FULLY CONVOLUTIONAL NEURAL NETWORK FOR EDGE-AWARE DOMAIN TRANSFORM IMAGE PROCESSING COMPUTER VISION APPLICATIONS

A dual degree project report

submitted by

MOHAMMED KHANDWAWALA

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JUNE 2021

THESIS CERTIFICATE

This is to certify that the thesis entitled **Fully Convolutional Neural Network for Edge-Aware Domain Transform Image Processing Computer Vision Application** , submitted by **Mohammed Khandwawala (EE16B117)**, to the Indian Institute of Technology Madras, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Mansi Sharma
Research Guide
INSPIRE FACULTY
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 1st June 2021

ACKNOWLEDGEMENTS

I would like to thank Dr. Mansi Sharma for her insights, motivation and constant support in guiding me through this Dual Degree project.

ABSTRACT

KEYWORDS: Sketching; Deep Learning; Stylization; Bilateral Grid

Deep Learning (DL), a subset of machine learning algorithms, has been the front runner in solving a wide range of computer vision problems. They do come with caveats like the need for humongous data to be trained. This leads to huge training time and the models themselves being large to be run on hardware on the edge.

Deep learning based solutions for computer vision problems outperforms other solutions by a far margin, that they become the de facto solution. Sketch Generation is one of the areas in the art space where DL methods are assisting artists in the modern digital art landscape.

This thesis presents a novel technique to create sketches from real-life scene based on our BG3DIN-DeepSketch. In contrast to existing traditional approaches that require tweaking of multiple parameters we provide automatic model that can handle more general and challenging input of real life scenes ranging simple urban landscape to complicated textures such as trees. We convert the real like scene into a simplified sketch version which is then usable for vector art. This is all done in a fully automatic way without user intervention. Our model consists of a Inception Szegedy *et al.* (2015) like 3D convolutional neural network inspired by and outputs a simplified sketch which has the same dimensions as the input image. In order to teach our model to learn better edge details, we use bilateral grid data structure as our model input. Finally, we display our results and various NPR (Non Photo-realistic Rendering) applications ranging from colourization, domain transform stylization and sketch generations.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
Res-Net	Residual Network
NPR	Non Photorealistic Rendering
NFT	Non-fungible Token

CHAPTER 1

SKETCHING

1.1 Motivation

With the rise of digital art and sketching in the digital format has reached peak. It has wide and diverse application from traditional manga art to mainstream abstract art. With the rise of NFT's digital art is now regarded on the same level as tradition art. Sketching is often the first step in creating a scene which is an iterative process of refinement and design. The process of manually tracing a scene to produce a sketch, as one would expect, is time-consuming and tedious. In this work we aim at automatically converting image of a scene into a simplified sketch. As more and more artists are using digital tablets than paper and pencil.

Our approach can speed up the initial step as it can get the vision of the artist on the paper. Our method is based on 3-D Bilateral Grid based Convolution Neural Networks (CNN). This bilateral grid helps the network with processing high dimensional features. For our work we use BIPED edge detection dataset for training as it has the most accurate training labels . Our CNN based sketching method has two important advantages: It learns all the details of different textures such as trees or urban landscape from training data and it is fully automatic.

1.2 Problem definition

Sketching is a sub-domain of Non photo-realistic rendering (NPR). NPR comprises of algorithms that try to animate and represent items inspired by drawings, cartoons, anime, painting and other sources that do not feature photo-realism. This has wide variety of applications in animation, comics and manga.

Sketching is one of the most basic pictorial languages in visual arts to abstract natural scenes. Sketch synthesis methods generally fall into two categories: 2D image-based rendering and 3D model-based rendering. The majority of recent research along the line to mimic some artistic style. There are variety of sketching results with its own unique nature just artists in real life.

CHAPTER 2

Related work

In this section, we discuss different sketching methods that create pencil sketches from colour image.

Most of pencil sketching works are about portrait sketch in comparison with natural image sketching Wang *et al.* (2017) .Compared with the natural scenery sketch, portrait sketch has more regularity characteristics in the process of analysis. Xu *et al.* (2011) advocated an L_0 smoothing filter which is applicable to pencil stroke generation. Also a lot related method require adjusting certain parameters manually and lack generalization in handling different scenarios.

Traditional Methods

Using tradition edge detection methods paper Lu etal. Cewu Lu (2012) they propose a new system to produce pencil drawing from natural images. By generating Line drawing from image gradients and a tone map that contains textures. The results are obtained by combining them. The results contain various natural strokes and patterns,and are structurally representative.

Another interesting work Zhang *et al.* (2017) proposes method to first extracts edges at different resolutions. Firstly, edges are extracted from an image. The main edges are distorted strongly, and the texture edges are weakly distorted. Next, the brightness of each layer is varied. Then synthesize them and apply the texture. The results have a very natural look.

CNN Based Method

In recent years, paper Lu etal. Cai and Song (2018) has made significant improvement in pencil-drawing-synthesized method. The sketch and style are learned from the edge of original natural image and one pencil image exemplar of artist's work. They accomplished this using the convolutional neural network feature maps of a natural image and an exemplar pencil drawing style image. Generated by their VGG-16 like architecture. Large-scale bound-constrained optimization(L-BFGS) is applied to synthesize the new pencil sketch whose style is similar to the exemplar pencil sketch. The output from their method is very natural and has close resemblance with artists style.

Another method by Li *et al.* (2019) presents more control in styling. There method breaks the problem in two parts, first part involves outline extraction. They use the Extended Difference-of-Gaussians (XDoG) filter Winnemöller *et al.* (2012) for the outline task. The method also needs a tone map with an edge map for shading task , which is then processed by their Auto-encoder architecture. Then the result on of shading task and outline task are combined to give the final output. This method provides versatility in the form of tone map and can therefore produce outputs with different styles. However this method needs an additional outline extraction step.

CHAPTER 3

Proposed Algorithm - BG3DIN-DeepSketch

3.1 Method

The model is based on the idea of processing image features as bilateral grid in our network.

Bilateral Grid

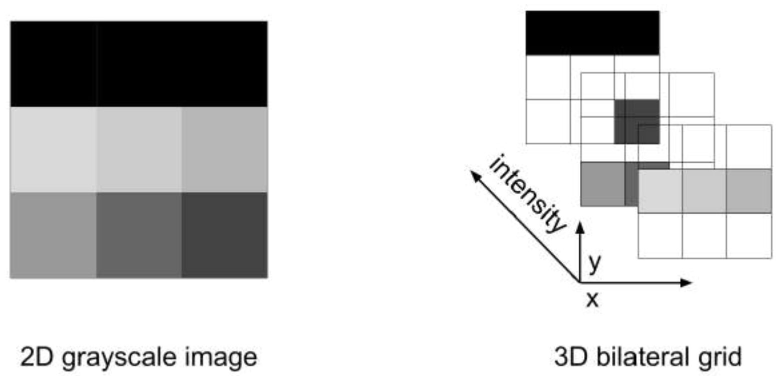

Figure 3.1: 2D image represented as a 3D bilateral grid

Bilateral grid was introduced in Chen *et al.* (2007) as a data structure for images to perform edge aware computations on them. A bilateral grid is a 3D representation of a 2D image that separates pixels not only by their spatial position but also their respective intensity values. Bilateral grids offer a simple yet robust way to preserve and retrieve edges present in the original RGB image after performing computations

Let $I(x, y) = z$ be a gray scale image where x, y are the pixel indices and z is the intensity value, its corresponding bilateral grid is given by

$$BG(x, y, z) = z \quad \forall x, y, z \in I \tag{3.1}$$

Any 2D operation on image space, becomes a 3D operation in the bilateral space. This is the motivation behind the fact that the network we trained is a 3D CNN.

Bilateral Grid for CNN

For high resolution images converting them to High Dimensional Grid can be computationally expensive in conversion and also processing by the CNN. Hence the spatial dimensions are sampled by some factor while conversion, in our case we chose this factor to be 5. The choice of intensity dimension is 32 or 64, we chose 32 after some experimentation.

Using a normalized image from [0,1], f_x and f_y are the spatial sampling rate and c is the range sampling rate, we construct the bilateral grid Γ 0as follows:

- **Initialization** For all grid nodes(i, j, k)

$$\Gamma(i, j, k) = (0, 0) \tag{3.2}$$

- **Filling** For each pixel at position (x, y)

$$\Gamma([x/f_x], [y/fy], [I(x, y)/c]) += (I(x, y), 1) \tag{3.3}$$

  where[·]is the closest-integer function.

Note that we accumulate both the image intensity and the number of pixels into each grid cell using homogeneous coordinates.

3.1.1 Motivation for Grid

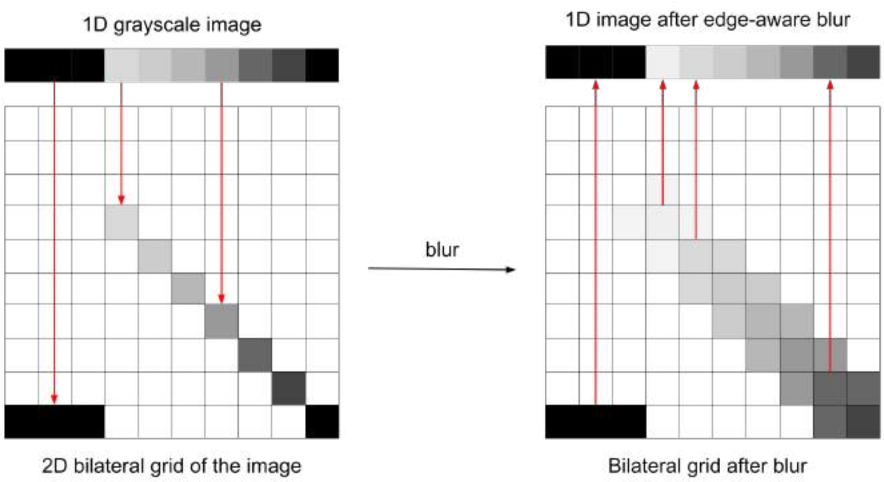

Figure 3.2: How bilateral grids preserve edges

Bilateral Grid based processing in CNN architecture has shown improvement to the existing methods Sharma *et al.* (2021). Bilateral grids have edge preserving properties which make them robust for performing computation on them. Taking Blurring operation as an example, the Gaussian blur kernel does not have edge preserving properties, it blurs out the image evenly. Now the same kernel in 3D (Equivalent 3D version of Gaussian Kernel), since bilateral grid collects image pixels according to the intensity values. The edges which are nothing but a boundary in intensity get separated. Hence the kernel does not affect the edges too much. from the 3.2 we can clearly observe this phenomena.

3.2 Proposed CNN Architecture

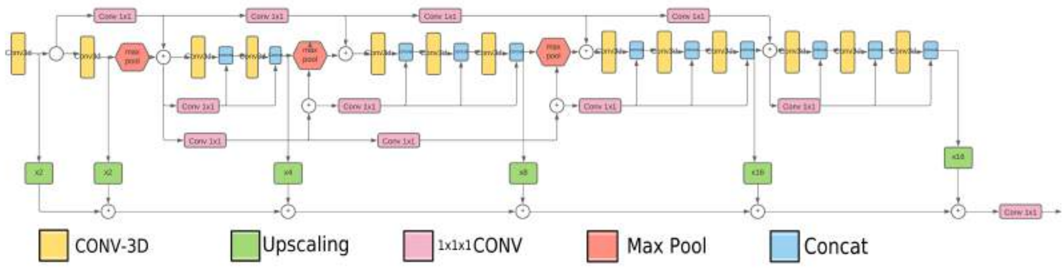

Figure 3.3: Proposed model architecture dubbed BG3DIN-DeepSketch

In this section, we explain our proposed convolutional network architecture, dubbed BG3DIN-DeepSketch Architecture. A block diagram is illustrated in 3.3. The objective of proposed BG3DIN-DeepSketch to strike a balance between quality and runtime complexity. The BG3DIN-DeepSketch built up of 4 major components :

- 6 Repeating Convolution Blocks
- Up-scaling Blocks
- Output Fuse layer
- 1x1x1 Convolution Connections

The model is inspired from Xception blocks Chollet (2017), with a few key differences. Instead of separable 2D convolutions we are using 3D Convolutions , network flow has also been modified at some places and also extra skip connections are added.

The inputs to the network are Bilateral Grids of the input image. Note in the bilateral grid x and y co-ordinates are spatial position and z co-ordinate is the sampled intensity. Further, the proposed BG3DIN-DeepSketch architecture efficiently learns the edge information.

Input Layer - The input image is first mean centered with the following mean values (103.939,116.779,123.68) Deng *et al.* (2009). Then the image is normalized from 0-255 to 0-1 . And then it is converted to Bilateral Grid with is the input to the BG3DIN-DeepSketch model.

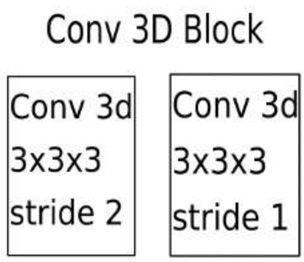

Figure 3.4: Expanded single unit of 3D convolution Block in BG3DIN-DeepSketch

3D Convolution Blocks - The main block architecture has been inspired by Inception Szegedy *et al.* (2015). In Inception 2D convolutions were but here we have standard 3D convolutions so as to accommodate Bilateral Grid. Each block has multiple units of 3-D convolution layers of 3x3 and 1x1 in parallel. The network is modular as in the number of such inception blocks output is taken after each block after up-scaling. Each 3D convolution blocks consist of two 3x3x3 convolutions with first convolution layer having a stride of 2 and second convolution layer with stride 1. shown in 3.4. Number of filters inside each convolution layers were changed as per training experiments.

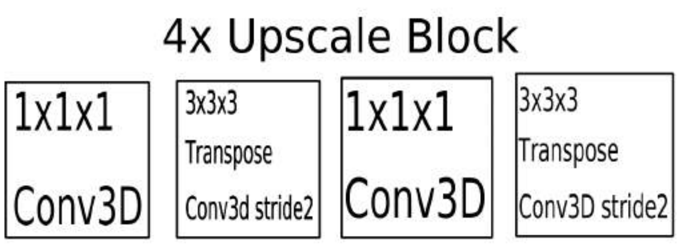

Figure 3.5: Expanded up-scaling unit in the BG3DIN-DeepSketch

Up-Convolution Blocks - Blocks of 3D transpose convolution are used at the output of every inception stage to get the original Bilateral Grid dimension. Each of this block outputs the final bilateral grid in itself. However output is obtained as the average or fused output of all the six blocks. 3.5 shows the block for 4x Up-Convolution unit.

Similar blocks are stacked for other upscalers as well. The number of filters in each layers 16 except the last block.

Output Fuse Layer - The model generates two outputs Fused and average. All the six outputs from the Up-Convolution blocks are averaged to get the output. Another output is obtained by stacking all the six up-convolution block outputs and passing through 1x1x1 convolution layer with stride 1 to get the fused output.

1x1x1 Convolution Connections - Parallel Filters are used in Convolutional Units of kernel size 1x1. Multiple Such 1x1 filters are added to BG3DIN-DeepSketch. Since many convolutions are performed, every block looses important edge features just one main-connection is not sufficient.

CHAPTER 4

Experiments and Results

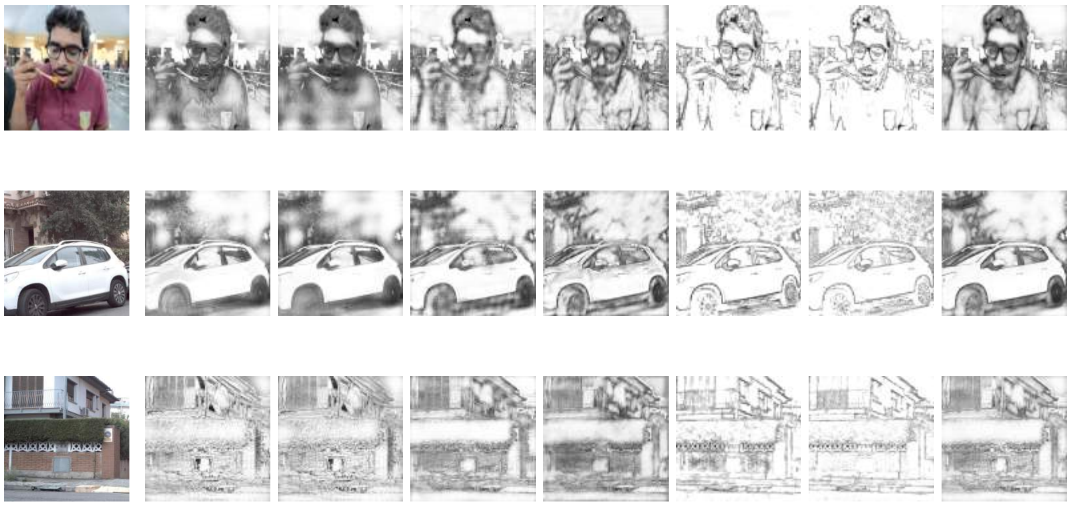

Figure 4.1: From left to Right Input-Image followed by output from each block after upsampling block, and fused output

4.1 Dataset

Barcelona Images for Perceptual Edge Detection (BIPED) BIPED dataset contains 250 outdoor images of 1280×720 pixels each. These images have no redundancy and has been carefully annotated by computer vision experts. This has ensured the quality of the dataset also the ground truth edges have been manually verified to remove wrong edges .This dataset is available publicly for training and evaluating edge detection techniques. For training our sketching model we have used BIPED for the accuracy in its ground truth. For the training purpose from the BIPED dataset, 50 images have been randomly selected for testing and the remainders 200 for training and validation. Since BIPED dataset only has 250 images Data Augmentation was performed to increase training samples.

4.1.1 Augmentation

The following steps were performed for the dataset Augmentation.

- As BIPED images are high resolution they are split up in the half of image width size
- Each of the images after split is rotated by 15 different angles uniformly spaced and cropped to fit
- After rotation the images are horizontally flip
- Two different values of gamma corrections i) 0.3030 and ii) 0.6060 have been applied

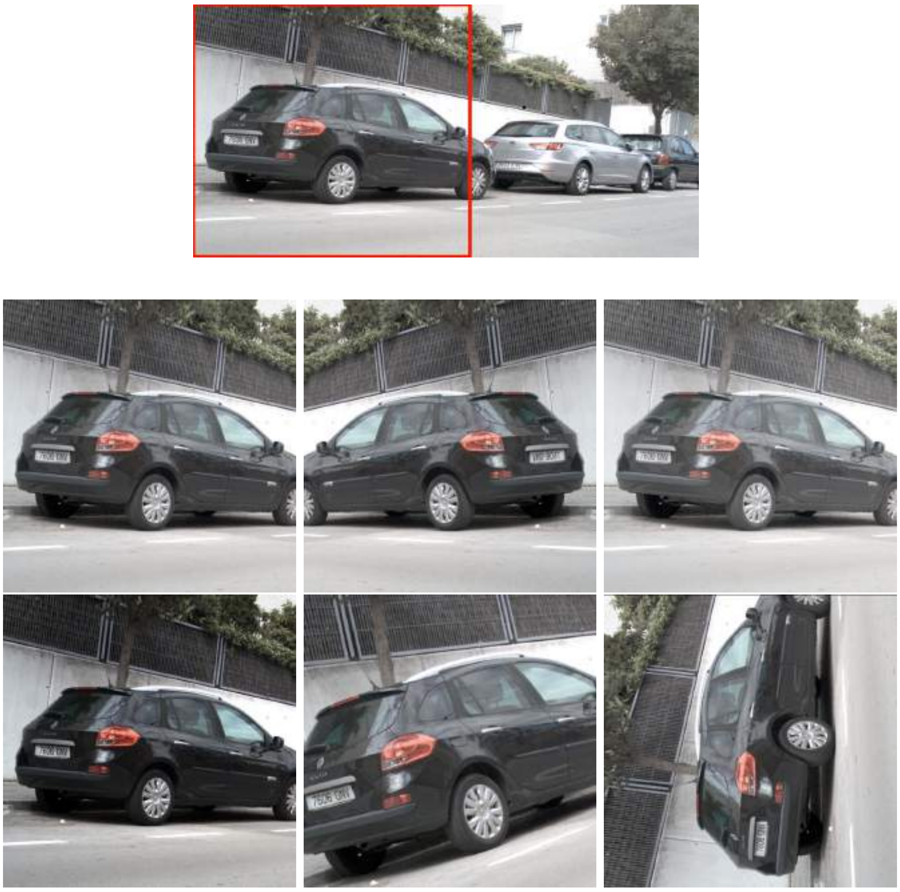

Figure 4.2: (i) Top: Orignal BIPED image with 1280x720 (ii) Bottom : Left to Right ; Cropped , Flipped , Gamma Correction 30 , Gamma Correction 60, Rotation 19 degrees, Rotation 90 degrees

4.2 Implementation Details and Experimental Settings

The model was trained on the hardware with the following specifications Intel i7-9750H CPU, 16 GB RAM, RTX 2080 GPU with 8 GB VRAM. The model was implemented on tensorflow-2 framework and trained with GPU acceleration. For training Adam Optimizer with moment term β= 0.5 was used. Learning rate for training was 0.002 and ran the model for 10 epochs.

For weights initialization in the model, we used Glorot Uniform Initializer Glorot and Bengio (2010). Below is the distribution from which weights are sampled, fan_in and fan_out are the number of input units in the weight tensor and the number of output unit is the weight tensor respectively.

$$w \sim U(-limit, limit), \qquad where\ limit = \sqrt{\frac{6}{(fan\_in + fan\_out)}}$$

During training we also implemented regularization using l2 norm of weights in some layers with weight decay set to 0.001. To improve generalization. Also Batch Normalization is used after each layer.

4.3 Results

4.3.1 Comparative Results

Since the NPR, especially Sketching is a subjective matter. We compare our results with those produced by several representative image based pencil sketching approaches and commercial software. We will compare our results with im2pencil Li *et al.* (2019), Lu *et al.* (2013) and also a commercial photo-editing software Krita kri (2019). For this comparison we have used the fused output from our model.

For comparison im2pencil the output was obtained with clean configuration and cross hatching style. For Lu *et al.* (2013) the output was obtain with default settings.

Looking at the output on under different scenes 4.3 all 4 outputs have their own unique style. Output from im2pencil and Lu *et al.* (2013) have very sharp output and has a very realistic feel to the sketch. Where as output from our network has its own sketching style and resembles more to Cartoon like - comic book sketches.

Details - Output from im2pencil has the highest amount of details present in the sketch followed by Lu *et al.* (2013). However, all the methods in this comparison were able to capture sufficient details in their artistic capacity.

Strokes - In pencil sketching strokes are an important part which adds character to the sketch. Lu *et al.* (2013) enables multiple stroke customization and has overall the best contrast. Ours, BG3DIN-DeepSketch's output also has smooth strokes and descent contrast.

Tone Map/Shading - Im2pencil's model as well as Lu *et al.* (2013) architecture has flexibility in choosing different shading style/ tone. In our model 4.2 output from different blocks provide different level shading providing multiple choices.

Artistic Appeal - It is a matter of individual taste and use case. All the four methods chosen for comparison have their own unique style. And can be used according to the required scenario. Our proposed method gives an alternative option with unique style which can suite multiple scenarios, some of them we will discuss in applications.

Figure 4.3: (i) Top : source image (ii) Bottom: Left to Right Lu *et al.* (2013), Li *et al.* (2019), Krita and BG3DIN-DeepSketch (Ours)

CHAPTER 5

Applications

Stylizing images in an artistic manner has been widely studied in the domain of NPR rendering .Traditional approaches develop dedicated algorithms for specific styles. However, substantial efforts are required to produce fine-grained styles that mimic variety if art styles.

In our work we have shown how scenes can be converted to sketched for the digital artists. There are also other styilization application that can help digital artists. In this work we are discussing three different applications.

5.0.1 Domain transform stylization

This Stylization aims to produce digital imagery with a wide variety of effects not focused on photorealism. Edge-aware filters are ideal for stylization, as they can abstract regions of low-contrast while preserving, or enhancing, high-contrast features. Sketching can be combined with other stylization methods to create new styles. Using recursive filtering method Gastal and Oliveira (2011) with edge aware 1D blur filter (σ_s =100, σ_r=0.45) and superimposing it with sketch from our BG3DIN-DeepSketch. The resulting output looks like an oil painting with outlines as shown in 5.2

5.0.2 Anime like Art stylization

Line art colorization in an important step in artistic work such as the composition of illustration and animation. Similar to previous approaches Chen *et al.* (2018) of Anime style like colourization we would want to take an RGB image as input for the scene and generate a anime/cartoon like colourized sketch. For this application cGAN based colorization model Ci *et al.* (2018) which takes sketch as an input along with colour palette and outputs Colourized output. We want to evaluate colourization results on our sketching model. The first input to the cGAN based colourization system is a grey-scale line art image X $\in 1 \times H \times W$, which is a sparse, binarized image synthesized from our model. The model was run is automatic configurations. The results obtained are shown in 5.4.

5.0.3 Sketch Simplification

Sketch output from our BG3DIN-DeepSketch can be simplified and modified as per artists requirement. For this application FCNN network Simo-Serra *et al.* (2016) which is designed to covert rough hand drawn sketches into simplified version. Using this model on BG3DIN-DeepSketch output images results in simplified sketches , hence providing more versatility in the art style. By down-scaling the input image, it is possible to obtain more simplified sketches and vice-versa. The first input to the sketch simplification model is the sketch from our BG3DIN-DeepSketch, the model supports input of arbitrary resolution. The results obtained are shown in 5.5.

Figure 5.1: Source RGB Image

Figure 5.2: Ours BG3DIN-DeepSketch network output from the final block

Figure 5.3: Domain transfer stylization results

Figure 5.4: Anime-like colourization Results

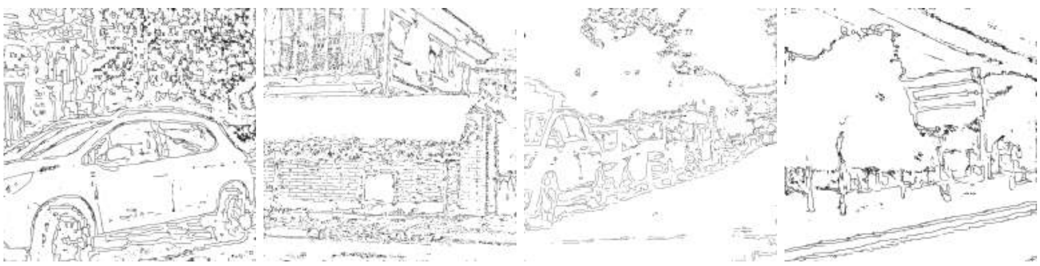

Figure 5.5: Simplified Sketch output

CHAPTER 6

Future Work and Conclusion

6.1 Conclusion

We have presented a novel automated end-to-end system that takes RGB scenes and outputs high quality vectorized sketches. Our model is based on Inception Szegedy *et al.* (2015) based convolution operations on 3D Bilateral Grid for capturing higher geometric information, and is able to handle very challenging scenes images from various sources. We also present several applications alongside that can complement out network providing more versatility for the artists. Our approach is fully automatic and requires no user intervention. Our results show that our approach is able to provide competitive results in sketching despite being able to handle wide range of scenes and textures and also maintains a computation time of 2.7 second for 720x720 image on Nvidia Geforce 940m. We believe our proposed approach is an important step towards being able to integrate sketching into artist's everyday work flow.

6.2 Future Work

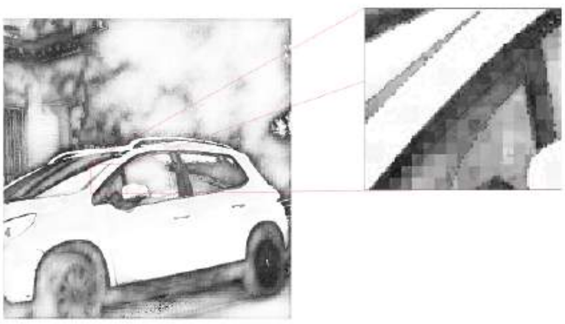

Figure 6.1: Zoomed in result from BG3DIN-DeepSketch

In the future work, due to the importance of portrait, we would like to investigate how to improve stylization for human faces. Although we design our loss functions to tackle general nature of sketching, similar ideas are useful for other image synthesis tasks, which we will investigate further. We also plan to add sequential constraints to the training process to extend our method to handling videos by reducing inference time.

As the output of out network in bilateral grid and the image is obtained after conversion. This results in this subsampling problem and the output looks pixelated and lacks sharpness. In the future there is scope to improve on this problem, as bilateral grid has shown promising outputs in capturing edge details.

REFERENCES

1. (2019). Krita. `https://krita.org/en/`.

2. **Cai, X.** and **B. Song** (2018). Image-based pencil drawing synthesized using convolutional neural network feature maps. *Machine Vision and Applications*, **29**, 503–512.

3. **Cewu Lu, J. J., Li Xu**, Combining sketch and tone for pencil drawing production. *In International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2012), June, 2012*. 2012.

4. **Chen, J.**, **S. Paris**, and **F. Durand** (2007). Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (TOG)*, **26**(3), 103–es.

5. **Chen, Y.**, **Y.-K. Lai**, and **Y.-J. Liu**, Cartoongan: Generative adversarial networks for photo cartoonization. *In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

6. **Chollet, F.** (2017). Xception: Deep learning with depthwise separable convolutions.

7. **Ci, Y.**, **X. Ma**, **Z. Wang**, **H. Li**, and **Z. Luo** (2018). User-guided deep anime line art colorization with conditional adversarial networks. *Proceedings of the 26th ACM international conference on Multimedia*. URL `http://dx.doi.org/10.1145/3240508.3240661`.

8. **Deng, J.**, **W. Dong**, **R. Socher**, **L.-J. Li**, **K. Li**, and **L. Fei-Fei**, Imagenet: A large-scale hierarchical image database. *In 2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009.

9. **Gastal, E. S. L.** and **M. M. Oliveira** (2011). Domain transform for edge-aware image and video processing. *ACM TOG*, **30**(4), 69:1–69:12. Proceedings of SIGGRAPH 2011.

10. **Glorot, X.** and **Y. Bengio**, Understanding the difficulty of training deep feedforward neural networks. *In* **Y. W. Teh** and **M. Titterington** (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*. PMLR, Chia Laguna Resort, Sardinia, Italy, 2010. URL `http://proceedings.mlr.press/v9/glorot10a.html`.

11. **Li, Y.**, **C. Fang**, **A. Hertzmann**, **E. Shechtman**, and **M.-H. Yang** (2019). Im2pencil: Controllable pencil illustration from photographs.

12. **Lu, C.**, **L. Xu**, and **J. Jia** (2013). Combining sketch and tone for pencil drawing production.

13. **Sharma, M.**, **A. Sharma**, **K. R. Tushar**, and **A. Panneer** (2021). A novel 3d-unet deep learning framework based on high-dimensional bilateral grid for edge consistent single image depth estimation.

14. **Simo-Serra, E.**, **S. Iizuka**, **K. Sasaki**, and **H. Ishikawa** (2016). Learning to simplify: Fully convolutional networks for rough sketch cleanup. *ACM Trans. Graph.*, **35**(4). ISSN 0730-0301. URL `https://doi.org/10.1145/2897824.2925972`.

15. **Szegedy, C.**, **V. Vanhoucke**, **S. Ioffe**, **J. Shlens**, and **Z. Wojna** (2015). Rethinking the inception architecture for computer vision.

16. **Wang, N.**, **S. Zhang**, **X. Gao**, **J. Li**, **B. Song**, and **Z. Li** (2017). Unified framework for face sketch synthesis. *Signal Process.*, **130**(C), 1–11. ISSN 0165-1684. URL `https://doi.org/10.1016/j.sigpro.2016.06.014`.

17. **Winnemöller, H.**, **J. Kyprianidis**, and **S. Olsen** (2012). Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Comput. Graph.*, **36**, 740–753.

18. **Xu, L.**, **C. Lu**, **Y. Xu**, and **J. Jia** (2011). Image smoothing via l0 gradient minimization. *ACM Trans. Graph.*, **30**(6), 1–12. ISSN 0730-0301. URL `https://doi.org/10.1145/2070781.2024208`.

19. **Zhang, J.**, **R.-Z. Wang**, and **D. Xu**, Automatic genaration of sketch-like pencil drawing from image. *In 2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*. 2017.