPool layer 1, ConvPool layer 2, ConvPool layer 3, ConvPool layer 4, ConvPool layer 5. Layer used for extracting content from photo is ConvPool layer 1.
    c. Combining MRF and Neural Network:This algorithm can be applied to both of above CNN and same layers are used as above.
4. A random image of size 227*227 is generated using the Normal (0,1) distribution.
5. This random image is treated as parameter and the output for this image is recorded from each layer considered for style or content representation.
6. Image is generated by updating the parameter ( which is image to be generated here) optimisation algorithm.Optimisation algorithms used are ADAM and limited-memory BFGS. See section 4.1 for details.
7. The cost function optimised by above algorithms is $L = \alpha L_{content} + \beta L_{style}$ .
    a. $L_{content} = min_x || F_x - F_c ||$ ; where $F_x$ and $F_c$ are the feature representation extracted from specified layers (see point 3), of the photo and generated image.
    b. $L_{content}$ is same for all algorithms- VGG Net , ALEX Net and for Combining MRF and CNN .
    c. $L_{style} = min_x || G(F_x) - G(F_c) ||$; where feature correlations are given by the Gram matrix G, $F_x$ and $F_c$ are the feature representation extracted from specified layers (see point 3), of the art image and generated image.
    d. If MRF and CNN are combined then features are divided into patches to extract features locally from every patch. In this case only the $L_{style}$ differs from previous algorithms. Here $L_{style}$ is defined as : $L_{style-MRF} = \sum_{i=1}^{m} || \Phi_i(F_x) - \Phi_i(F_c) ||^2$ ; where $\Phi_i()$ is G( $F_{x_i}$) mapping function to extract features of patch i of $F_x$ and similar for $F_c$ .
8. α and β are hyper parameters here. In results section one can see different generated images for different α/β ratio.

## 4.1. Deep Learning Network

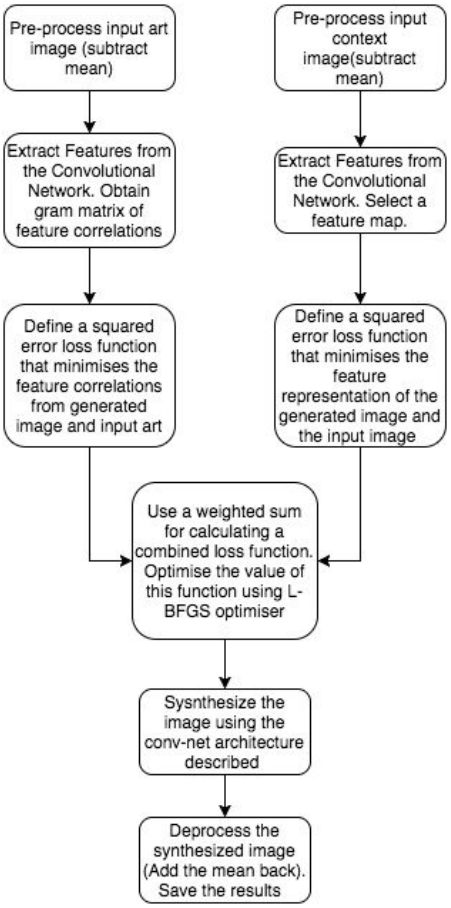



Figure2: Working Model

The architectures of VGG-net and ALEX-net are shown in the appendix. The working model is presented in Figure 2. For the MRF model, the only difference is while calculating the loss function for the style component, where instead of taking a global constraint for non-photorealistic synthesis, we use a local constraint that works for both photorealistic and non-photorealistic synthesis.

For SGD based gradient descent updates, we employed 2 algorithms for training the convolutional networks namely ADAM and L-BFGS.

Adaptive Moment Estimation (Adam) [10] is an

optimisation method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients similar to momentum. The gradients are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively, hence the name of the method.The pseudo code for the method is written in the next section.

BFGS is numerical computation algorithms that is used in constrained optimization: Find derivatives → Find direction that yields maximum estimated objective function change → Use line search to find optimal step size → Move, and repeat → One additional 'correction term' in BFGS → Treatment of roundoff errors → tolerances is different as compared to quasi-newton method. Limited-memory BFGS (L-BFGS or LM-BFGS) is an optimization algorithm in the family of quasi-Newton methods that approximates Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm using a limited amount of computer memory and is thus a popular algorithm for parameter estimation in machine learning. L-BFGS uses an estimation to the inverse Hessian matrix to steer its search through variable space, but where BFGS stores a dense $n \times n$ approximation to the inverse Hessian ($n$ being the number of variables in the problem), L-BFGS stores only a few vectors that represent the approximation implicitly. Due to its resulting linear memory requirement, the L-BFGS method is particularly well suited for optimization problems with a large number of variables. Pseudo code is provided in the next section.

## 4.2. Software Design

Pseudo code for Adam:

Require: Step-size α.
Require: Decay rates ρ1 and ρ2, constant ε.
Require: Initial parameter θ.

Initialize 1st and 2nd moment variables s = 0, r = 0.
Initialize time step t = 0.
while Stopping criterion not met do

Sample a minibatch of m training set examples {x1, ..., xm}.
Set g = 0.
for i = 1 to m do

Compute gradient: g←g+∇θL(f(xi;θ),yi)/m
end for
t←t+1 Get biased first moment:
s ← ρ1s + (1 − ρ1)g Get biased second moment:
r ← ρ2r + (1 − ρ2)g2
Compute bias-corrected first moment:
$\hat{s} \leftarrow \frac{s}{1-\rho 1^t}$
Compute bias-corrected second moment:
$\hat{s} \leftarrow \frac{r}{1-\rho 2^t}$
Compute update:
$\Delta\theta \leftarrow -\alpha \frac{s}{\sqrt{r}+\varepsilon} \odot g$ (element-wise)
Apply update: θ ← θ + Δθ
end while

The authors of Adam propose default values of 0.9 for ρ1, 0.999 for ρ2, and 10−810−8 for ε. They show empirically that Adam works well in practice and compares favorably to other adaptive learning-method algorithms. But for our project, we discovered that L-BFGS gave more visually plausible and faster results.

Pseudo code for L-BFGS:
function FINDSEARCHDIRECTION(f,x)
z ← ∇f(x)
for i ← k − 1, k − 2, . . . , k − m do
αi ← ρi ·s $i^T$ .z .
Store αi
z ← z − αi ·yi
end for
z ← H0 ·z
for i ← k − m, k − m + 1, . . . , k − 1 do
βi ← ρi ·y $i^T$ i ·z
z ←∼z +si · (αi − βi )
end for
return −z
end function

## 5. Results and Insights

### 5.1. Project Results

(Note: Please check appendix for larger/ clearer images)

As mentioned above, α is the weight of content image and β is the weight of weight of style image that is used while calculating the total loss function. Figure 3 depicts reconstructions of the desired image with varying degrees of α/β using the feature maps of VGG-net. Figure 4 depicts the reconstruction using the feature maps of ALEX-net.

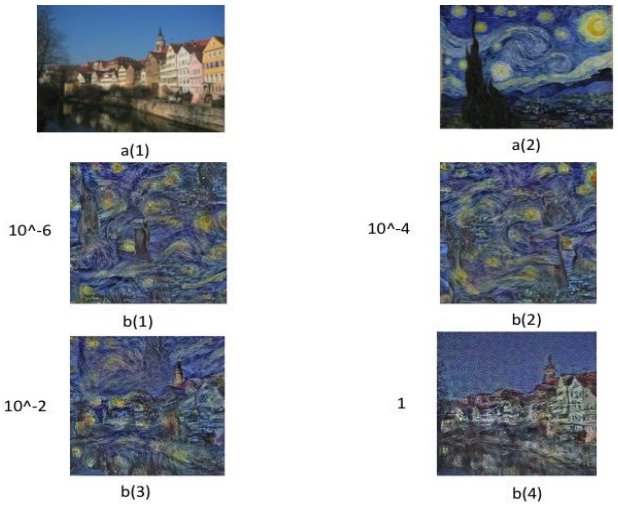



Figure 3

Figure3.a shows original images used for the experiment. a(2) is the artwork whose texture is mapped onto the content of a(1). Figure3.b depicts the generated image with different values of $\alpha/\beta$ . As it is evident from the figure $\alpha/\beta = 10^{-2}$ gave the most visually appealing result. We have progressed in our project by taking this ratio. (For a better representation, these images can be easily generated through the code provided in [1] by setting appropriate values for the parameters)

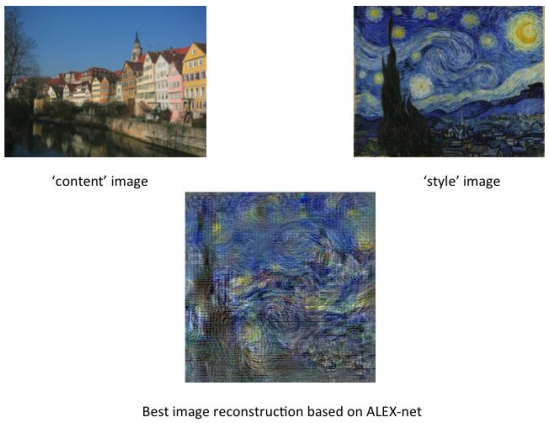



Figure 4

The resulting image based on ALEX-net is not as visually appealing as compared to it VGG counterpart. This is due to the more complex structure of VGG-net making its performance comparable to that of biological visual networks.

In the total loss function : $L = \alpha L_{content} + \beta L_{style}$, $\alpha$ and $\beta$ play an important role as it can be seen in figure 3 that changing $\alpha/\beta$ ratio considerably effects the generated image.

For $\alpha/\beta = 10^{-2}$ we have obtained the most visually convincing generated image, when using a(1).figure3 for content and a(2).figure3 for style features. Therefore using $\alpha/\beta = 10^{-2}$ for ALEX-net loss function. But as it can be seen in figure-4 that Alex-net does not give convincing generated image for same a(1) and a(2) as of VGG-net.

As VGG-net gave visually better results than ALEX-net, we have used VGG-net to generate images with style from one image and content from another. We have used $\alpha/\beta = 10^{-2}$ ratio for total loss function. Figure 5 shows the generated images (at right-most corner) with content of bridge image and style from middle images.

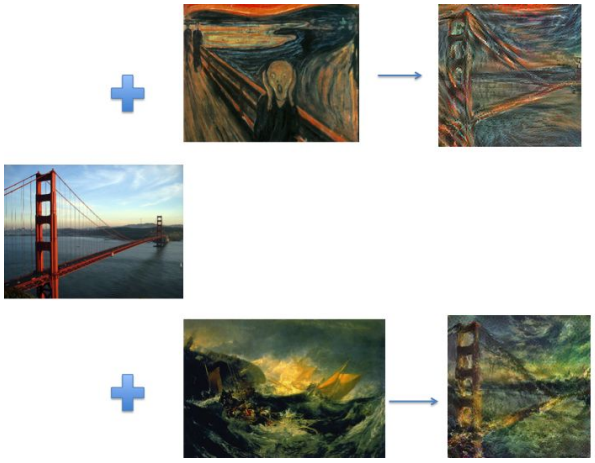


Figure5
Different Textures on the same content Image

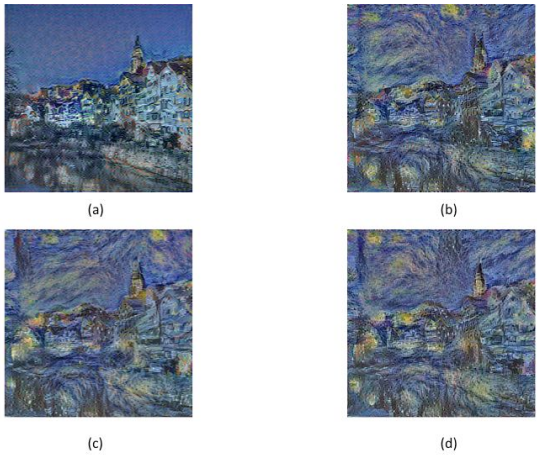



Figure 6
Figure 6 depicts the results generated by VGG-based model with the neural algorithm as presented in [7]. The four images depict how the generated image changes when we take different numbers of feature maps while calculating the gram matrix for style loss computations. .a) has the layers [conv1_1, conv2_1, conv3_1]. b) has the layers [conv1_1, conv2_1, conv3_1, conv4_1], c) has the layers[ conv2_1, conv3_1, conv4_1, conv5_1] ,and d) has the layers [conv1_1, conv2_1, conv3_1, conv4_1, conv5_1]

To extract style features from the most relevant layers of the CNN we tried different combination of layers. Figure 6 shows different generated images obtained for different number of layers. Model with style feature extracted from all the convolutional layers of VGG-net gave the most visually appalling generated image. Style of the art image is captured the most from this model.

Figure 7
The left image depicts the style reconstruction using the algorithm provided in the original paper [7] and the right image is the result for the MRF based local style-optimisation algorithm provided in [8] . The second image does give encouraging results for style transfer between 2 images but the results are not that pronounced.

Figure 7 shows the generated image for combination of MRF and CNN model.

## 5.2. Discussion of Insights Gained

1. Despite the advantages in patch matching and blending, it is still possible for a dCNN to generate implausible results. For example, the matched patches from different layers might not always fire at the same place in the image space.
2. The MRF prior only offers advantages when style and content images consists of similarly shaped elements without strong changes in perspective, size, or shape, as covered by the invariance of the high-level neural encoding. Otherwise, artifacts might occur. For pure artistic styles, the increased rigidity can then be a disadvantage.
3. We observed that the style representation better match as we include higher layers of the network for style feature calculation. This is primarily because the local image structure are better captured when including the contribution of higher layers of the network for calculating style features.
4. Results of the VGG network are more appealing than that of the ALEX-net (Figure 4). The VGG architecture has significantly more number of layers than the ALEX-net architecture. Thus making VGG to resemble more with biological visual networks.

5. Since the $L_{Total}$ converges very fast, good quality results are generated in few epochs (~10-15)

## 6. Conclusion

1. The representations of style and content in the CNN are separable, and therefore they could be suitably modified to generate perceptually meaningful image.
2. Local image structures captured by the style representation increase in size and complexity when including style features from higher layers of the network. This can be explained by the increasing receptive field sizes and feature complexity along the network's processing hierarchy.
3. Deeper layer architectures tend to perform better than shallower networks.
4. Having carefully implemented the model as proposed by [7], we are able to separate and recombine content and style of arbitrary images, to generate artistic images of high perceptual quality.

As a further improvement of the project, it would be interesting to have a combination of content and style of arbitrary images, such that it retains the colour shades of the content image while learning the style.. This ability to separate the content from style, and the ability to manipulate them to generate appealing arts might give a good understanding about how humans perceive things.


## 6. Acknowledgement

We would like to thank Professor Zoran Kostic for his indelible teachings and all the TAs of ECBM4040 (Fall 2016) for their constant guidance and support in coursework, assignment and project.


## 7. References


Include all references - papers, code, links, books.

[1] https://bitbucket.org/e_4040_ta/e4040_project_asvm

[2] Zhu et. al., "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling", *International Journal of Computer Vision 27(2), 107–126 (1998)*

[3] Portilla & Simoncelli, "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients", *International Journal of Computer Vision 40(1), 49–71, 2000*

[4] Gatys et. al., "Texture Synthesis Using Convolutional Neural Networks"

[5]A.Mahendran, A.Vedaldi, "Understanding Deep Image Representations by Inverting Them"

[6] *A.Dosovitsky, T.Brox*, "Inverting Visual Representations with Convolutional Networks"

[7] Gatys, Ecker, Bethge, "A Neural Algorithm of Artistic Style"

[8]C.Li, M.Wand, "Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis"

[9]A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. http://googleresearch.blogspot. com/2015/06/inceptionism-going-deeper-into-neural.html, 2015

[10]Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations,

[11] https://github.com/Lasagne/Recipes/

[12] Keras Examples (Neural Style Transfer) https://github.com/fchollet/keras/tree/master/examples

[13] Alexnet Architecure Weights: https://dataverse.scholarsportal.info/dataset.xhtml?persistentId=hdl:10864/10911

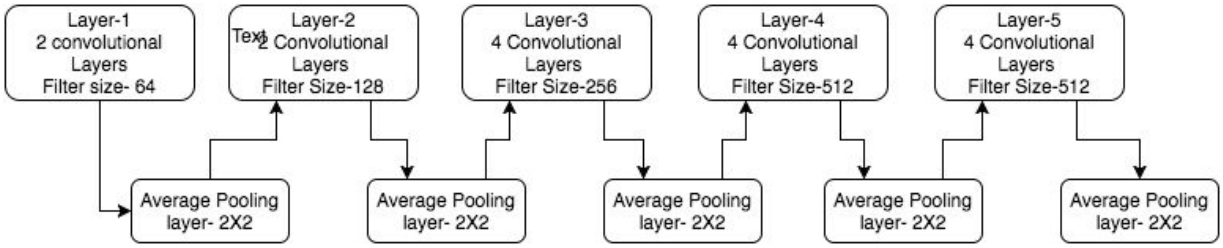
[14] Gatys, L. A., Ecker, A. S. & Bethge, M. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. arXiv:1505.07376 [cs, q-bio] (2015). URL http://arxiv.org/abs/1505.07376. ArXiv: 1505.07376.

## 8. Appendix

### 8.1 Individual student contributions - table

| uni | am4590 | ss5136 | vm2486 |
|---|---|---|---|
| Last Name | Mudgal | Singh | Mahajan |
| Fraction of (useful) total contribution | 1/3 | 1/3 | 1/3 |

8.2




Architecture of the VGG-net used:

Figure 3

Figure3.a shows original images used for the experiment. a(2) is the artwork whose texture is mapped onto the content of a(1). Figure3.b depicts the generated image with different values of $\alpha/\beta$ . As it is evident from the figure $\alpha/\beta = 10^{-2}$ gave the most visually appealing result. We have progressed in our project by taking this ratio. (For a better representation, these images can be easily generated through the code provided in [1] by setting appropriate values for the parameters)

a(1)

a(2)

10^-6

b(1)

10^-4

b(2)

10^-2

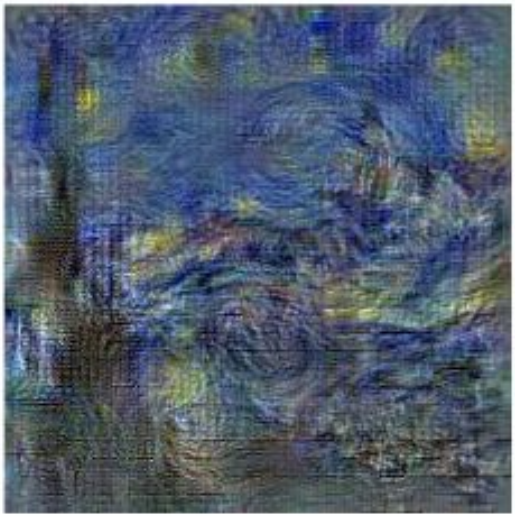
b(3)

1

b(4)

'content' image

'style' image



Best image reconstruction based on ALEX-net

Figure 4
The resulting image based on ALEX-net is not as visually appealing as compared to it VGG counterpart. This is due to themore complex structure of VGG-net making its performance comparable to that of biological visual networks.

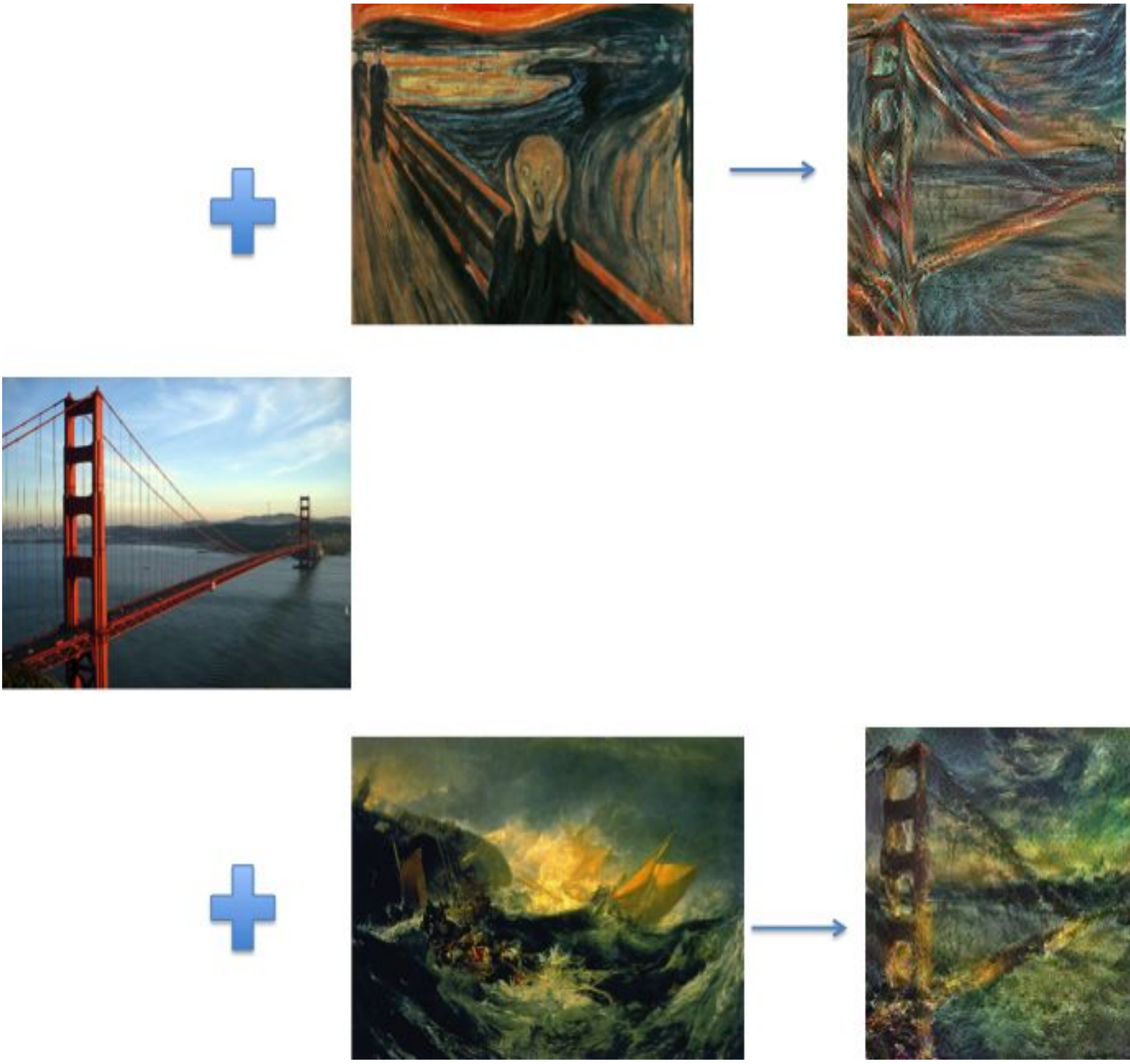


Figure5
Different Textures on the same content Image

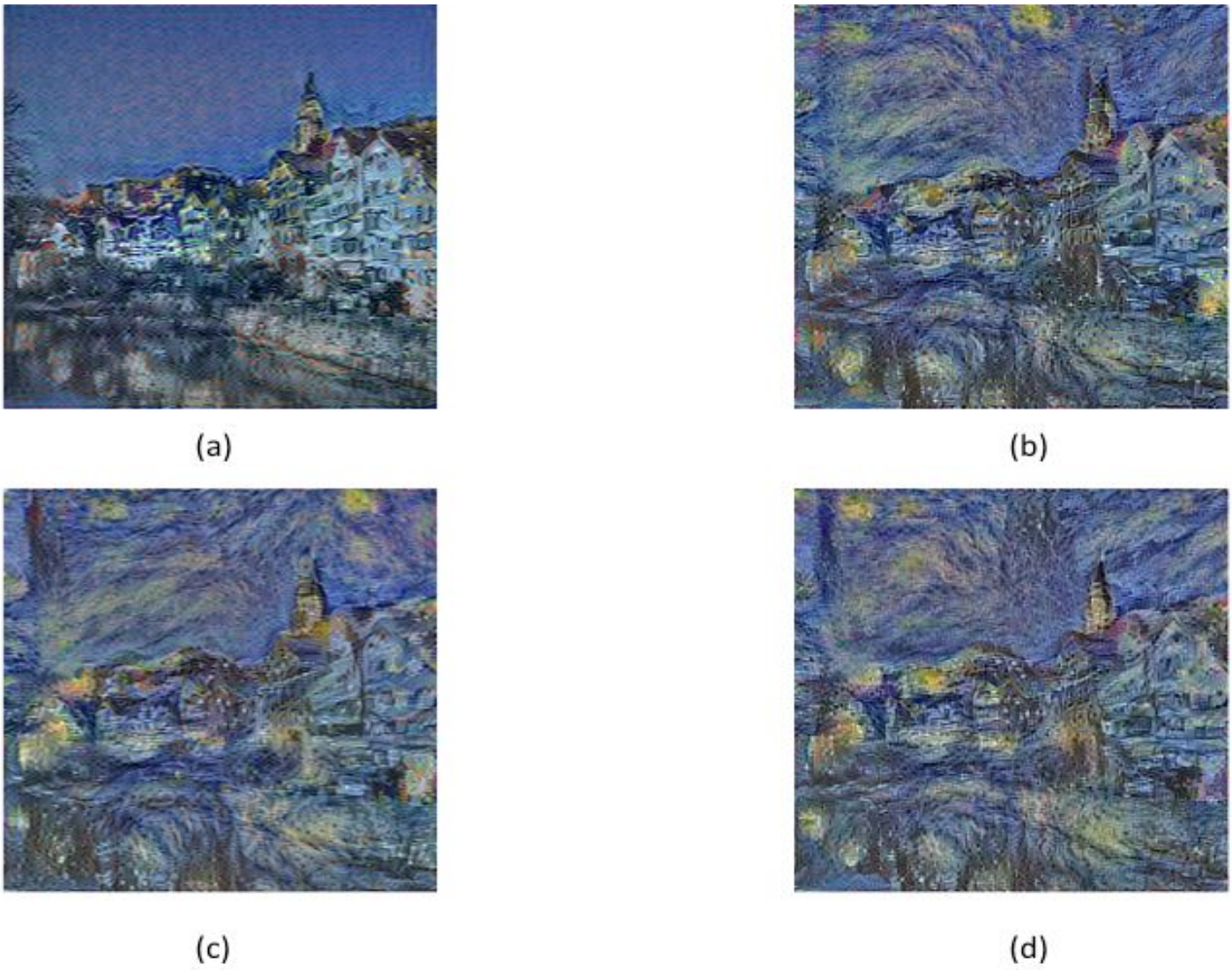



Figure 6

Figure 6 depicts the results generated by VGG-based model with the neural algorithm as presented in [7]. The four images depict how the generated image changes when we take different numbers of feature maps while calculating the gram matrix for style loss computations. .a) has the layers [conv1_1, conv2_1, conv3_1]. b) has the layers [conv1_1, conv2_1, conv3_1, conv4_1], c) has the layers[ conv2_1, conv3_1, conv4_1, conv5_1] ,and d) has the layers [conv1_1, conv2_1, conv3_1, conv4_1, conv5_1]

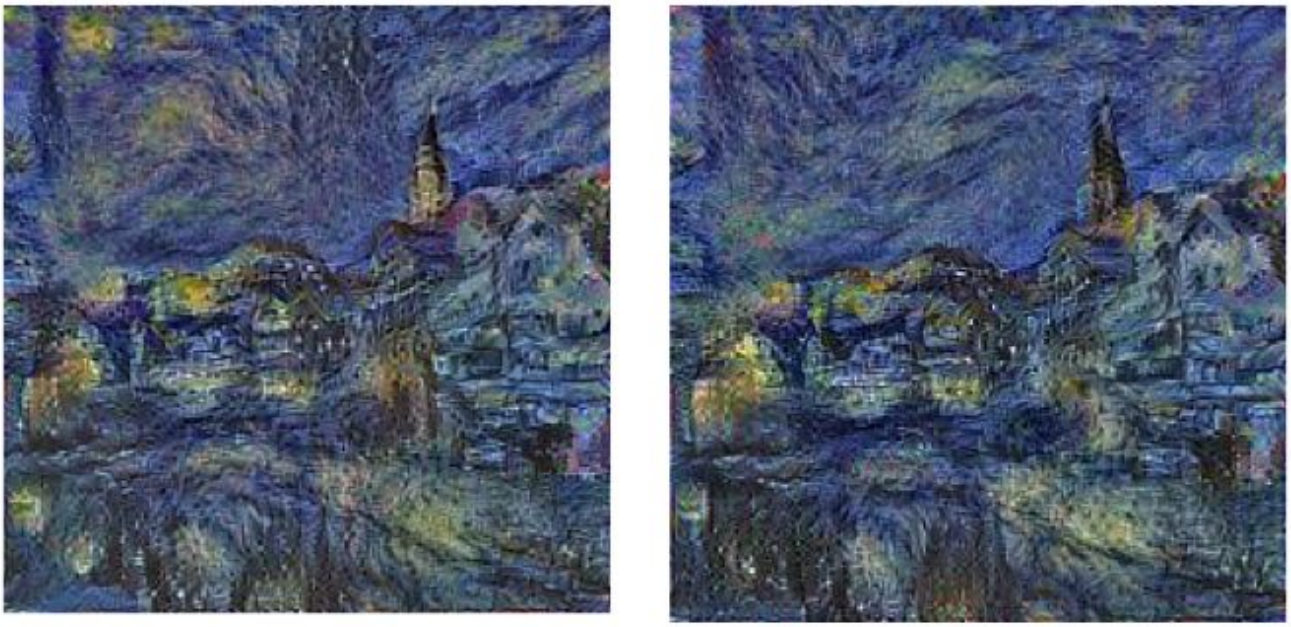


Figure 7
The left image depicts the style reconstruction using the algorithm provided in the original paper [7] and the right image is the result for the MRF based local style-optimisation algorithm provided in [8] . The second image does give encouraging results for style transfer between 2 images but the results are not that pronounced.