

# **PENCIL IMAGES TRANSLATION INTO REALISTIC IMAGES USING GAN**

**Raigam Koralege Binuri Yeshani**

**199378T**

**M.Sc. in Computer Science**

**Department of Computer Science & Engineering  
Faculty of Engineering**

**University of Moratuwa  
Sri Lanka**

**April 2024**

# **PENCIL IMAGES TRANSLATION INTO REALISTIC IMAGES USING GAN**

**Raigam Koralege Binuri Yeshani**

**199378T**

**Thesis submitted in partial fulfillment of the requirements for the  
degree of Master of Science.**

**Department of Computer Science & Engineering  
Faculty of Engineering**

**University of Moratuwa  
Sri Lanka**

**April 2024**

## DECLARATION

I declare that this is my own work and this Dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 2024-04-28

The supervisor should certify the Dissertation with the following declaration. The above candidate has carried out research for the Name of the Degree Dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Charith Chitranjan

Signature of the Supervisor:

Date:

## **ACKNOWLEDGEMENT**

I wish to extend my deepest appreciation to my supervisor, Dr. Charith Chitraranjan for guidance, support, and encouragement throughout my research. And I extend my thanks to all staff members of the Computer Science department for their assistance and cooperation. I am thankful to my family for their love, support, and understanding throughout this journey. Lastly, I would like to thank all my friends and colleagues who have supported me in various ways, contributing to the successful completion of this thesis.

## ABSTRACT

This thesis presents an innovative approach to image-to-image translation, focusing on the conversion of colour pencil art, black and white pencil images, and grayscale images into RGB images using the Pix2Pix Generative Adversarial Network (GAN). Building upon the established Pix2Pix framework, which utilises a UNet-based architecture with skip connections for the generator and a PatchGAN discriminator, this research introduces a novel modification by incorporating an intermediate layer into the Unet architecture. This modification aims to refine the translation process, addressing the intricate details and colour dynamics of the input images. Empirical results demonstrate that this enhanced generator architecture significantly outperforms the traditional UNet architecture in producing more vivid, detailed, and colour accurate RGB images. By leveraging this advanced method, the study not only contributes to the body of knowledge in neural network applications for artistic image translation but also opens new avenues for further research in high-fidelity image conversion technologies. This work underscores the potential of targeted architectural modifications within GAN frameworks to achieve superior image translation outcomes, thereby setting a precedent for future explorations in the field.

**Keywords:** Generative adversarial networks, Conditional GAN, Pix2pix GAN, Image-to-image translation.

# Contents

Declaration . . . . .	iii
Acknowledgement . . . . .	iv
Abstract . . . . .	v
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation . . . . .	1
1.3 Objectives of the Study . . . . .	2
1.4 Structure of the Thesis . . . . .	2
<b>2 Background</b>	<b>3</b>
2.1 Deep Neural Networks . . . . .	3
2.1.1 Convolutional Neural Networks (CNN) . . . . .	3
2.2 Generative Adversarial Networks(GAN) . . . . .	3
2.2.1 Generator . . . . .	3
2.2.2 Discriminator . . . . .	4
2.2.3 Latent Space . . . . .	4
2.2.4 GAN network evaluation and Loss function . . . . .	4
2.3 Encoder-Decoder Architecture . . . . .	5
2.3.1 U-Net . . . . .	5
2.4 GAN Architectures and Variants . . . . .	5
2.4.1 Vanila GAN . . . . .	5
2.4.2 Conditional GAN . . . . .	6
2.4.3 Deep Conditional GAN (DCGAN) . . . . .	6
2.4.4 Wasserstein GAN . . . . .	7
2.4.5 Pix2Pix GAN . . . . .	7
2.4.6 Cycle GAN . . . . .	7
2.4.7 Style GAN . . . . .	8
<b>3 Literature Review</b>	<b>9</b>
3.1 Background . . . . .	9
3.2 Grayscale Image Colorization . . . . .	9
3.3 Sketch Image Colorization . . . . .	10
3.4 Infra-red Image Colorization . . . . .	10
3.5 Domain Specific Image Colorization . . . . .	10

3.6	Text to Image Synthesis . . . . .	11
3.7	Image to Image Translation . . . . .	11
<b>4</b>	<b>Methodology and Implementation</b>	<b>12</b>
4.1	Introduction . . . . .	12
4.2	Datasets . . . . .	12
4.2.1	Synthesise pencil drawings using real image . . . . .	12
4.3	Proposed GAN Architectures . . . . .	14
4.3.1	Paired Image to Image translation . . . . .	14
4.4	Model Evaluation . . . . .	14
4.5	Experimental Setup . . . . .	15
4.6	Programming languages and Computational setup . . . . .	15
<b>5</b>	<b>Results</b>	<b>16</b>
5.1	Pix2Pix GAN model training with Paired images . . . . .	16
5.1.1	Color pencil image translation into RGB images - Training phase. . . . .	16
5.1.2	Grayscale Pencil image to RGB image - Training phase . . . . .	19
5.1.3	Grayscale image to RGB - Training phase . . . . .	21
5.2	Testing results – for synthesized data . . . . .	24
5.2.1	Oxford 102 flowers dataset . . . . .	24
5.2.2	CUB-102-2011 birds dataset . . . . .	25
5.3	Testing results – for actual data . . . . .	27
5.3.1	Color pencil image to RGB . . . . .	27
5.3.2	Grayscale pencil image to RGB . . . . .	28
<b>6</b>	<b>Discussion and Conclusion</b>	<b>29</b>
<b>7</b>	<b>Future Works</b>	<b>30</b>
	<b>Bibliography</b>	<b>32</b>

## List of Figures

2.1	GAN training architecture. (image source Goodfellow’s paper) . . . . .	4
2.2	U-Net model architecture of encoding and decoding layers) . . . . .	5
2.3	The structure of the Conditional-GAN . . . . .	6
2.4	Paired data( $Y_i$ image which is maps with $x_i$ image) and unpaired data(no target existing $x_i$ image maps in $y$ domain) . . . . .	7
4.1	Hatching Patterns I, II, III, IV, V are from leftmost to rightmost respectively. . . . .	12
4.2	Synthesized pencil drawing and color pencil drawings . . . . .	13
4.3	There is main two steps b) and c); b) the stroke layer c) tonal texture and d) is the final output a combination of b and c. . . . .	13
4.4	The framework is to translate RGB image into color/grayscale pencil image(1). . . . .	13
4.5	Translated images are presented for CUB-200-2011 dataset. RGB image and synthesized color pencil image are presented respectively according to the paper (1) proposed. . . . .	13
4.6	Translated images are presented for Oxford 102 flowers dataset. RGB image and synthesized color pencil image are presented respectively according to the paper (1) proposed. . . . .	14
5.1	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are presented from left to right respectively. . . . .	16
5.2	Generator and discriminator loss at each epoch ( $lr=0.0002$ , epochs 1500)	17
5.3	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are presented from left to right respectively . . . . .	17
5.4	Generator and discriminator loss at each epoch ( $lr=0.0002$ , epochs 1500)	17
5.5	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	18
5.6	Generator and discriminator loss at each epoch ( $lr=0.0002$ , epochs 2000)	18

5.7	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	18
5.8	Generator and discriminator loss at each epoch (lr=0.0002, epochs 2000)	19
5.9	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are presented from left to right respectively . . . . .	19
5.10	Generator and discriminator loss at each epoch (lr=0.0002, epochs 1500)	19
5.11	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are from left to right respectively . . .	20
5.12	Generator and discriminator loss at each epoch (lr=0.0002, epochs 1500)	20
5.13	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	20
5.14	Generator and discriminator loss at each epoch (lr=0.0002, epochs 2000)	21
5.15	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	21
5.16	Generator and discriminator loss at each epoch (lr=0.0002, epochs 2000)	21
5.17	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are presented from left to right respectively. . . . .	22
5.18	Generator and discriminator loss at each epoch (lr=0.0002, epochs 1500)	22
5.19	Real image, color pencil input image, generated image at 700 epochs, generated image at 1500 epochs are presented from left to right respectively. . . . .	22
5.20	Generator and discriminator loss at each epoch (lr=0.0002, epochs 1500)	23
5.21	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	23
5.22	Generator and discriminator loss at each epoch (lr=0.0002, epochs 2000)	23
5.23	Real image, color pencil input image, generated image at 1000 epochs, generated image at 2000 epochs are presented from left to right respectively. . . . .	24
5.24	Generator and discriminator loss at each epoch (lr=0.0002, epochs 2000)	24
5.25	Input image, output-U-Net without intermediate layer, output-U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	24
5.26	Input image, output-U-Net without intermediate layer, output-U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	25

5.27	Input image, output–U-Net without intermediate layer, output–U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	25
5.28	Input image, output–U-Net without intermediate layer, output–U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	25
5.29	Input image, output–U-Net without intermediate layer, output–U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	26
5.30	Input image, output–U-Net without intermediate layer, output–U-Net with intermediate layer, Actual images are presented from left to right respectively . . . . .	26
5.31	Original pencil image ( <i>image source</i> ), generated image-U-Net with immediate layer and generated image-U-Net without intermediate layer are presented left to right respectively. . . . .	27
5.32	Original Pencil Image ( <i>image source</i> ), generated image-U-Net with immediate layer and generated image-U-Net without intermediate layer are presented left to right respectively . . . . .	28
5.33	Original pencil image ( <i>image source</i> ), generated image-U-Net with immediate layer and generated image-U-Net without intermediate layer are presented left to right respectively. . . . .	28
5.34	Original Pencil Image ( <i>image source</i> ), generated image-U-Net with immediate layer and generated image-U-Net without intermediate layer are presented left to right respectively . . . . .	28

## List of Tables

5.1	Comparison of IS and FID scores for different U-Net variants on CUB-201-2011 and Oxford-102 datasets. . . . .	27
-----	---------------------------------------------------------------------------------------------------------------	----