Jamdani Motif Generation using Conditional Generative Adversarial Networks

MD Tanvir Rouf Shawon	160104138
Raihan Tanvir	160104140
Susmoy Kar	160104141
Humaira Ferdous Shifa	160104144

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

December 2020

Jamdani Motif Generation using Conditional Generative Adversarial Networks

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

MD Tanvir Rouf Shawon	**160104138**
Raihan Tanvir	**160104140**
Susmoy Kar	**160104141**
Humaira Ferdous Shifa	**160104144**

Supervised by

Mohammad Imrul Jubair
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

December 2020

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mohammad Imrul Jubair, Assistant Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

MD Tanvir Rouf Shawon
160104138

Raihan Tanvir
160104140

Susmoy Kar
160104141

Humaira Ferdous Shifa
160104144

CERTIFICATION

This thesis titled, **"Jamdani Motif Generation using Conditional Generative Adversarial Networks"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2020.

Group Members:

MD Tanvir Rouf Shawon	**160104138**
Raihan Tanvir	**160104140**
Susmoy Kar	**160104141**
Humaira Ferdous Shifa	**160104144**

Mohammad Imrul Jubair
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Professor Dr. Kazi A. Kalpoma
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

First and foremost, we are grateful to Almighty Allah for blessing us with good health and well being that we were able to complete our research on time successfully.

Next, we would like to express our sincere gratitude to our beloved supervisor, Mohammad Imrul Jubair, Assistant Professor, Dept of CSE, AUST, for his patience, motivation, and immense knowledge. It was his guidance and constant support that inspired us to take up our research from the scratch and kept us going.

Alongside our supervisor, We would like to take this opportunity to thank Dr. Kazi A. Kalpoma, Professor and Head, Dept. of CSE, AUST and the faculty members and staffs of the Department of CSE, AUST, who believed in us with their support and encouragement.

Our heartiest thanks to the honorable external examiners: Mr. Faisal Muhammad Shah, Dr. Nusrat Sharmin, Mr. Junaed Younus Khan and Mr. Tonmoy Hossain who patiently scrutinized our work and offered thoughtful suggestions to improve and extend our research in the long run.

We can not but thank Mr. Mir Tafseer Nayeem, Assistant Professor, Dept of CSE, AUST, for his kind mentorship throughout our thesis period.

Our earnest regard to Mr. Juned Ahmad Muhtaseem from National Craft Council Bangladesh and Mr. Shojib, owner of Shahina Jamdani Weaving Factory for lending us a helping hand in preparing the required data set for our experiment and enlightening us with the century old knowledge of Jamdani.

Last but not least, we want to thank our parents who made us come this far by always being there for us with their unfailing love and blessing.

Dhaka
December 2020

MD Tanvir Rouf Shawon

Raihan Tanvir

Susmoy Kar

Humaira Ferdous Shifa

ABSTRACT

Jamdani is the strikingly patterned textile heritage of Bangladesh. The exclusive geometric motifs woven on the fabric are the most attractive part of this craftsmanship having a remarkable influence on textile and fine art of our land. But the glory of Jamdani has faced a few stumbling blocks that has threaten its existence. In our research, we have developed a technique based on Generative Adversarial Network that can learn to generate entirely new Jamdani patterns from a collection of Jamdani motifs that we assembled, the newly formed motifs can mimic the appearance of the original designs. Whether a user can select a preset from our system or draw the skeleton of the desired motif by hand, our system will generate a completely new motif based on that input having the attributes of the original ones. The potential of Generative Adversarial Network is huge as it can learn to mimic any distribution of data. We want our generator to learn the different aspects of the motifs without human supervision. The use of this technology will ensure the creation of new plausible motifs on demand. To serve this purpose, we collected and pre-processed a dataset containing a large number of Jamdani motifs images from authentic sources via fieldwork and applied a state-of-the-art method called pix2pix on it. To the best of our knowledge, this dataset is currently the only available dataset of Jamdani motifs in digital format for computer vision research. A model like this will help us digitize this sector in an effective and efficient way and preserve the endangered heritage. The genuine traits of the motifs can be used without any alteration using this neural network. Our experimental results of the pix2pix model on this dataset show satisfactory outputs of computer-generated images of Jamdani motifs. Though to produce picture perfect output more data is needed but nevertheless the outputs produced with the newly built dataset possesses remarkable attributes already. It is important to introduce restructuring ideas to the dying Jamdani industry in order to improve the current scenario and to define new dimension of possibilities of transformation. The loss of numerous spectacular motifs cannot be repaired but it is high time we take action to protect the surviving portion of this master craftsmanship. This research of ours is the entrance of the bridging point connecting latest Computer Vision model with the age old textile marvel. We intend to revolutionize this industry with the blessing of technology and creativity.

Contents

CANDIDATES' DECLARATION i

CERTIFICATION ii

ACKNOWLEDGEMENT iii

ABSTRACT iv

List of Figures viii

List of Tables xii

1 Introduction 1

1.1 Research Domain 1
1.2 Naming Convention of Motifs 2
1.3 Research Motivation 2
1.4 Upbeat VS Downbeat: Illustration of Current Scenario 3
1.5 Anticipated Benefits Brought by the Proposed System 4
1.6 Our Contribution 5
1.6.1 Development of an Unique Dataset 5
1.6.2 Approach Towards Building the Proposed System 5

2 Background Studies & Related Works 7

2.1 Motivational Works 7
2.1.1 Image-to-Image Translation with Conditional Adversarial Networks (*pix2pix* GAN) 8
2.1.2 A Style-Based Generator Architecture for Generative Adversarial Networks 8
2.1.3 CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks 10
2.1.4 PixelDTGAN-Pixel-Level Domain Transfer 10
2.1.5 Enhanced Super-Resolution Generative Adversarial Networks (ES-RGAN) 12

2.1.6 LUCSS: Language-based User-customized Colourization of Scene Sketches . . . 12
2.1.7 Sketch-to-Image Generation Using Deep Contextual Completion . . 13
2.1.8 Stroke of Genius: GauGAN Turns Doodles into Stunning, Photorealistic Landscapes . . . 14
2.2 Related Work . . . 14
2.2.1 Hand-loom Design Generation using Deep Neural Networks . . . 14

3 Data Collection & Processing **17**
3.1 Our Dataset: **The Jamdani Noksha** . . . 17
3.1.1 Data Authenticity . . . 17
3.1.2 Data Collection . . . 18
3.1.3 Data Processing . . . 21
3.1.3.1 Skeleton Version . . . 22
3.1.3.2 Reduced Branch Version . . . 26
3.1.3.3 Sketch Version . . . 27
3.1.3.4 Boundary Version . . . 28
3.1.3.5 Enhanced Resolution Version . . . 29

4 Methodology **32**
4.1 The Core : Generative Adversarial Networks . . . 32
4.2 The Adaption : Image-to-Image Translation with Conditional Adversarial Network (Pix2Pix) . . . 34
4.2.1 Objective Function . . . 34
4.3 Network Architectures . . . 35
4.3.1 U-Net Generator . . . 35
4.3.2 PatchGAN Discriminator . . . 36
4.3.3 Optimization and Inference . . . 37
4.3.3.1 Generator Loss Function . . . 38
4.3.3.2 Discriminator/Adversarial Loss Function . . . 40

5 Experiments and Results **41**
5.1 Experimental Setup . . . 41
5.2 Experimental Result . . . 42
5.2.1 Experimenting with Deep Convolutional GAN . . . 42
5.2.2 Training with *Boundary* and Output . . . 42
5.2.3 Training with *Enhanced Resolution* and Output . . . 43
5.2.4 Training with *Reduced Branch* and Output . . . 45
5.2.5 Training with *Skeleton* and Output . . . 46
5.2.6 Training with *Sketch* and Output . . . 47

5.2.7 Loss Graph Analysis 48
5.3 Web Application using the Generative-Jamdani Engine 49
5.4 Evaluation 51
5.4.1 Human Evaluation 51
5.4.1.1 Response Collection 51
5.4.1.2 Result Analysis 55

6 Future Work and Conclusion **56**
6.1 Limitations And Future Work 56
6.2 Conclusion 58

References **59**

List of Figures

1.1 A traditional Jamdani Saree having different motifs on it (*left*) and a half weaved Jamdani Saree on the loom (*right*) 2

1.2 Various Jamdani Paar. *Source:* [1] 3

1.3 Samples of old (*left*) and contemporary (*right*) Jamdani designs 3

1.4 Motifs called by the same name but with different appearances 4

1.5 Some output of our work: Generating Jamdani motif (*right*) from sketch (*left*). 5

1.6 Process model for our thesis. We provide a sketch of the motif (256×256) for the image to image translation with conditional GAN model. Finally the resultant image (256×256) which is a generated Jamdani motif is measured with human evaluation for accuracy. 6

2.1 *pix2pix* GAN Translation from Sketch to Bag (*top*) and Sketch to Shoe (*bottom*) . Input at (*left*) ground truth at (*middle*) and the output at (*right*) [2] . 9

2.2 Destination faces are given row wise and the source faces are given column wise. Output is given at the cross points of the source and destination images. 10

2.3 Monet paintings to real scenery (*Upper left*), real scenery to Monet painting (*lower left*) and Summer to winter (*Upper right*) and Winter to summer (*lower right*) describing *CycleGAN* [3] 11

2.4 In each group a model is wearing a cloth (*left*) and the suggested cloth is shown separately (*right*) [4] 11

2.5 In each group the image which is to be enhanced is at *left* and the gradually enhanced versions are given side by side. [5] 12

2.6 An example of sketch with different colors according to the sentences given below the individual images [6] 13

2.7 Input sketch at the left and output for CE [7], *pix2pix* [2] and Deep Contextual Completion [8] are given one by one for each group. 13

2.8 The given segmentation at the left and the output for GauGAN [9] at the right. 14

2.9 Examples of Normal sarees (*left*) and the Handloom sarees (*right*) from the paper [10] 15

2.10 Samples of Saree → Handloom using CycleGAN [3] (*left*) and DiscoGAN [11] (*right*) version of their own dataset. In each group from left to right, the original saree image, translated image and the reconstructed image are shown respectively 16

3.1 Our team at Jamdani Festival, 2019 [12], Bengal Shilpalay, Dhaka (*top*) and some photos of Jamdani Festival (*bottom*) 18
3.2 Our team members while visiting the weaving factory, Rupganj (*top*) and Data collection from Shahina Jamdani Weaving Factory, Jamdani Polli (*bottom*). . 19
3.3 Sample design collected from sarees and panjabees while visiting the weaving factory, Rupganj 20
3.4 Data sources book: "Traditional Jamdani Designs" by National Craft Council Bangladesh [1] (*top*), Some Jamdani designs collected from Traditional Jamdani Book [1] (*bottom*) 20
3.5 An example of cropping the motifs from a complete design. The black square boxes indicate the motifs which are cropped out for our dataset. 21
3.6 (a) Data format for storing samples in our dataset. Processed (A) and original version (B) are kept sidewise. (b) Edge to Bag data sample (c)Edge to Shoe data sample 22
3.7 In each group *Left to right:* A jamdani motif image at the left and result of binarizing at the middle and the result of binarizing after applying closing operation at the right. 23
3.8 In each group *Left to right:* A jamdani motif image at the left and result of binarizing at the middle and the result of binarizing after applying opening operation at the right. 23
3.9 In each group *Left to right:* A jamdani motif image at the left and result of binarizing after applying different morphological image processing technique and cropping operations at the right. 24
3.10 For each group multiplication of original image (*upper left*) with the binary image (*upper right*) and the result (*below*) 24
3.11 Skeleton (*right*) of a motif in the image (*left*) for each group 24
3.12 Examples of merging the skeleton (*left*) and the motif (*right*) for each data . 25
3.13 Complete flow of processing for preparing a data 25
3.14 Complete flow of processing for preparing a data 26
3.15 Flow of processing for reducing the small boundary branches of a skeleton . 27
3.16 Before (*left*) and after (*right*) reduction of the branch for both (*top*) and (*bottom*). In each pair from left to right, a user given stroke and the corresponding motif is shown respectively 28

3.17 Samples of hand-drawn (*left*) and skeleton (*right*) version of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown respectively 28
3.18 Samples of the **Boundary** version of our dataset. In each pair from left to right, a user given boundary and the corresponding motif are shown. 29
3.19 Samples of poor resolution data of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown. 30
3.20 Samples of Enhanced resolution data of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown. 30
3.21 A visual comparison of the different versions of our *Jamdani Noksha* dataset for a particular motif. In each pair from left to right, a user given boundary and the corresponding motif are shown. 31

4.1 High level representation of generative adversarial networks. 32
4.2 Initially, the generator generates an image from random noise that the discriminator detects as fake. After some iterations,the generator produces an image that somewhat looks real, yet the discriminator is not fooled. After training for enough iterations the generator becomes competent to generate an image that is good enough to fool the discriminator. 33
4.3 Training a conditional GAN to map $skeletons \rightarrow motifs$ 34
4.4 Two choices for the architecture of the generator. The "U-Net" is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks. 36
4.5 Generator loss function architecture 38
4.6 Discriminator loss function architecture 39

5.1 Sample output produced by DCGAN [13]. Since it's output is not conditioned based on the input, hence no input sketch is provided. From the outputs it is evident that DCGAN failed to produce high resolution images on the Jamdani domain. 42
5.2 Loss graph for training on Boundary 43
5.3 Sample output (*middle column*) for model trained on *Jamdani Noksha*'s Boundary Dataset compared to ground truth (*right column*). *Left column* shows input strokes from user. 43
5.4 Loss graph for training on Enhanced Resolution 44
5.5 Sample output (*middle column*) for model trained on *Jamdani Noksha*'s enhanced version, compared to ground truth (*right column*). *Left column* shows input strokes from user. 44
5.6 Loss graph for training on Reduced Branch 45

5.7 Sample output (*middle column*) for model trained on *Jamdani Noksha*'s reduced branch version, compared to ground truth (*right column*). *Left column* shows input strokes from user. 46
5.8 Loss graph for training on Skeleton version 46
5.9 Sample output (*middle column*) for model trained on *Jamdani Noksha*'s skeleton version, compared to ground truth (*right column*). *Left column* shows input strokes from user. 47
5.10 Loss graph for training on Sketch version 47
5.11 Sample output (*middle column*) for model trained on *Jamdani Noksha*'s hand-drawn sketch version, compared to ground truth (*right column*). *Left column* shows input strokes from user. 48
5.12 Comparison of sample output produced by model trained on different version of dataset. 49
5.13 Sequential snapshots of the web application. The **Preset Sketch** button loads a random sketch from the library, **Clear** button clears the canvas and the **Convert** button generates Jamdani Motif from the input sketch. 50
5.14 A snapshot of the form used to collect the response for human evaluation. . . 52
5.15 Pie graph describing the male female ratio of the respondents generated on the google form summary. 53
5.16 Samples randomly selected from the output of the model trained on *Skeleton* version of our dataset for evaluation. In each pair from left to right, the output generated by the model and the corresponding real motif is shown respectively. 53
5.17 Samples randomly selected from the output of the model trained on *Sketch* version of our dataset for evaluation. In each pair from left to right, the output generated by the model and the corresponding real motif is shown respectively. 54

6.1 Jamdani patterns on different products. 58

List of Tables

3.1 Sizes of different versions of *Jamdani Noksha* dataset 30

5.1 Loss for model trained on different version of *Jamdani Noksha* dataset 45
5.2 Average score of each individual samples of fig 5.16. The score is rated on 10.0 54
5.3 Average score of each individual samples of fig 5.17. The score is rated on 10.0 54

Chapter 1

Introduction

Jamdani is a unique century-old handloom creation embedded in Bangladeshi history having a great socio-economic impact. It is the only surviving fine cotton of the 28 varieties of Muslin. This handloom piece of cotton—originated from Bengal and Pundra—has a much older history since the third century BC which can be found in the scripture written by ancient philosopher Chanakya. In 2013, UNESCO inscribed Bangladesh's Jamdani in the Representative List of Intangible Cultural Heritage of Humanity [14].

1.1 Research Domain

The hallmark that makes Jamdani an exceptional piece of art is the ornamental motifs; these distinctive opaque forms are consistently woven in geometrical order which creates a poetic play of light and shadow on the sheer pellucid background, with great mastery and craftsmanship. The weaving of Jamdani is an aesthetic process of tedious, labor-intensive, and time-consuming work that is created by artisans of exclusive skills. The geometric patterns are not sketched on the fabric but created from the imagination of the artisans directly on the loom through a mathematical interlacing of warp and weft the inspired by the flora and fauna of Bangladesh.

The same framework, however, does not apply to old photo restoration and the reason is three-fold. First, the degradation process of old photos is rather complex, and there exists no degradation model that can realistically render the old photo artifact [15]

[1]. In figure 1.1 we can see a traditional Jamdani Saree. Again figure 1.2 shows some Jamdani motifs.

Figure 1.1: A traditional Jamdani Saree having different motifs on it (*left*) and a half weaved Jamdani Saree on the loom (*right*)

1.2 Naming Convention of Motifs

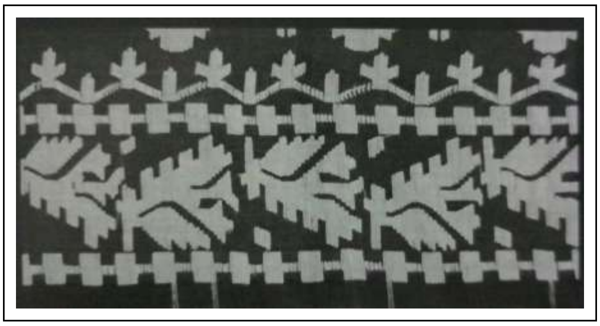
Jamdani motifs are the geometric exposition of nature. Each motif has its own distinct name according to the element which inspired the creation of that motif. There are approximately 7 varients of Jamdani motifs that has survived till this day. They are know as: Boarder Design, Buti Design, Techri Design, Jaal design, Chita Design, Aanchal Design and Haisa Design. These 7 varients are also divided into numerous subcategories. A visual comparison between the old and contemporary Jamdani designs are shown in figure 1.3

There are also motifs inspired from the same source called by the same name but having a different appearances. This is because different artisans interpreted one single element in their own imaginative way. Some motifs having the same name but different appearance are shown here in 1.4

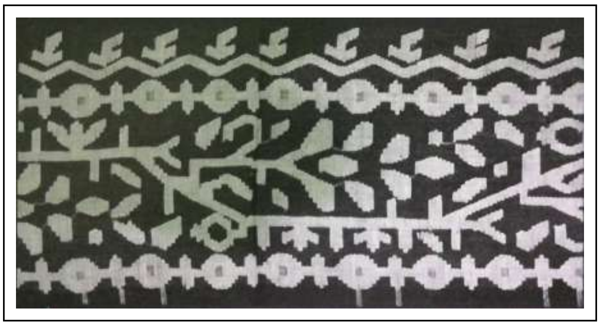
1.3 Research Motivation

Once flourishing Jamdani industry faced periods of stagnation. The negligence towards this cultural heritage created a butterfly effect that brought this industry on the verge of extinction. For many centuries the Jamdani weavers grappled the handicraft against all odd making the Jamdani industry stand still till today.

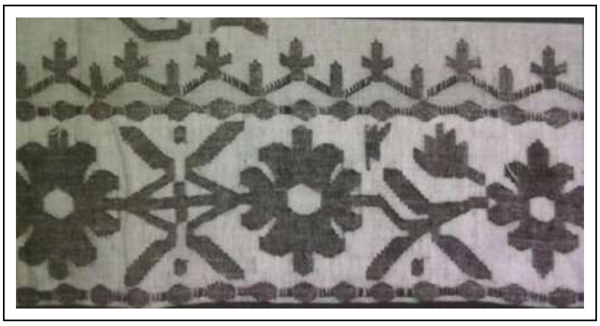
So in order to understand what made us interested to choose this particular field for our research, we will need to point out the facts responsible for the declining the prospects of this industry and the scopes our innovation will create to eliminate them.



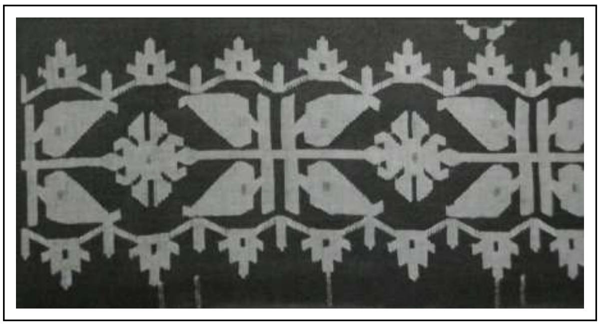
(a) Indur Paar



(b) Gach Paar

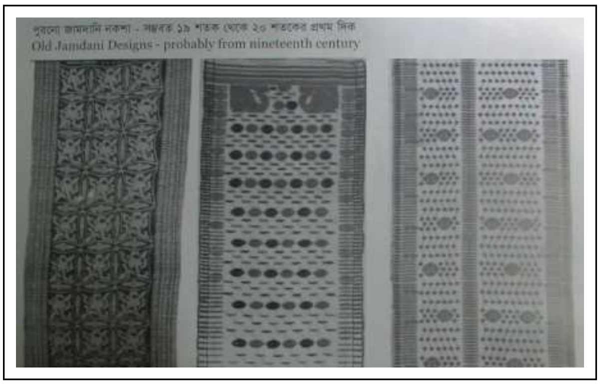


(c) Kauwa-thang Paar



(d) Belpata Paar

Figure 1.2: Various Jamdani Paar. *Source:* [1]



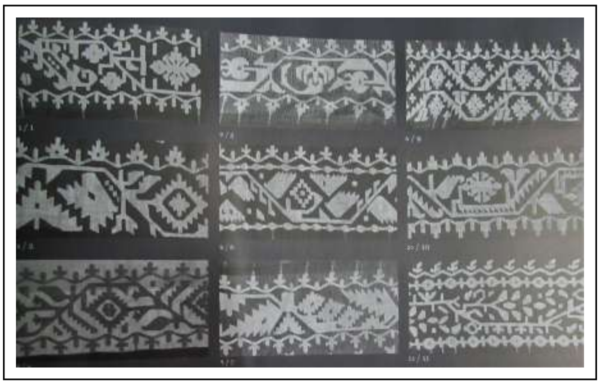
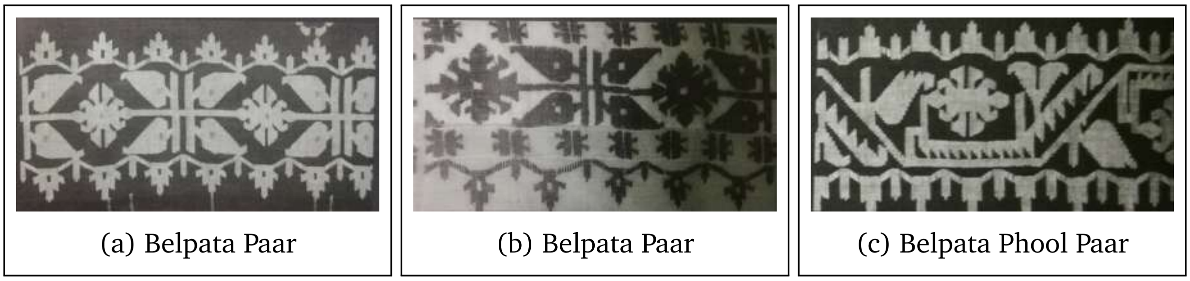


Figure 1.3: Samples of old (*left*) and contemporary (*right*) Jamdani designs

1.4 Upbeat VS Downbeat: Illustration of Current Scenario

The instability Jamdani industry went through and the new scopes arising in the demand of Jamdani motifs are described below:

- In the past, a Jamdani weaver had to serve both as an artisan and a weaver. The designs were at that time the creation of an artisan's mind which were weaved directly on the loom. There is no alternative of perseverance, concentration and mathematical calculation while weaving each motif. With time there has been change in the dedication to work, love for the craft and urge to continue the traditional profession. For ages weavers were not properly rewarded for their efforts which made it hard for



(a) Belpata Paar (b) Belpata Paar (c) Belpata Phool Paar

Figure 1.4: Motifs called by the same name but with different appearances

them to sustain. The discouraging payment and the struggle of artisans with poverty. This made artisans shift from this profession while on the other hand many master artisans passed away. This lead the industry face a critical phase losing significant amount of motifs (many complex ones) and designs in the mists of time.

- Even to these days, each weaver is paid a sum of BDT 14 per hour for working. Thus they are bound to work 12 to 14 hours per day to make an income of around BDT 100 only which is very demotivating for them. Though in recent days measures are being taken to revive this industry from the verge of extinction, the motifs which got lost in time can never be restored.

- In modern times the motifs are no longer confined in textile only. Nationally and internationally the motifs are being made into splendid materials such as jewelries, home decor, curtains, utensils etc. On the other hand there is no Jamdani Artisan alive now who are capable of weaving the motifs directly on the loom from imagination. The weavers now knit the fabric by following designs from catalogs provided by fashion designers. So with the expanding use of motifs, the motif creation is no longer confined in the hands of weavers only. Artists, fashion designers, entrepreneurs and craft enthusiasts are also designing new motifs for creative purpose.

1.5 Anticipated Benefits Brought by the Proposed System

By analyzing the current synopsis this can be said that, in order to revive the endangered heritage there are two main concerns to be focused on: (1) Resuscitating the industry by motivating the weavers and one way to do it is through launching collaboration between weaver and designers. (2) The enlarging scope of motifs generation may cause the traditional motifs get altered. Hence, we take such attempt to develop a system which will mimic the pattern of Jamdani design in the digital images and generate motifs that has the essence of the real ones. This creative machine dedicated to generate Jamdani motifs will:

- Play the role of an intelligent artist.

- Bridge the gap between the weavers and designers.
- Become a handy tool for art enthusiasts create motif effectively in a efficient way.
- preserve the motif from further extinction by carrying on the legacy of the surviving motifs.
- Expose this exquisite industry in a international level.
- The artificial production of Jamdani motifs will unveil a new source of inspiration for artists by playing the role of an intelligent tool of imagination. Our system is trained in such a way that the visual and artistic appeal of the produced motifs remains intact.

1.6 Our Contribution

1.6.1 Development of an Unique Dataset

We generated a dataset called **Jamdani Noksha** containing images of authentic Jamdani motifs. Each data of our dataset is a combination of two domain; a source domain and a target domain merged side by side. The source domain is the skeleton of the motif and the target domain is the authentic Jamdani motif collected from difference sources. The dataset is available here: https://github.com/raihan-tanvir/generative-jamdani

1.6.2 Approach Towards Building the Proposed System

We presented a motif generation technique that will create a new Jamdani motif from a given stroke. Our system is trained on our **Jamdani Noksha** dataset and uses conditional Generative Adversarial Network approach. Our system inputs a rough stroke from the user as an initial guess of the motif. Samples of given input and generated output is demonstrated in figure 1.5. A simulation of our application can be seen in 5.13).

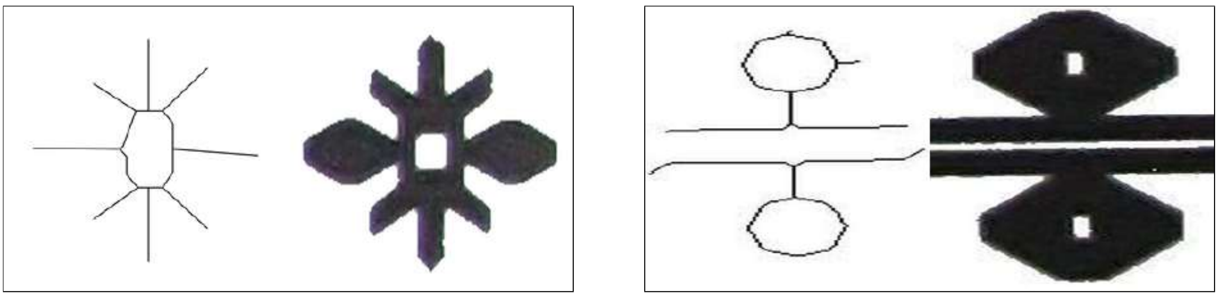


Figure 1.5: Some output of our work: Generating Jamdani motif (*right*) from sketch (*left*).

Being trained, our model can generate entirely new samples of Jamdani motif which can later be referred for weaving. We investigated for different input strokes and demonstrated the outputs. The steps we followed are described below and shown in figure 1.6

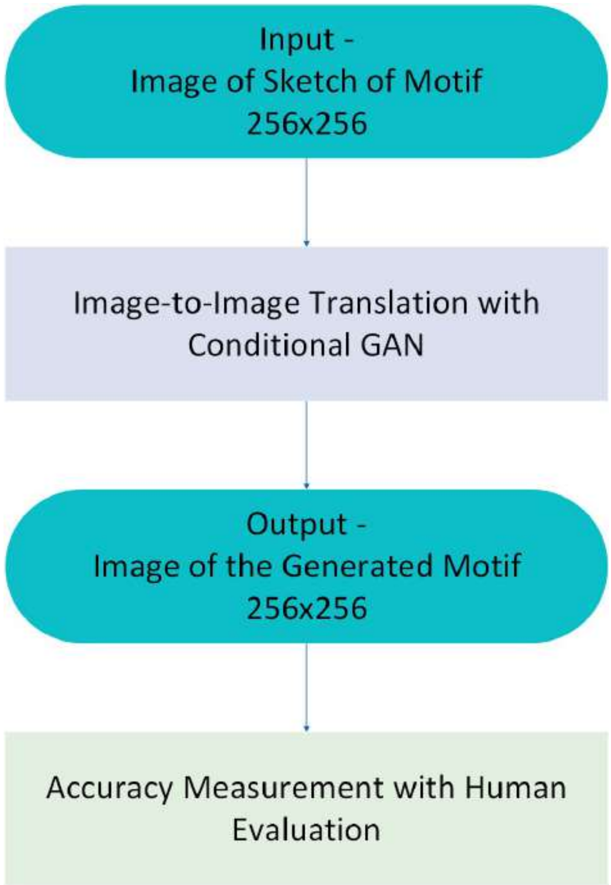


Figure 1.6: Process model for our thesis. We provide a sketch of the motif (256 × 256) for the image to image translation with conditional GAN model. Finally the resultant image (256 × 256) which is a generated Jamdani motif is measured with human evaluation for accuracy.

The initial research on Jamdani Motif Generation has been accepted in International Conference on Computer and Information Technology (ICCIT), 19-21 December, 2020, Dhaka, Bangladesh. Our paper, "**Jamdani Motif Generation using Conditional GAN** " is to be published in IEEE Xplore Digital Library.

Chapter 2

Background Studies & Related Works

Firstly, we explain the concept of Generative Adversarial Networks [13] —a groundbreaking invention in the field of Deep Learning—since we exploited this in our work. The basic components of GAN are two neural networks—a generator that synthesizes new samples from scratch, and a discriminator that takes samples from both the training data and the generator's output and predicts if they are 'real' or 'fake'. The generator input is a random vector or can be stated as a noise and therefore its initial output is also noise. Over time, as it receives feedback from the discriminator, it learns to synthesize more realistic images. The discriminator also improves over time by comparing generated samples with real samples, making it harder for the generator to deceive it. Many improvements to the GAN architecture have been proposed through enhancements to the discriminator model with the idea that a better discriminator model will, in turn, lead to the generation of more realistic synthetic images.

2.1 Motivational Works

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. The goal of Generative Adversarial Networks is to synthesize artificial samples, such as images, that are indistinguishable from authentic images that mean it can be used to create anything in any domain like images, music, speech, and writing. They are like robotic artists and their outputs are pretty impressive. Choosing a research area is a hard nut to crack. But the way the field of computer vision has expanded has astonished us. We have gone through certain works related to this field and the mesmerizing outputs have created an interest within us to dig into deeper. There are numerous ways of adding diversity to this field. The idea of making computers doing automated tasks as human vision do is certainly fascinating.

Below we list different variations of Generative Adversarial Networks that inspired us to use them in our research.

2.1.1 Image-to-Image Translation with Conditional Adversarial Networks (*pix2pix* GAN)

The *pix2pix* GAN [2] is a general approach for image-to-image translation. It is based on the conditional generative adversarial network, where a target image is generated, conditional on a given input image. *pix2pix* GAN changes the loss function so that the generated image is both plausible in the content of the target domain, and is a plausible translation of the input image.

The approach was presented by Phillip Isola, et al. in their 2016 paper titled "Image-to-Image Translation with Conditional Adversarial Networks" and presented at CVPR in 2017. The GAN architecture is an approach to training a generator model, typically used for generating images. A discriminator model is trained to classify images as real (from the dataset) or fake (generated), and the generator is trained to fool the discriminator model. The Conditional GAN, or cGAN, is an extension of the GAN architecture that provides control over the image that is generated, e.g. allowing an image of a given class to be generated. *pix2pix* GAN is an implementation of the cGAN where the generation of an image is conditional on a given image.

The generator model is provided with a given image as input and generates a translated version of the image. The discriminator model is given an input image and a real or generated paired image and must determine whether the paired image is real or fake. Finally, the generator model is trained to both fool the discriminator model and to minimize the loss between the generated image and the expected target image.

2.1.2 A Style-Based Generator Architecture for Generative Adversarial Networks

The Style Generative Adversarial Network, or StyleGAN [16] for short, is an extension to the GAN architecture that proposes large changes to the generator model, including the use of a mapping network to map points in latent space to an intermediate latent space, the use of the intermediate latent space to control style at each point in the generator model, and the introduction to noise as a source of variation at each point in the generator model. The resulting model is capable not only of generating impressively photo-realistic high-quality photos of faces, but also offers control over the style of the generated image at different levels of detail through varying the style vectors and noise.

Figure 2.1: *pix2pix* GAN Translation from Sketch to Bag (*top*) and Sketch to Shoe (*bottom*) . Input at (*left*) ground truth at (*middle*) and the output at (*right*) [2]

Figure 2.2: Destination faces are given row wise and the source faces are given column wise. Output is given at the cross points of the source and destination images.

An experiment based on StyleGan [16] is given in figure 2.2 Here no faces are real human face but they look incredibly like a real human.

2.1.3 CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Cross-domain transfer GANs [3] will be likely the first batch of commercial applications. The goal of this model is to learn a mapping $G : X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, they coupled it with an inverse mapping $F : Y \rightarrow X$ and introduce a cycle consistency loss to enforce $F(G(X)) \approx X$ (and vice versa) [3]. These GAN transform images from one domain to another domain e.g: real scenery to Monet paintings or Van Gogh paintings and Summer to Winter and Winter to Summer image transfer. We can visualize the examples in figure 2.3

2.1.4 PixelDTGAN-Pixel-Level Domain Transfer

Now-a-days suggesting merchandise based on celebrity pictures has been popular for fashion blogger and e-commerce. *PixelDTGAN* [4] creates clothing images and styles from an image. An input domain is transferred to a target domain at the semantic level and generates the target image at the pixel level in this model. This work is distinct in terms of *image-conditioned image generation*. They take an image as a conditioned input lying in a domain and re-draw a target image lying on another. There are two domains defined

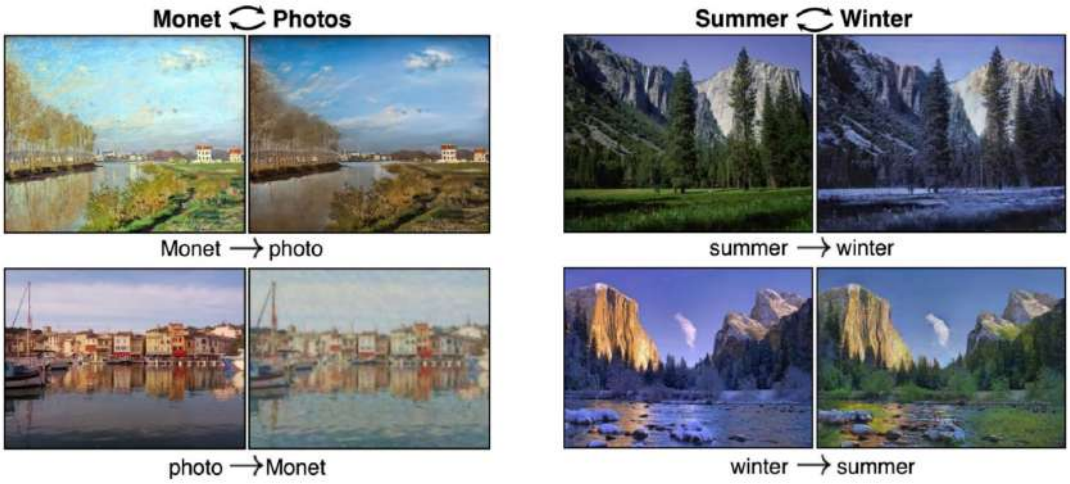


Figure 2.3: Monet paintings to real scenery (*Upper left*), real scenery to Monet painting (*lower left*) and Summer to winter (*Upper right*) and Winter to summer (*lower right*) describing *CycleGAN* [3]

Figure 2.4: In each group a model is wearing a cloth (*left*) and the suggested cloth is shown separately (*right*) [4]

here; a source domain and a target domain. These domains are connected by semantic meaning. For instance, If an image of a dressed person is defined as a source domain, the person's clothing is then called the target domain. They have presented a high-quality clothing dataset containing the previously discussed two domains and succeeded to demonstrate a decent result. [4]. Some examples of *PixelDTGAN* is as follows 2.4

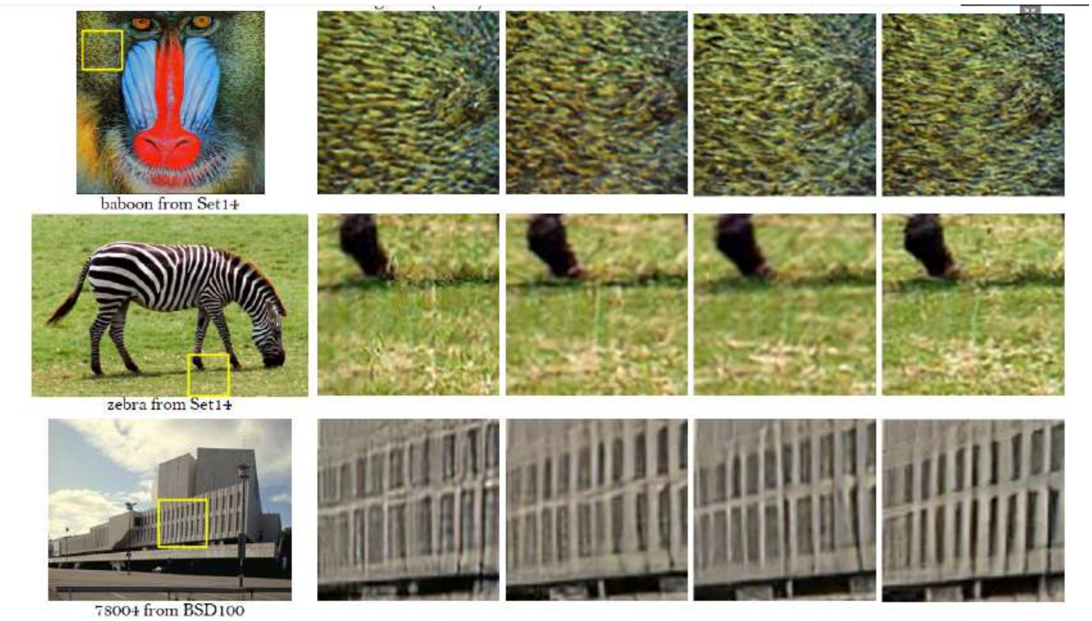


Figure 2.5: In each group the image which is to be enhanced is at *left* and the gradually enhanced versions are given side by side. [5]

2.1.5 Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN)

The Super-Resolution Generative Adversarial Network (SRGAN) is a seminal work that is capable of generating realistic textures during single image super-resolution. However, the hallucinated details are often accompanied by unpleasant artifacts. To further enhance the visual quality, three key components of SRGAN - network architecture, adversarial loss, and perceptual loss, and improve each of them to derive an Enhanced SRGAN (ESRGAN) [5].

2.1.6 LUCSS: Language-based User-customized Colourization of Scene Sketches

Here in this paper [6] we find a technique which can colorize any scene sketch according to a sentence where the colors are defined. Three sequential modules combinedly create this model. A sketch is given at first. Then a captioning module is used to generate the description of a text with spatial relationships based on the instance-level segmentation results. Then the colorization module gives chance to the user to edit the caption and create a colored image of the sketch based on the edited caption. See figure 2.6

Figure 2.6: An example of sketch with different colors according to the sentences given below the individual images [6]

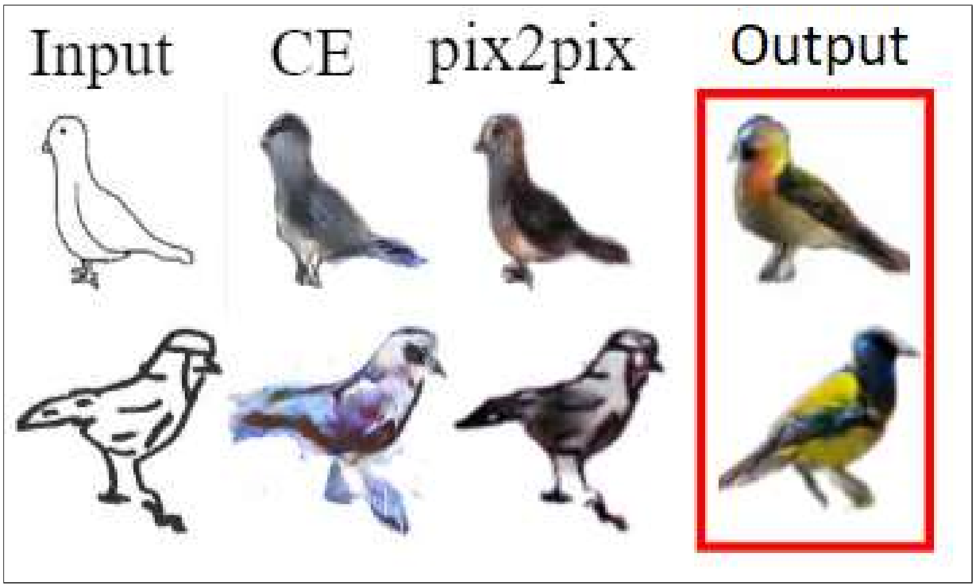


Figure 2.7: Input sketch at the left and output for CE [7], *pix2pix* [2] and Deep Contextual Completion [8] are given one by one for each group.

2.1.7 Sketch-to-Image Generation Using Deep Contextual Completion

In *pix2pix* [2] the output edges follows the input edges but in this paper [8] the output edges do not necessarily follow the input edges. Here in this model, they introduced an image generation technique using a novel joint image completion approach. A sketch is used to provide the context of the image for generating a new image. Instead of having a strict alignment, this model is still faithful to the input, resulting in more realistic images. See 2.7

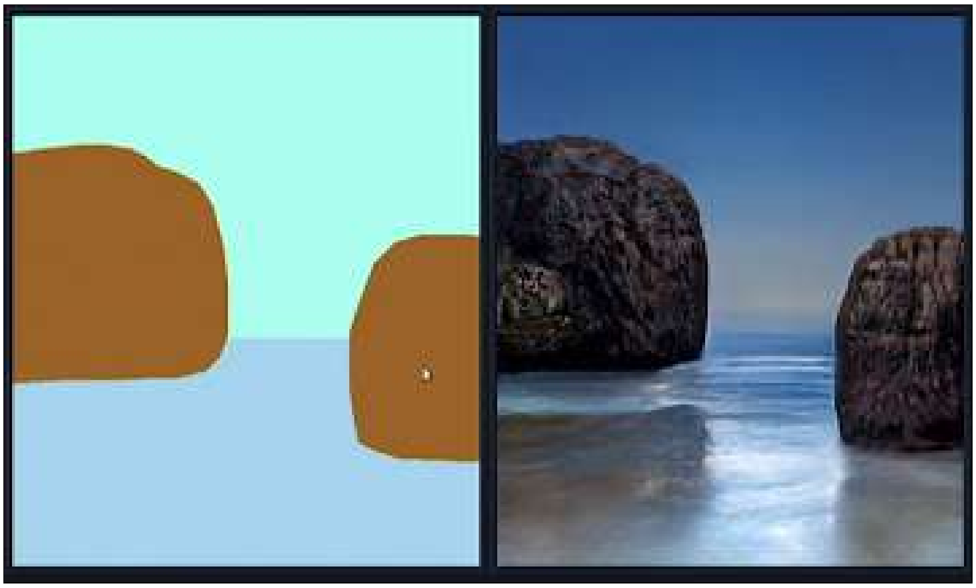


Figure 2.8: The given segmentation at the left and the output for GauGAN [9] at the right.

2.1.8 Stroke of Genius: GauGAN Turns Doodles into Stunning, Photo-realistic Landscapes

GauGAN [9] can convert segmentation maps into lifelike images. This method allows a user to draw a segmentation map of his/her imagination and manipulate the scene. To achieve this the user has to label each of his drawn segments with labels like the sea, spring, sand, sky, etc. Catanzaro, vice president of applied deep learning research at NVIDIA said about GauGAn that it was like a coloring book picture that described where a tree was, where the sun was, where the sky was. This GAN model can offer a tool that is very powerful for creating virtual worlds for everyone from architects and urban planners to landscape designers and game developers [9]. A great demo of GauGAN is given in figure 2.8.

2.2 Related Work

2.2.1 Hand-loom Design Generation using Deep Neural Networks

The aims of this model to use techniques like Conditional GAN, Image to Image translation, Texture, and content transfer for aiding as a design tool for handloom weavers and also designers and industry as a whole [10]. This paper is accepted at the 27th IEEE International Conference on Image Processing (ICIP 2020) but not yet published as a full paper.

Figure 2.9: Examples of Normal sarees (*left*) and the Handloom sarees (*right*) from the paper [10]

Similarities

- The main target of both the work is to generate images aiming to work towards the betterment of weavers and the handloom industry.
- Both the work is based on Generative Adversarial Network (GAN). They have also worked on the Conditional GAN and a state-of-the-art method *pix2pix* [2].
- They have also applied image-to-image translation techniques to generate new samples like us.
- They have developed an application using the model they trained to generate the output from the given input. We have also prepared a similar application that can take a stroke and generate new motifs.
- They have also created a dataset for this work like us. Their dataset consists of two collections of images - Normal sarees and Handloom sarees which are 26100 and 1185 in size respectively. The samples of their Normal and Hand-loom version of their dataset is given in figure 2.9

Dissimilarities

- Though our approach has similarities with this work in terms of the uniqueness of the dataset, the biggest difference is the domain of both works. Our work is completely based on the Jamdani motifs but they have worked on different hand-loomed sarees generally.
- The experiments are slightly different for these two works. We have experimented on *pix2pix* method using various versions of our dataset as our main target is to generate

Figure 2.10: Samples of Saree → Handloom using CycleGAN [3] (*left*) and DiscoGAN [11] (*right*) version of their own dataset. In each group from left to right, the original saree image, translated image and the reconstructed image are shown respectively

Jamdani motifs from a stoke given by users. But they have also explored CycleGAN [3] and DiscoGAN [11] which are different image-to-image translation methods to generate new designs.

- There is a slight difference in our data formats. The resolution of our data is $256 \times 256 \times 2$ as we are using the *pix2pix* data format. But the resolution of their data is $256 \times 256 \times 3$ as they have worked on different methods where the first part of the data format is the original image, the second part is the translated one and the third part is the reconstructed image.

- Our main focus is to create the basic motif from stroke whereas the referenced authors attempted creating a whole design for better representation on clothes. Some output of their experiments are shown in figure 2.10

Though there are some similarities and dissimilarities, the ultimate target of both the work is to work towards the betterment of weavers and handloom industry. Both the work can serve the designing industry in a better way.

Chapter 3

Data Collection & Processing

3.1 Our Dataset: The Jamdani Noksha

As there was no previous dataset of Jamdani motifs available for us to work on, the first and foremost step was building a data set of our own to serve our intended purpose. This is the key to our research and the entire output depends on this.

3.1.1 Data Authenticity

In order to build a data set the collected data or images needed to be authentic and suitable for performing computer vision-related research work. But as there has not been any similar research performed earlier in this area, we were the spearheads. So making sure the data is authentic as well as suitable to match our needs was vital. Though there is a huge collection of Jamdani photographs stored on the internet. But it was not possible for us to rely on them. This is because:

- The authenticity of data found on the internet cannot be assured.
- The data found there was not suitable as the photographs were not done to serve any research purpose.

So there was no option for us other than collecting and building the dataset on our own while keeping the authenticity intact.

3.1.2 Data Collection

Our exploration began with the visit to Jamdani festival, 2019 at Bengal Shilpalay, Dhaka, jointly organized by the National Crafts Council of Bangladesh and the Bengal Foundation in association with Aarong, Kumudini, Tangail Saree Kutir, and Aranya. For experiencing the history of Jamdani. The exhibition gave us an overview and evolution of the Jamdani industry to this day. There we observed the old and new motifs on different sarees and photographs and got introduced to a huge archive enriched with the knowledge of authentic Jamdani designs. We watched documentaries. We also interviewed designers, artists, weavers, and experts working on preserving and enhancing Jamdani history and culture. This enabled us to build basic knowledge about this ancient piece of excellence. The archive we came across at the festival is the initial source from where we have started collecting our dataset (figure 3.1).

Figure 3.1: Our team at Jamdani Festival, 2019 [12], Bengal Shilpalay, Dhaka (*top*) and some photos of Jamdani Festival (*bottom*)

Next, we visited *Sonargaon*, the birthplace of Jamdani. Our team was invited to Shahina Jamdani Weaving Factory, one of the oldest and most prominent Jamdani factories of Bangladesh, at Rujganj, Narayanganj. We had the expledid opportunity to witness the whole weaving process starting from preparing the yarn till the application of unique techniques of the weavers on the loom. From there, we photographed designs weaved on sarees and panjabees, and other fabrics for our initial data set. Mr. Juned Ahmad Muhtaseem, the organizing body member of Jamdani Festival, 2020 and Mr. Shojib, one of the owners of Shahina Jamdani Weaving Factory are the two contributors who have helped and facilitated us in collecting authentic Jamdani motifs 3.2. There we collected some pictures of Jamdani sarees and panjabees. Some of them were directly from the loom. Some of the collected images are shown in figure 3.3

Figure 3.2: Our team members while visiting the weaving factory, Rupganj (*top*) and Data collection from Shahina Jamdani Weaving Factory, Jamdani Polli (*bottom*).

The second source of the raw pictures for **Jamdani Noksha** dataset was a book called "Traditional Jamdani Designs" by National Craft Council Bangladesh [1]. It is an archive for authentic traditional Jamdani motifs. This book was published under the project Preservation of Jamdani Motifs and Designs- an endangered handloom of Bangladesh supported by Ambassador's Fund for Cultural Preservation, Embassy of the United States of America. We collected this book from Jamdani Festival, 2019. This book is not only a source of our data set but also an enriched source of information on the Jamdani industry of Bangladesh. The book was written in such a way that it is helping anyone interested to perform research on Jamdani from any walk of life. Some sample pictures of different designs are shown in figure 3.4.

Figure 3.3: Sample design collected from sarees and panjabees while visiting the weaving factory, Rupganj

Figure 3.4: Data sources book: "Traditional Jamdani Designs" by National Craft Council Bangladesh [1] (*top*), Some Jamdani designs collected from Traditional Jamdani Book [1] (*bottom*)

3.1.3 Data Processing

Motifs are the building blocks of Jamdani Designs. Each Jamdani design is a combination of different types of motifs. As we are working with the motifs and adopting *pix2pix*, we had to pre-process the photographs we collected from various authentic sources. Firstly every possible Jamdani motif was extracted by cropping the designs into smaller parts. An example of cropping the motifs from a complete design of a Jamdani saree is given in figure 3.5

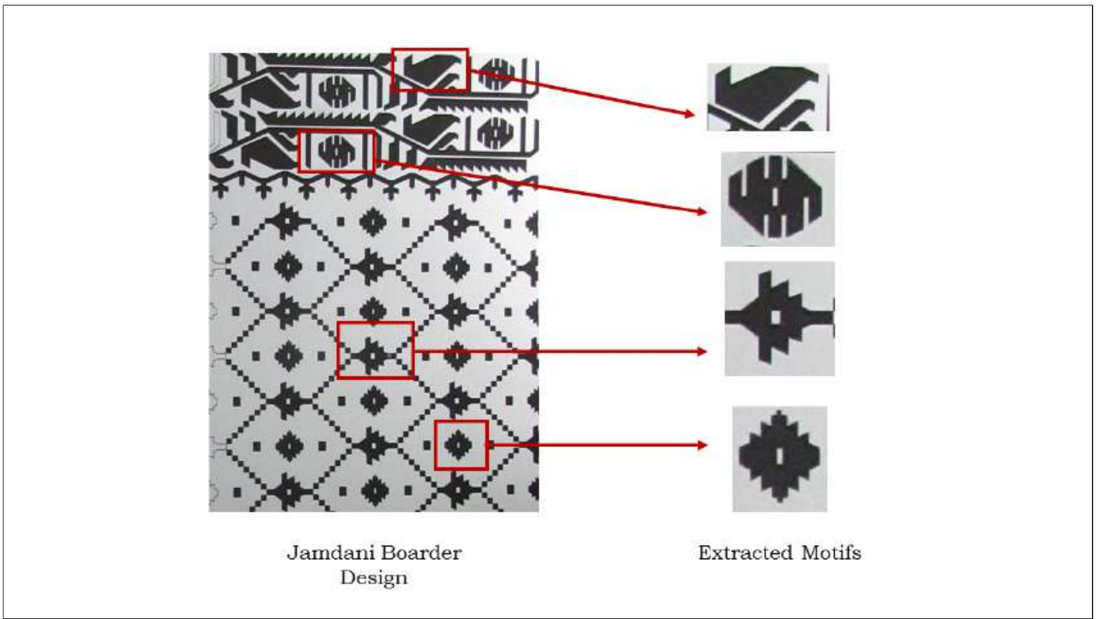


Figure 3.5: An example of cropping the motifs from a complete design. The black square boxes indicate the motifs which are cropped out for our dataset.

Our target is to generate a motif from a given stroke. Therefore the dataset must include samples as a pair of motifs and strokes so that the model can learn well. We followed pix2pix data format which is given in figure 3.6. Some example of different dataset of this format is also given in 3.6.

However, it is not feasible to include handmade strokes for each of the samples. We found different morphological operations as a replacement of the strokes and we pre-processed our dataset. We built five versions of our dataset—*Skeleton*, *Reduced Branch*, *Sketch*, *Boundary*, and *Enhanced Resolution*. Each samples in the dataset is formatted as a pair of processed and original images according to *pix2pix* [2] for conditional Generative adversarial networks. We explained the processing of this individual version below.

(a)

(b)

(c)

Figure 3.6: (a) Data format for storing samples in our dataset. Processed (A) and original version (B) are kept sidewise. (b) Edge to Bag data sample (c)Edge to Shoe data sample .

3.1.3.1 Skeleton Version

Generating Real Image to binary image: For this version, we consider morphological skeletons as a representation of the motifs. For this purpose, We applied different morphological image processing like opening, closing, dilation and erosion on the cropped images according to their needs.

We have collected our data from different sources such as fieldwork and the book called "Traditional Jamdani Designs" [1]. So there are different types of datasets. As the data collected from the fieldwork and the designs collected from the book which were the photographs of different sarees are not so tidy and the interlacing of the yarns are completely visible, we had to face a lot of problems processing them. Besides, the background is not clear. So after binarizing the images we found some unwanted foreground pixel (white pixel). Some of the pixels were inside the foreground and some were outside the foreground. So using the closing operation we cleared the gap inside the foreground which was created for the interlacing of the yarn. Some example of using closing operation is shown in figure 3.7

Again opening operation was used to clear the outside unwanted pixels which were the cause of unclear background. Some example of using opening operation is shown in figure 3.8

As we cropped out the motifs from a complete design some of the parts of other motifs were also visible as intruders. So after binarizing, we had to remove the unwanted parts of other motifs by cropping out them from the binary image and making the color of those intruder

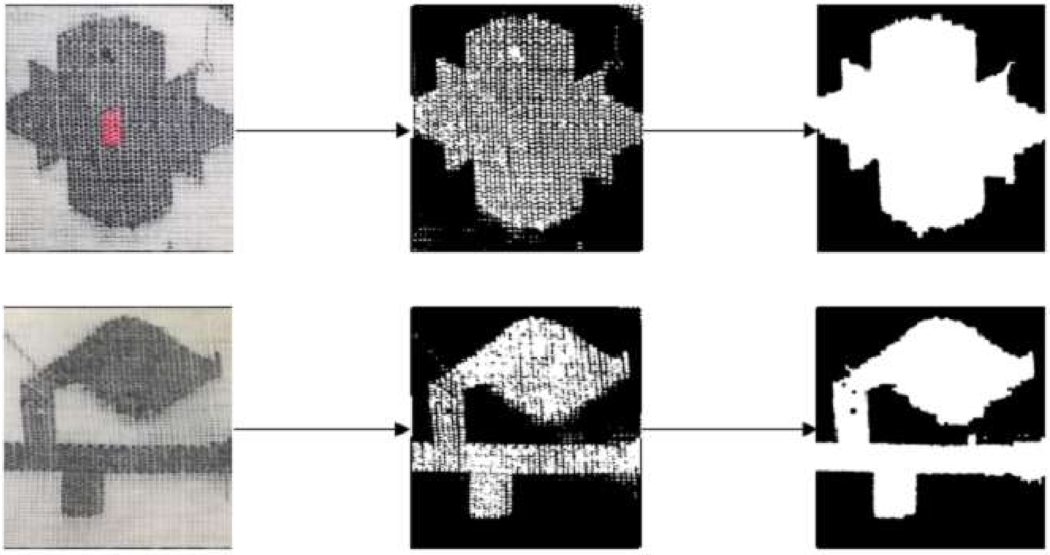


Figure 3.7: In each group *Left to right:* A jamdani motif image at the left and result of binarizing at the middle and the result of binarizing after applying closing operation at the right.

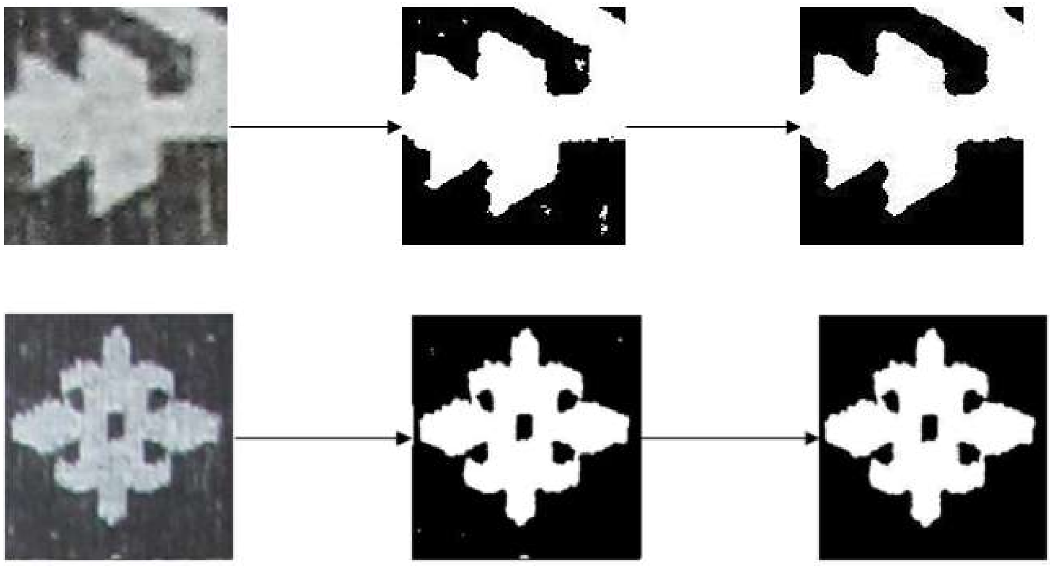


Figure 3.8: In each group *Left to right:* A jamdani motif image at the left and result of binarizing at the middle and the result of binarizing after applying opening operation at the right.

pixels to zero (Black). See figure 3.9.

Multiplication on Real and Binary Image: The original image is multiplied with the resulting image obtained from the previous step which is the binary image to remove the background and to keep only the portion of the image containing the motif. Examples describing the multiplication is given in figure 3.10.

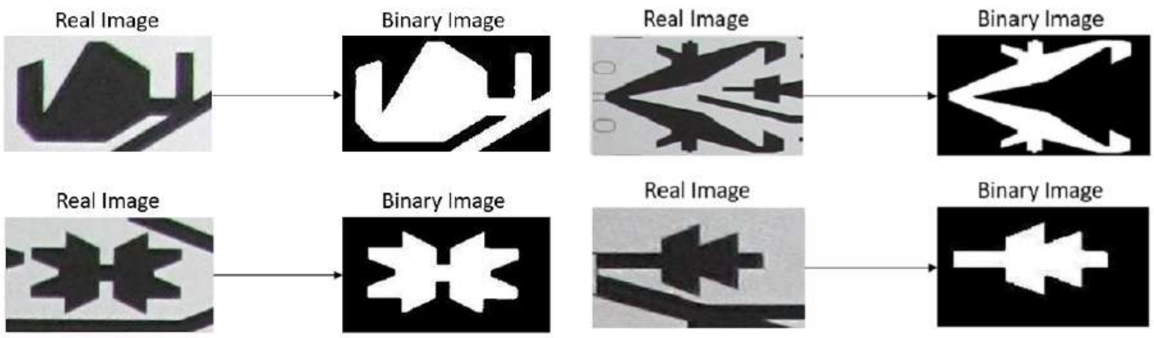


Figure 3.9: In each group *Left to right:* A jamdani motif image at the left and result of binarizing after applying different morphological image processing technique and cropping operations at the right.

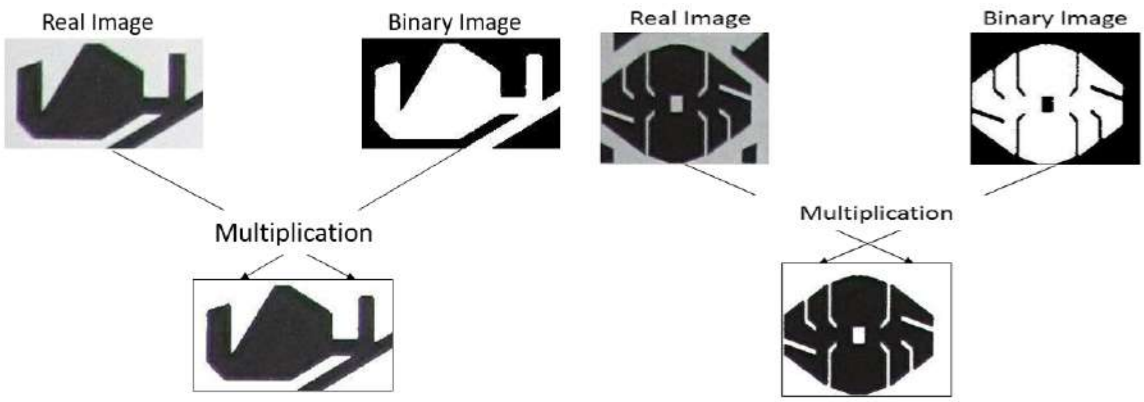


Figure 3.10: For each group multiplication of original image (*upper left*) with the binary image (*upper right*) and the result (*below*)

Generating Skeleton: The generated pattern is then used for making a skeleton using the basic skeletonize function [17] [18]. It works by making successive passes of the image, where on each pass, border pixels are removed maintaining the constrain that they do not break the connectivity of the corresponding object. Some example of skeletonizing are like 3.11

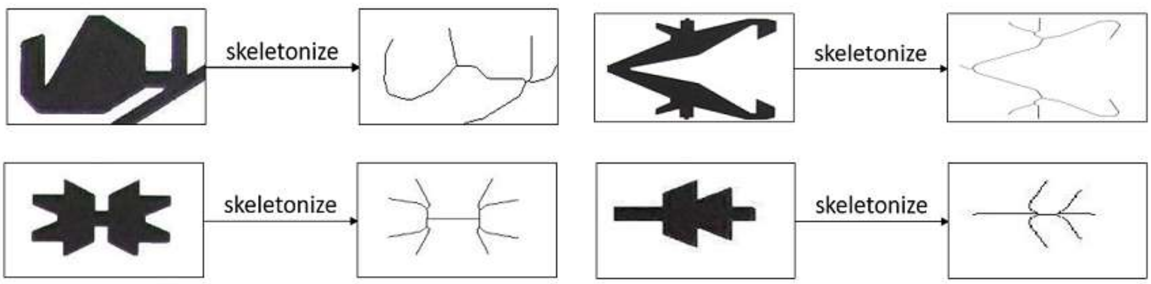


Figure 3.11: Skeleton (*right*) of a motif in the image (*left*) for each group

Merging the Skeleton and the Motif: Finally, the skeleton and the generated pattern are combined side by side and individual data is made. Both the skeleton and generated pattern has the same height (256 pixels) and width (256 pixel). After the combination, it has a height of 256 pixels and a width of 512 pixels. An example of merging the skeleton and the motif according to the data format stated before is given in figure 3.12.

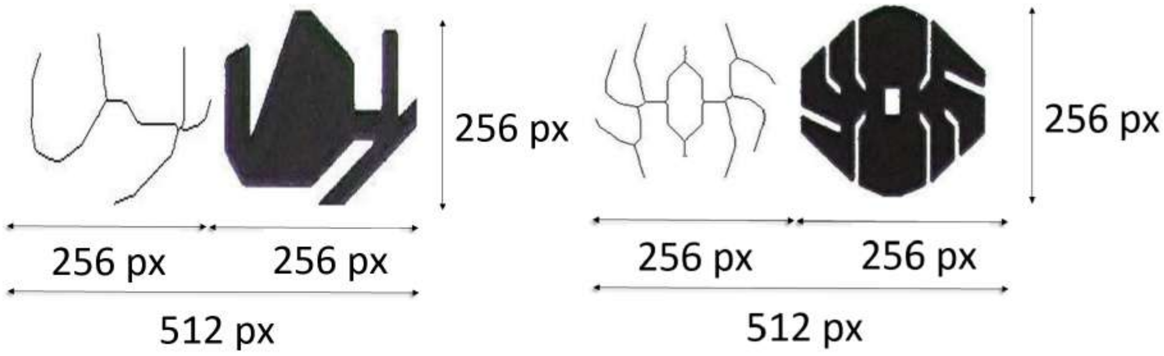


Figure 3.12: Examples of merging the skeleton (*left*) and the motif (*right*) for each data

The complete flow of operation to process a single data is shown in figure 3.13.

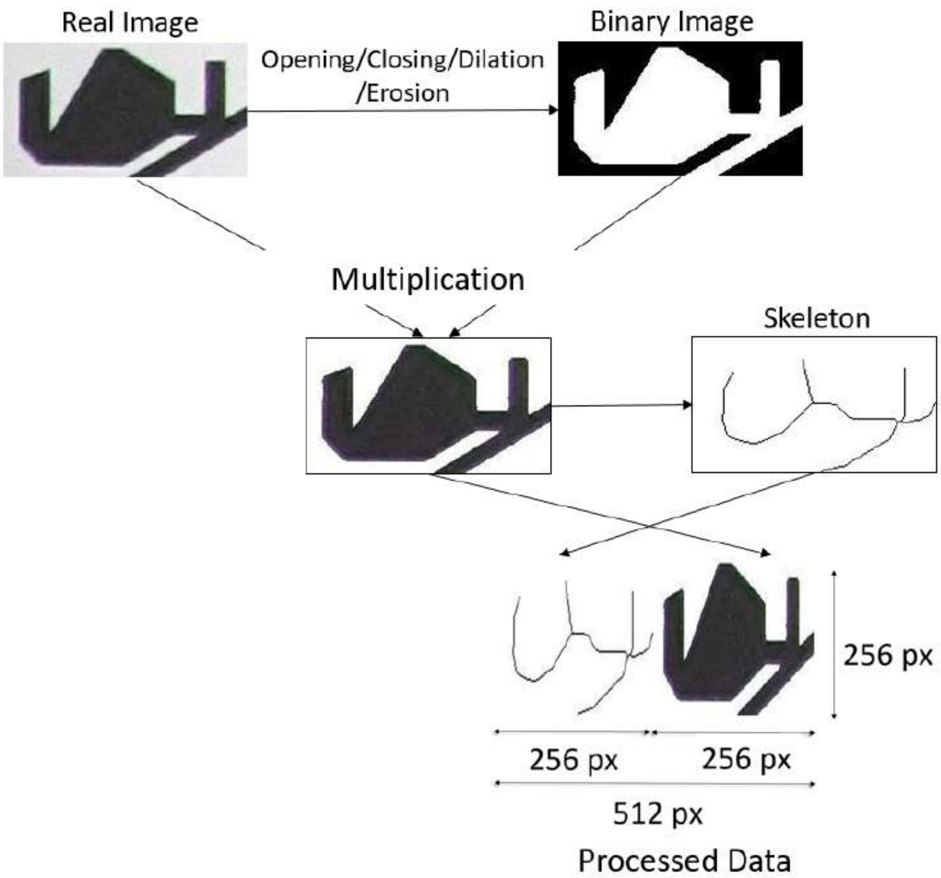


Figure 3.13: Complete flow of processing for preparing a data

A complete flow of operation to process a data for a different data is also shown in figure 3.14.

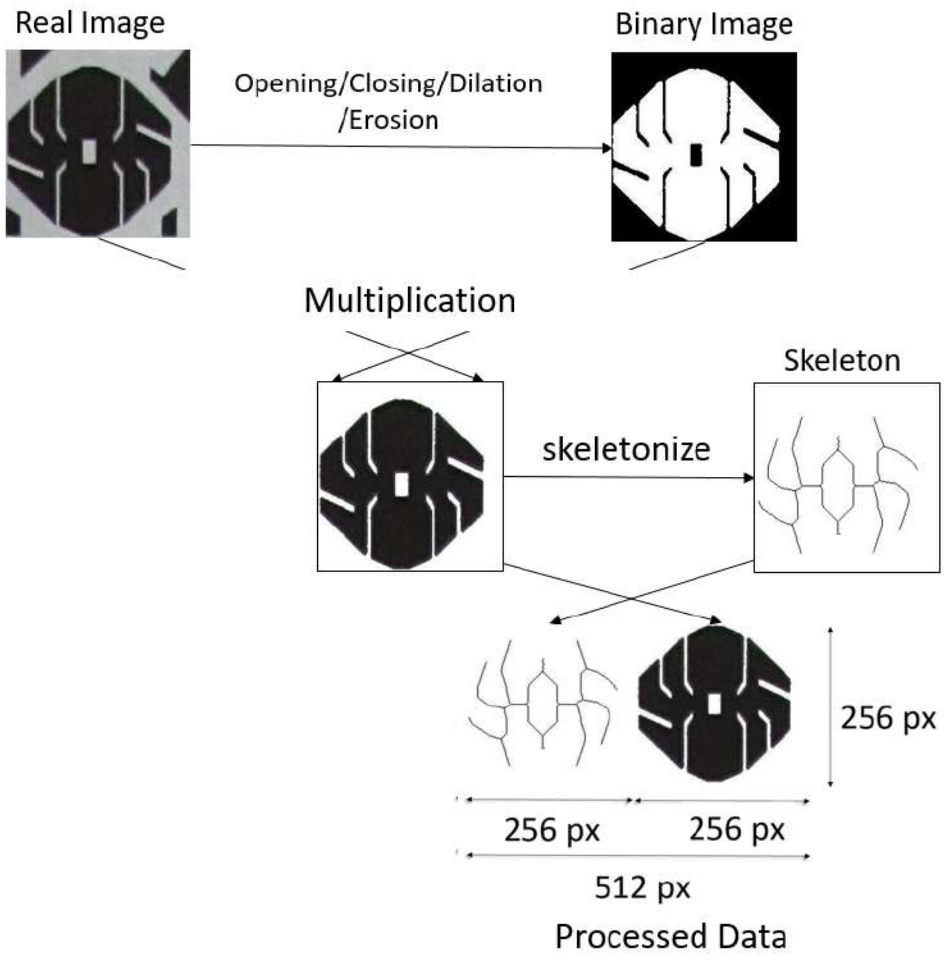


Figure 3.14: Complete flow of processing for preparing a data

3.1.3.2 Reduced Branch Version

Skeletonization very often provides unwanted branches which mislead the training. So we planned to remove the extra branches. To solve this, we tried morphological thinning but could not find a good result. Then we tried another method which can cut down the small branches keeping the major and long brunches at their place.

Firstly we performed an erosion operation on the binary image of the motif. So, the binary image lost some pixels from the boundary. After that, we multiplied the generated skeleton from the motif with the eroded version of the binary image. So the extra branches which were at the boundary were cut down and we got a new skeleton which was a reduced version of the auto-generated one. A figure visualizing the reducing process is given in 3.15

Some examples of differences between the auto generated and the reduced one is shown in figure 3.16.

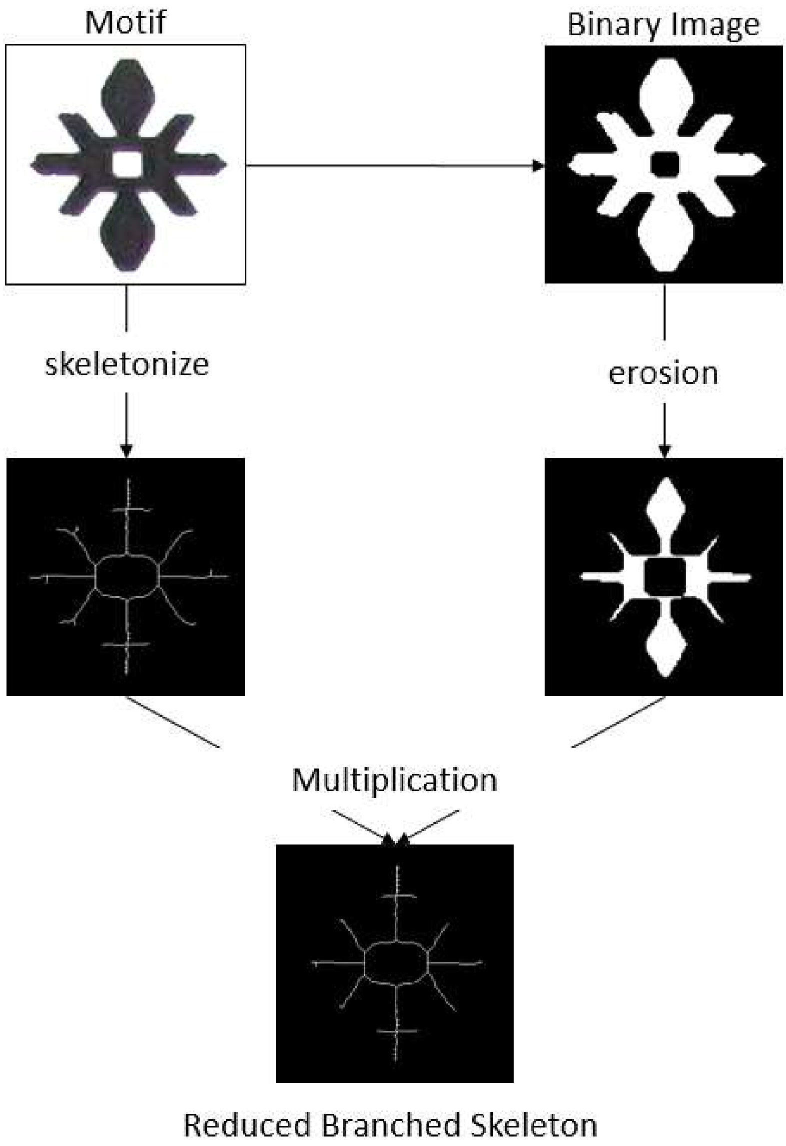


Figure 3.15: Flow of processing for reducing the small boundary branches of a skeleton

3.1.3.3 Sketch Version

So far we were working with the skeletons of the motifs which were generated using a built-in function called skeletonize function which works by making successive passes of the image, where on each pass, border pixels are removed maintaining the constrain that they do not break the connectivity of the corresponding object. [17] [18]. But this is a generalized process of generating the skeleton of an image. But the human mind is not generalized. Different people can draw different skeletons as the imagination of all the people is not the same. So, for further experiments to deal with realistic phenomena, we drew the sketches of the skeletons by hand for a more realistic dataset and created a dataset with human annotation. A group of volunteers contributed by sketching the blueprint of the motifs using a digital pen and no morphological operation was performed. Samples of hand

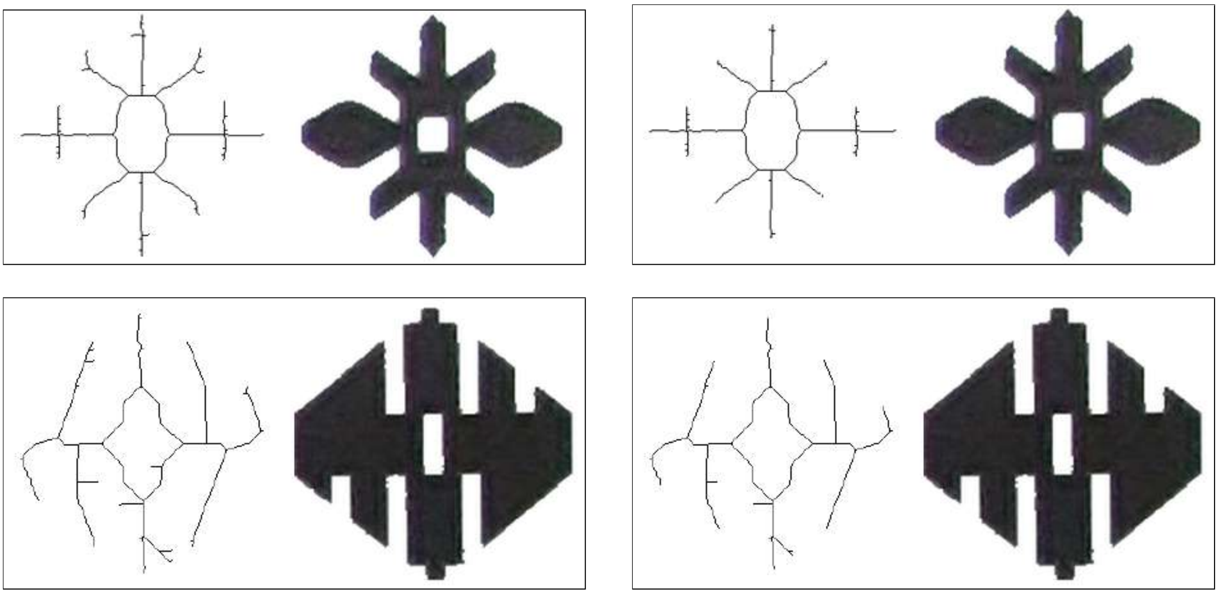


Figure 3.16: Before (*left*) and after (*right*) reduction of the branch for both (*top*) and (*bottom*). In each pair from left to right, a user given stroke and the corresponding motif is shown respectively

sketched data with a comparison with the auto generated one is given in figure 3.17.

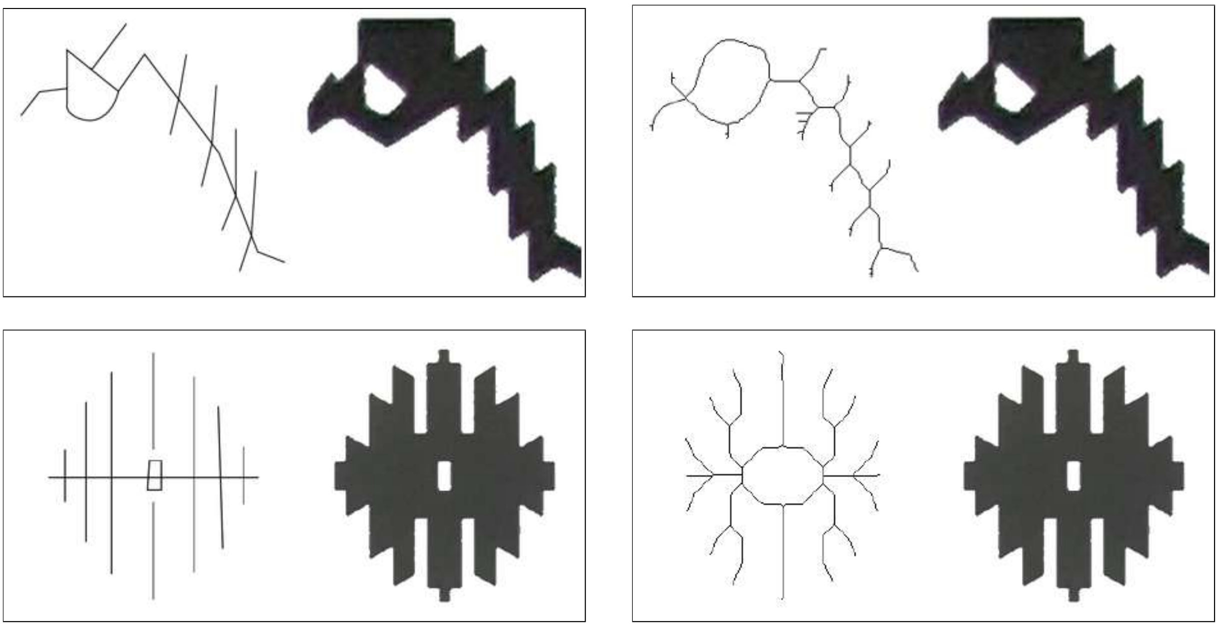


Figure 3.17: Samples of hand-drawn (*left*) and skeleton (*right*) version of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown respectively

3.1.3.4 Boundary Version

We also treated the boundaries of the motifs as input strokes. Therefore we created a version of our dataset with the contour of the patterns in images. As the motifs are the collections of some pixels and the intensity of the pixels are almost the same, we can detect the boundary using the Sobel edge detection technique [19]. [20]. Some samples of the **Boundary** version of our dataset are as follows 3.18

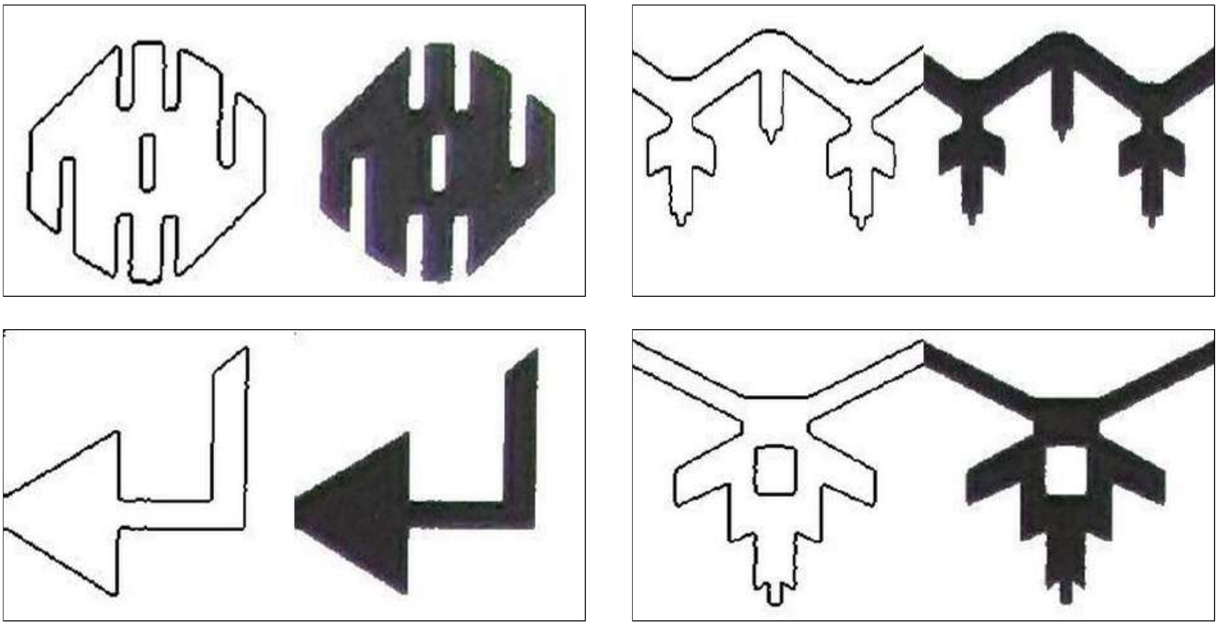


Figure 3.18: Samples of the **Boundary** version of our dataset. In each pair from left to right, a user given boundary and the corresponding motif are shown.

3.1.3.5 Enhanced Resolution Version

Some photographs are poor in terms of resolution which were taken directly from the field-work and the images of the sarees from the book [1]. The reasons behind the poor resolution are- (1) the threads that are clearly visible and (2) the background which is not clear enough to remove. The motif collected from these sources creates a lot of noise in the model. So, we separated the data with the better resolution which are taken from the book and prepared another dataset. Some samples of poor and enhanced resolution data is given in figure 3.19 and 3.20.

The 5 versions of our dataset and their sizes are given in the table 3.1.

A visual comparison among the versions of our *Jamdani Noksha* dataset are shown in figure 3.21 for a particular motif. As the version *Enhanced Resolution* is extracted from the the version *Skeleton*, data samples from these two versions of dataset are same.

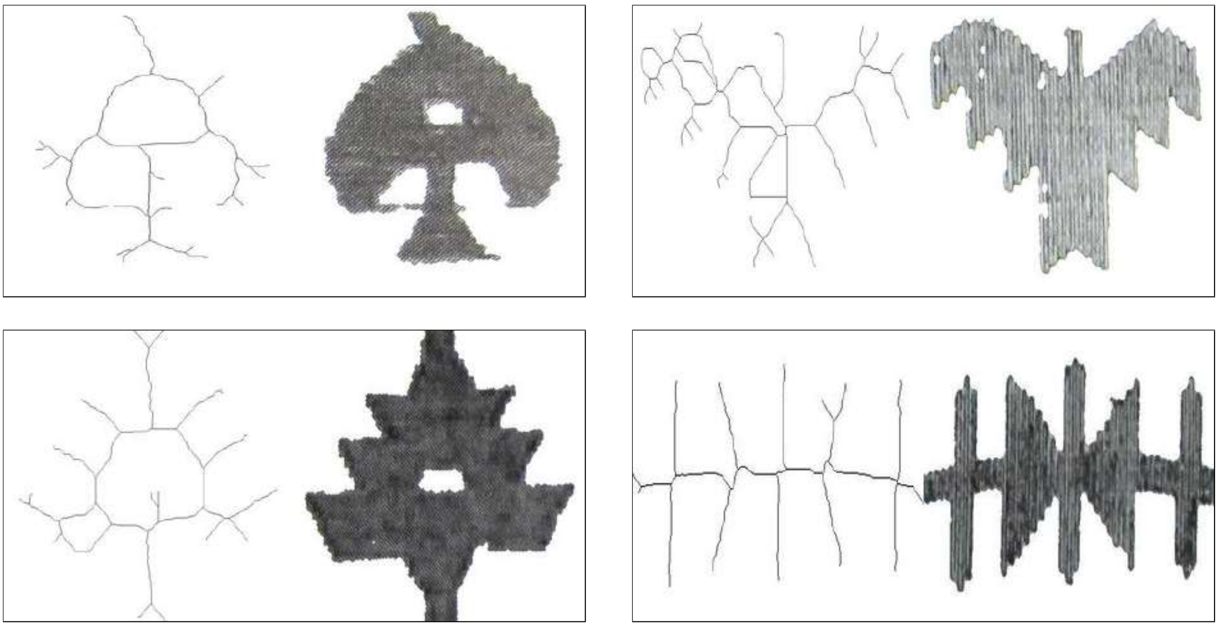


Figure 3.19: Samples of poor resolution data of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown.

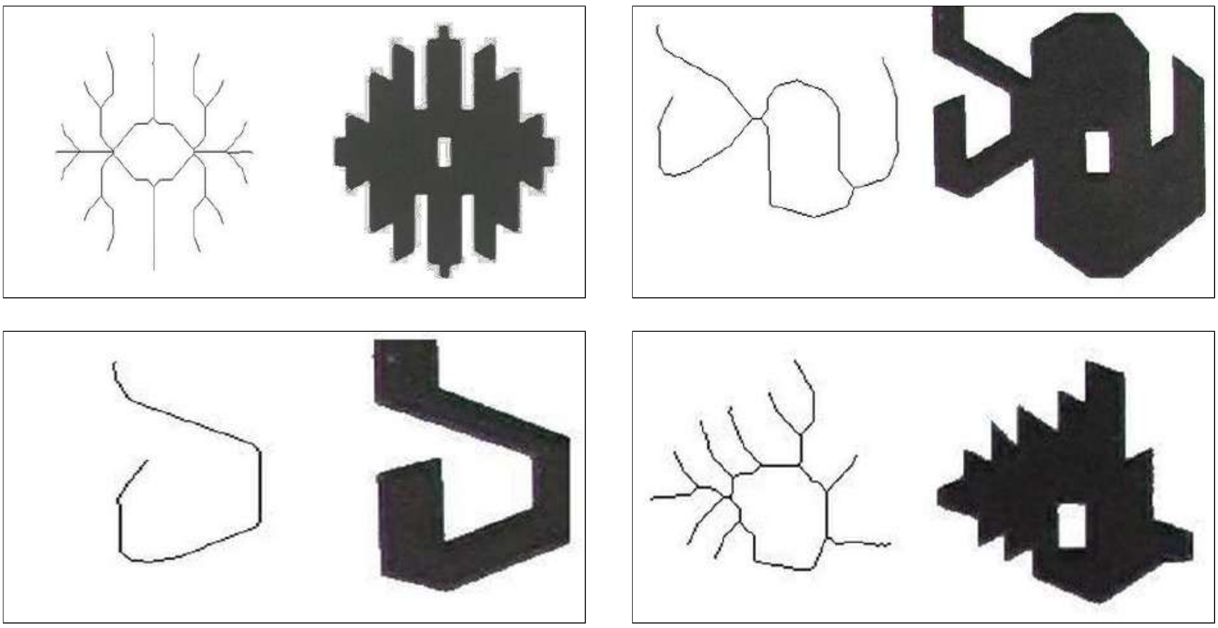
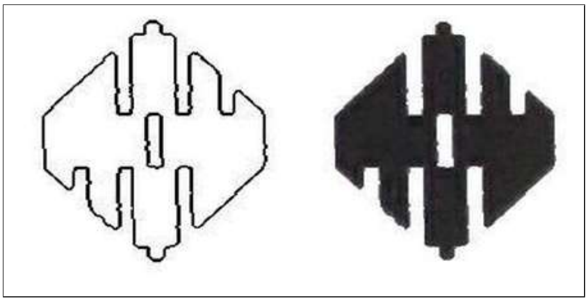


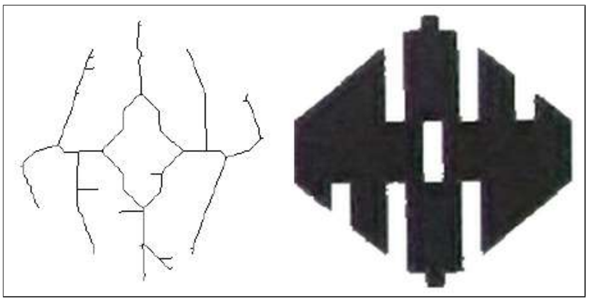
Figure 3.20: Samples of Enhanced resolution data of our dataset. In each pair from left to right, a user given stroke and the corresponding motif are shown.

SL No.	Version of Dataset	Size
1	Boundary	1116
2	Enhanced Resolution	1983
3	Reduced Branch	913
4	Sketch	910
5	Skeleton	7932

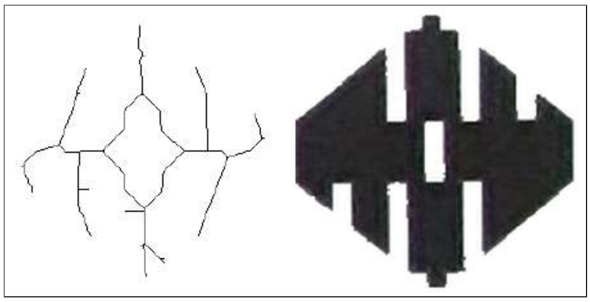
Table 3.1: Sizes of different versions of *Jamdani Noksha* dataset



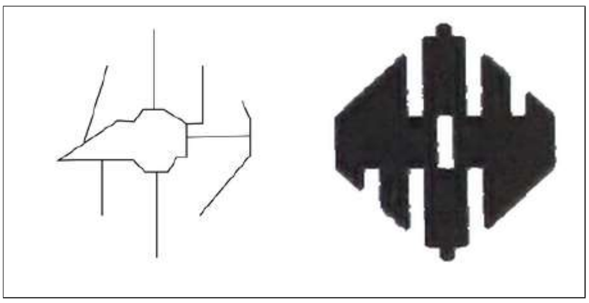
(a) *Boundary*



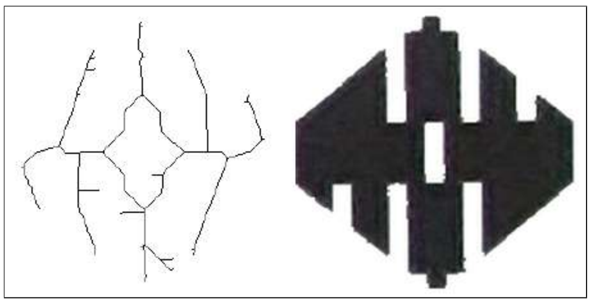
(b) *Enhanced Resolution*



(c) *Reduced Branch*



(d) *Sketch*



(e) *Skeleton*

Figure 3.21: A visual comparison of the different versions of our *Jamdani Noksha* dataset for a particular motif. In each pair from left to right, a user given boundary and the corresponding motif are shown.

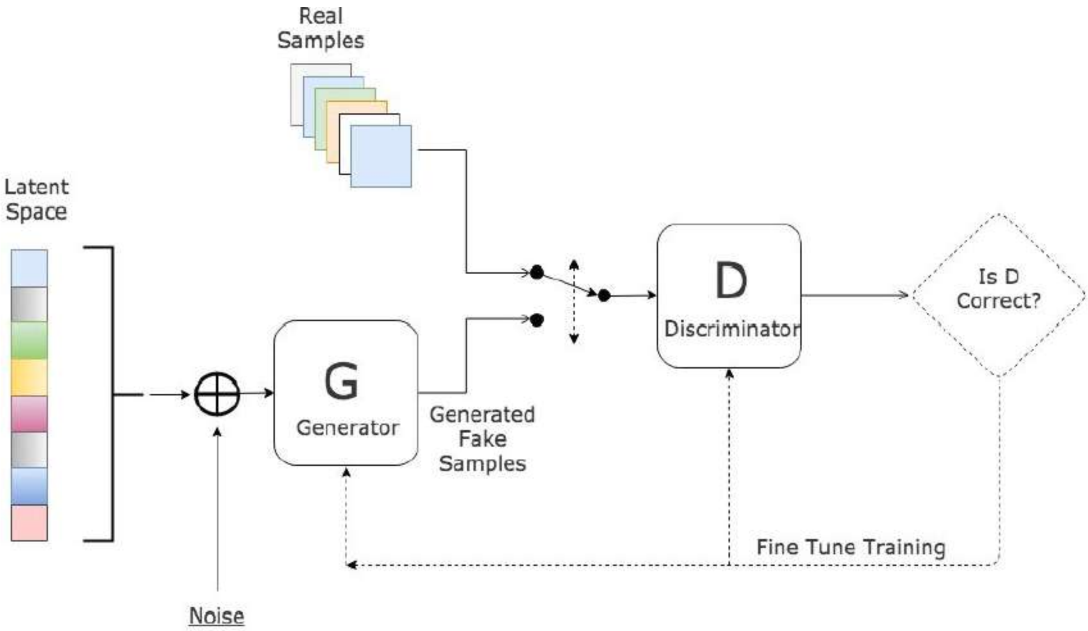
Chapter 4

Methodology

In this chapter, we discuss the different approaches we have accumulated for our model.

4.1 The Core : Generative Adversarial Networks

Generative Adversarial Networks are one of the most interesting ideas in computer science today. The basic idea behind GAN is actually very simple. At its core, a GAN includes two networks with competing objectives that work through opposing goals. This relatively simple setup results in both of the network's coming up with increasingly complex ways to deceive each other.



Source:http://www.google.com

Figure 4.1: High level representation of generative adversarial networks.

Generative Adversarial Networks take advantage of adversarial processes to train two neural networks who compete with each other until a desirable equilibrium is reached. In this case, we have a Generator Network $G(Z)$ which takes random noise as input and tries to generate data very close to the dataset we have. The other network is called the Discriminator Network $D(X)$ which takes generated data as input and tries to discriminate between generated data and real data. This network at its core implements a binary classification and outputs the probability that the input data actually comes from the real dataset (as opposed to the synthetic data).

In the formal sense the objective function of this whole process can be written as:

$$\mathscr{L}_{GAN}(G,D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,z))] \qquad ([13]) \tag{4.1}$$

The usual desirable equilibrium point for the above defined GANs (Eqn. 4.1) is that the Generator should model the real data and Discriminator should output the probability of 0.5 as the generated data is the same as the real data – i.e, it is not sure if the new data coming from the generator is real or fake with equal probability.

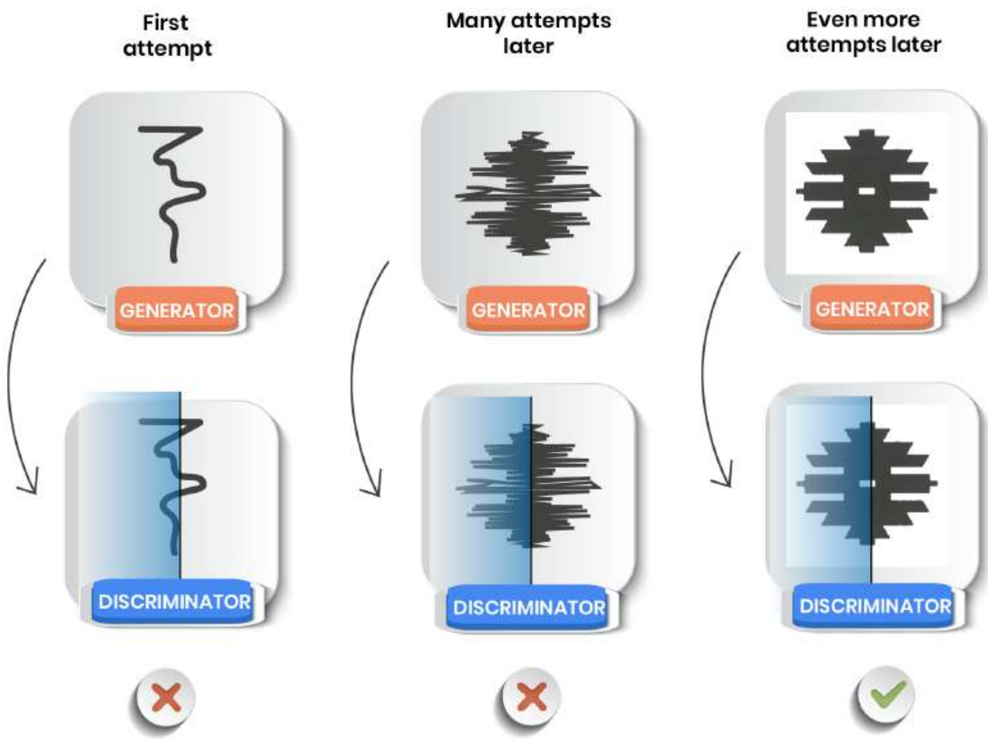


Figure 4.2: Initially, the generator generates an image from random noise that the discriminator detects as fake. After some iterations,the generator produces an image that somewhat looks real, yet the discriminator is not fooled. After training for enough iterations the generator becomes competent to generate an image that is good enough to fool the discriminator.

4.2 The Adaption : Image-to-Image Translation with Conditional Adversarial Network (Pix2Pix)

The pix2pix model [2] is a type of conditional GAN [21], where the generation of the output image is conditional on input, in this case, a source image. The discriminator is provided both with a source image and the target image and must determine whether the target is a plausible transformation of the source image.

The generator is trained via adversarial loss, which encourages the generator to generate plausible images in the target domain. The generator is also updated via L1 loss measured between the generated image and the expected output image. This additional loss encourages the generator model to create plausible translations of the source image.

GANs are generative models that learn a mapping from random noise vector z to output image y, $G : z \to y$. In contrast, conditional GANs learn a mapping from observed image x and random noise vector z, to $y, G : \{x, z\} \to y$. The generator G is trained to produce outputs that cannot be distinguished from real images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator's 'fakes'. This training procedure is diagrammed in Figure 4.3.

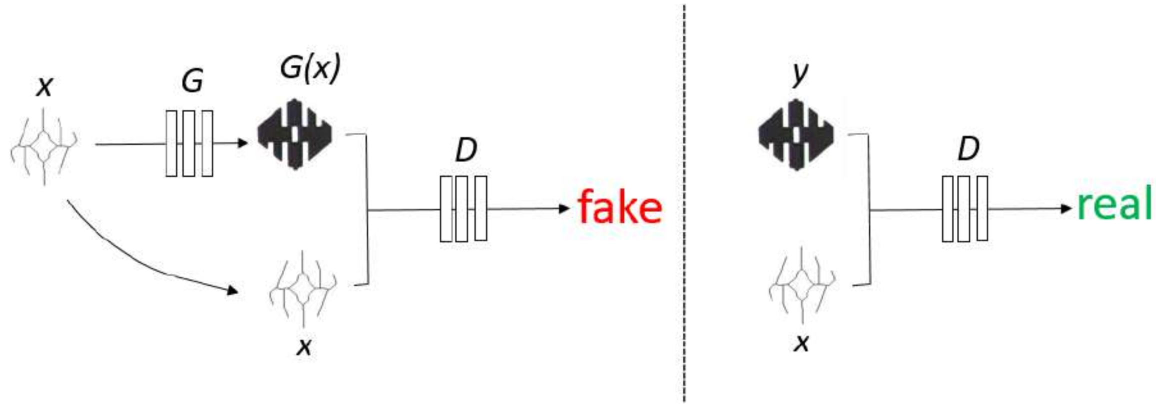


Figure 4.3: Training a conditional GAN to map $skeletons \to motifs$

4.2.1 Objective Function

The objective of a conditional GAN can be expressed as

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))], \qquad ([2]) \qquad (4.2)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. $G^* = \arg\min_G \max_D \mathcal{L}_{\text{cGAN}A_N}(G, D)$

Like basic Generative Adversarial Networks, the discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground-truth output. We also explore this option, using L1 distance,

$$\mathscr{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]. \qquad ([2]) \tag{4.3}$$

Our final objective function is

$$G^* = \arg\min_G \max_D \mathscr{L}_{cGAN}(G,D) + \lambda \mathscr{L}_{L_1}(G). \qquad ([2]) \tag{4.4}$$

4.3 Network Architectures

We adapt our generator and discriminator architectures from those in [2]. Both generator and discriminator use modules of the form convolution-BatchNorm-ReLu [22]. Details of the architecture are discussed below.

4.3.1 U-Net Generator

The generator model takes an image as input, and unlike a traditional GAN model, it does not take a point from the latent space as input. Instead, the source of randomness comes from the use of dropout layers.

- **Input**: Image from source domain
- **Output**: Image in target domain

A U-Net model architecture is used for the generator, instead of the common encoder-decoder model. In an encoder-decoder network [23], the input is passed through a series of layers that progressively downsample, until a bottleneck layer, at which point the process is reversed. Such a network requires that all information flow pass through all the layers, including the bottleneck.

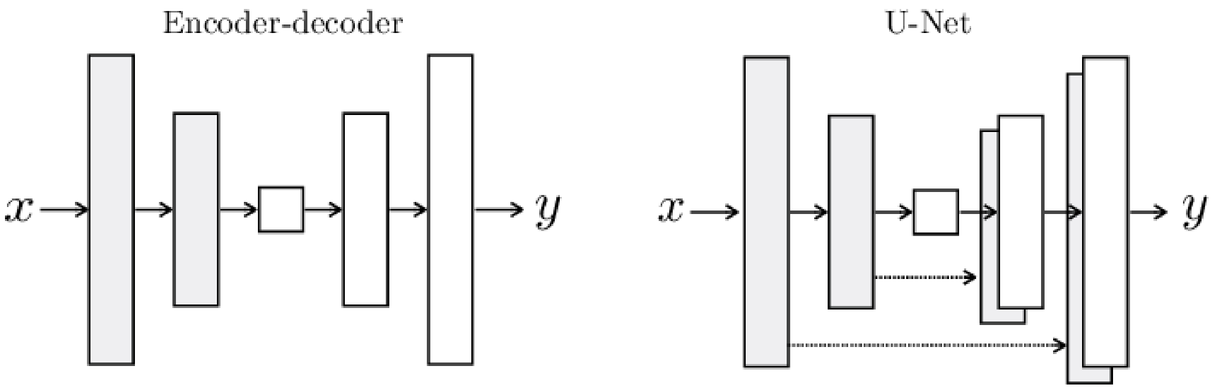


Figure 4.4: Two choices for the architecture of the generator. The "U-Net" is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

For image translation problems, there is a great deal of low-level information shared between the input and output, and it would be desirable to shuttle this information directly across the network. For example, in the case of image colorization, the input and output share the location of prominent edges. To give the generator a means to circumvent the bottleneck for information like this, we add skip connections, following the general shape of a "U-Net" [24]. Specifically, we add skip connections between each layer i and layer $n-i$, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer $n-i$.

4.3.2 PatchGAN Discriminator

The discriminator model takes an image from the source domain and an image from the target domain and predicts the likelihood of whether the image from the target domain is a real or generated version of the source image.

- **Input**: Image from source domain, and Image from the target domain.
- **Output**: Probability that the image from the target domain is a real translation of the source image.

The input to the discriminator model highlights the need to have an image dataset comprised of paired source and target images when training the model.

Unlike the traditional GAN model [13] that uses a deep convolutional neural network to classify images, the pix2pix model uses a PatchGAN [25]. This is a deep convolutional neural network designed to classify patches of an input image as real or fake, rather than the entire image.

The PatchGAN discriminator model is implemented as a deep convolutional neural network, but the number of layers is configured such that the effective receptive field of each output of the network maps to a specific size in the input image. Traditionally, the receptive field refers to the size of the activation map of a single convolutional layer with regards to the input of the layer, the size of the filter, and the size of the stride. The output of the network is a single feature map of real/fake predictions that can be averaged to give a single score.

4.3.3 Optimization and Inference

To optimize our networks, we follow the standard approach from [2]. We alternate between one gradient descent step on D, then one step on G. As suggested in the original GAN paper [13], rather than training G to minimize $\log(1-D(x, G(x, z))$, pix2pix [2] instead train to maximize $\log(D(x, G(x, z))$. In addition, pix2pix divides the objective by 2 while optimizing D, which slows down the rate at which D learns relative to G. We use minibatch SGD and apply the Adam solver [26], with a learning rate of $\alpha = 0.0002$, and momentum parameters $\beta 1 = 0.5$, $\beta 2 = 0.999$. At inference time, we run the generator network in exactly the same manner as during the training phase. This differs from the usual protocol in that we apply dropout at test time, and we apply batch normalization [22] using the statistics of the test batch, rather than aggregated statistics of the training batch. This approach to batch normalization, when the batch size is set to 1, has been termed "instance normalization".

To optimize the learning of our model we used two handcrafted loss functions, one for the generator network and the other for the discriminator network. The training of the discriminator is too fast compared to the generator, therefore the discriminator loss is halved in order to slow down the training process.

$$DiscriminatorLoss = 0.5 * DiscriminatorLoss$$

The generator model is trained using both the adversarial loss for the discriminator model and the L1 or mean absolute pixel difference between the generated translation of the source image and the expected target image. The adversarial loss and the L1 loss are combined into a composite loss function, which is used to update the generator model.

The adversarial loss influences whether the generator model can output images that are plausible in the target domain, whereas the L1 loss regularizes the generator model to output images that are a plausible translation of the source image. As such, the combination of the L1 loss to the adversarial loss is controlled by a new hyperparameter lambda, which is set to 100, e.g. giving 100 times the importance of the L1 loss than the adversarial loss to the generator during training.

$$GeneratorLoss = DiscriminatorLoss + \lambda * L1Loss$$

4.3.3.1 Generator Loss Function

- It is a sigmoid cross entropy loss of the generated images and an array of ones.
- L1 loss is also included, which is mean absolute error between the generated image and the target image.
- This allows the generated image to become structurally similar to the target image.
- The formula to calculate the total generator loss,

$$loss = gan\_loss + \lambda * l1\_loss, where \lambda = 100$$

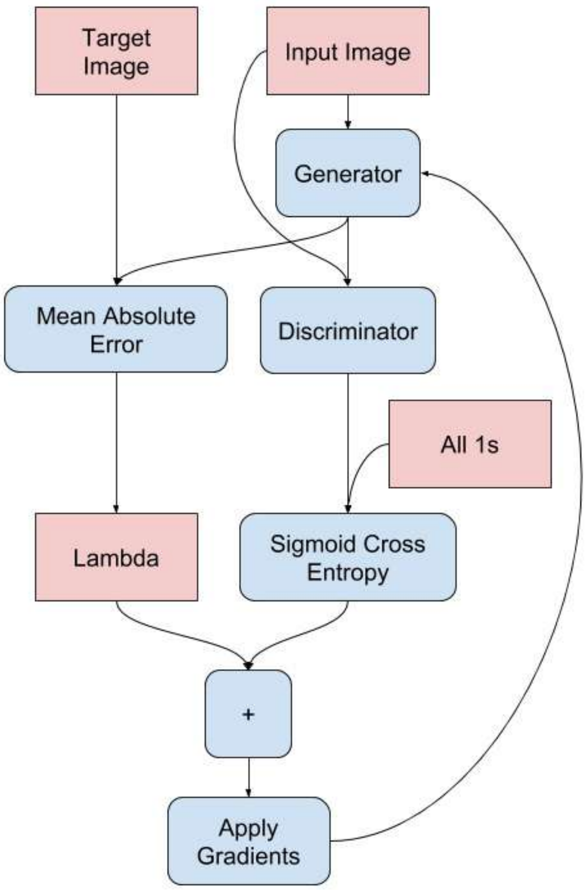


Figure 4.5: Generator loss function architecture

```python
def generator_loss(disc_generated_output, gen_output, target):
  gan_loss = loss_object(tf.ones_like(disc_generated_output), disc_generated_output)

  # mean absolute error
  l1_loss = tf.reduce_mean(tf.abs(target - gen_output))

  total_gen_loss = gan_loss + (LAMBDA * l1_loss)

  return total_gen_loss, gan_loss, l1_loss
```

Listing 1: Python code for loss function of the generator network.

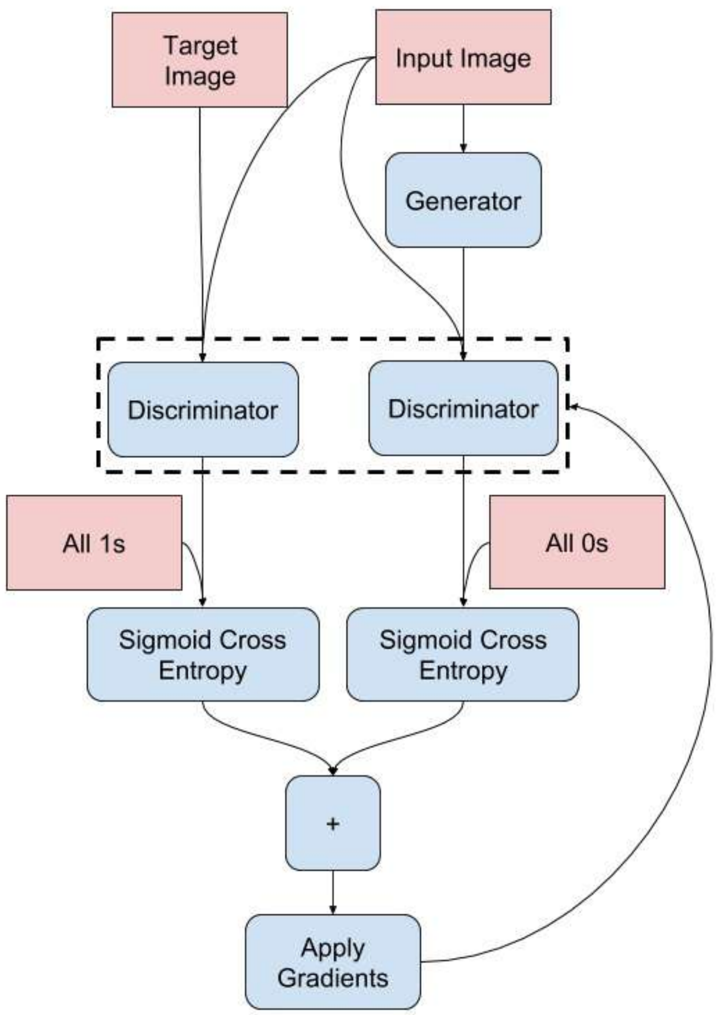


Figure 4.6: Discriminator loss function architecture

4.3.3.2 Discriminator/Adversarial Loss Function

- The discriminator loss function takes 2 inputs; real images, generated images
- real_loss is a sigmoid cross entropy loss of the real images and an array of ones(since these are the real images)
- generated_loss is a sigmoid cross entropy loss of the generated images and an array of zeros(since these are the fake images)
- Then the total_loss is the sum of real_loss and the generated_loss

```python
def discriminator_loss(disc_real_output, disc_generated_output):
  real_loss = loss_object(tf.ones_like(disc_real_output), disc_real_output)

  generated_loss = loss_object(tf.zeros_like(disc_generated_output), disc_generated_output)

  total_disc_loss = real_loss + generated_loss

  return total_disc_loss
```

Listing 2: Python code for loss function of the discriminator network.

Chapter 5

Experiments and Results

This section contains the experimental results of our proposed Jamdani motif generator with Image-to-Image Translation with Conditional Adversarial Networks [2] on the prepared **JamdaniNoksha** dataset. We first start with the experimental setup and then the outcome and comparisons of the model trained with different variations of dataset presented.

5.1 Experimental Setup

The nice thing about pix2pix is that it is not specific to some class i.e. it is generic. It does not require to define any relationship between the two types of images. It makes no assumptions about the relationship and instead learns the objective during training, by comparing the defined inputs and outputs during the training and inferring the objective. This makes pix2pix highly adaptable to a wide variety of situations. That is what drives us to go for a pix2pix based model.

To explore the generality of pix2pix, we followed two approaches.

- *Boundary* → *Image*
- *Sketch* → *Image*

The datasets were divided into training sets, containing 90% of the images and testing sets, containing 10% of the images. Our technique is implemented in TensorFlow-GPU V1.14.0 and cuDNN V9.0. The experiment has been conducted on a Google Colaboratory having CPU: 1× single core hyper-threaded i.e. (1 core, 2 threads) Xeon Processors @2.3Ghz (No Turbo Boost), 45MB Cache, GPU: 1X Tesla K80, having 2496 CUDA cores, compute 3.7, 12GB (11.439GB Usable) GDDR5 VRAM.

5.2 Experimental Result

To develop a pix2pix [2] based model, we followed various approaches. As mentioned above we created five versions of the dataset, we trained our model with these datasets individually. The outputs produced by the models and loss graphs analysis is provided below.

5.2.1 Experimenting with Deep Convolutional GAN

Before diving deep into conditional GAN, to exploit the possibility of generating Jamdani Motifs without providing an input sketch, we prepared an unpaired version of the dataset. We develop an architecture of the basic GAN [13] and train the model with the unpaired dataset. After the test run, it is found that the DCGAN model failed to produce high-resolution image in the target domain. We targeted for at least 256×256. Hence we dropped the idea of going further with DCGAN. Some sample outputs generated with this approach is provided in 5.1.

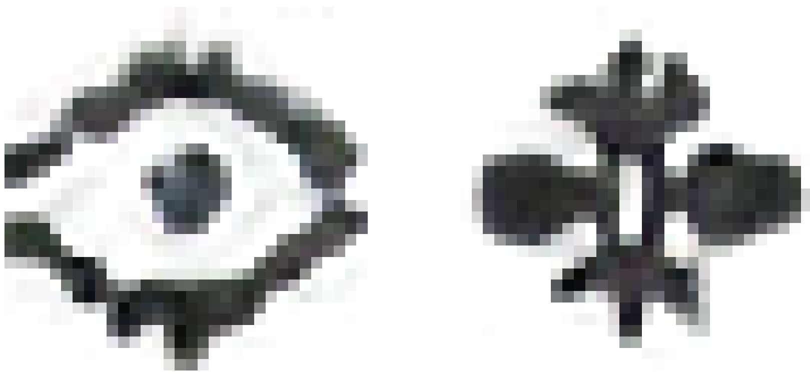
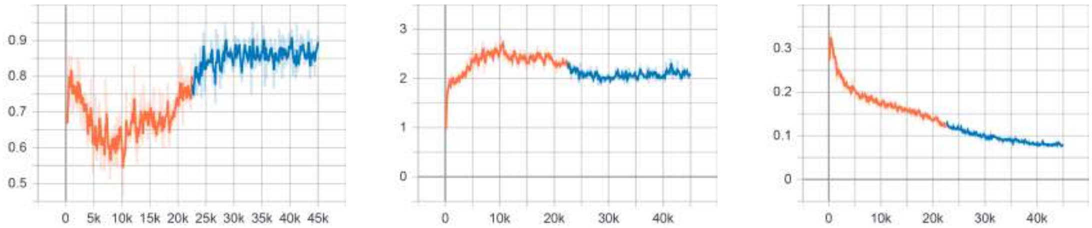


Figure 5.1: Sample output produced by DCGAN [13]. Since it's output is not conditioned based on the input, hence no input sketch is provided. From the outputs it is evident that DCGAN failed to produce high resolution images on the Jamdani domain.

5.2.2 Training with *Boundary* and Output

Initially, we followed the *Boundary* → *Image* [19] approach. We trained the model with the boundary dataset containing 1116 images with batch size 1 for 100 epochs. The training started with a discriminator loss 0.6226, generator loss 0.908 and $L1$ loss i.e. mean absolute pixel difference between the target image and generated image 0.2728 and ended the training with the value 0.8952, 2.104 and 0.975 respectively. See figure 5.2.

In *Boundary* based approach, the output looks sharp and crisp. It's quite obvious because we're providing more information in the input to impose a condition on the output. Sample outputs are shown in figure 5.3.



(a) discriminator_loss_1 (b) generator_loss_GAN (c) L1_loss

Figure 5.2: Loss graph for training on Boundary

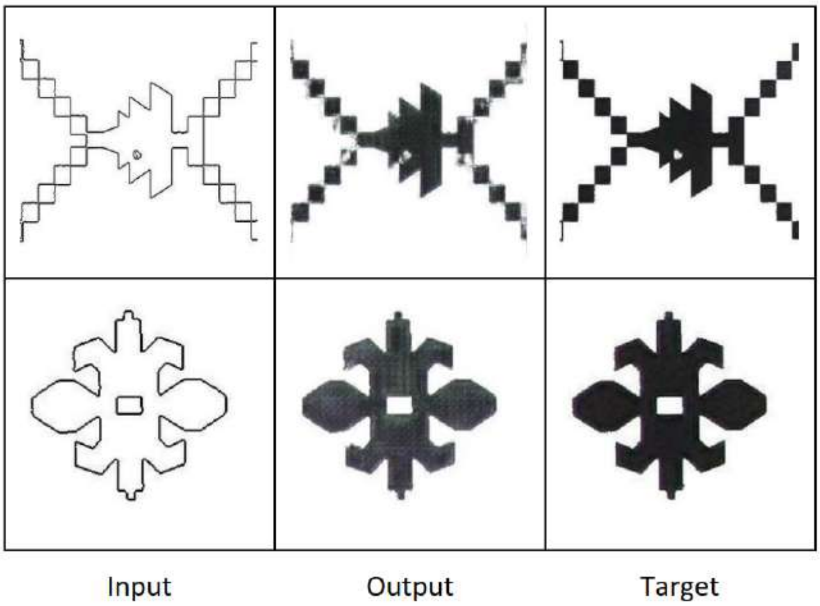
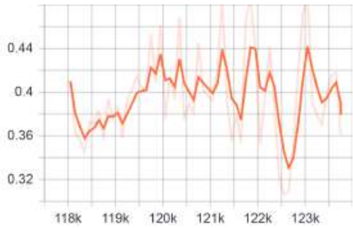


Figure 5.3: Sample output (*middle column*) for model trained on *Jamdani Noksha*'s Boundary Dataset compared to ground truth (*right column*). *Left column* shows input strokes from user.

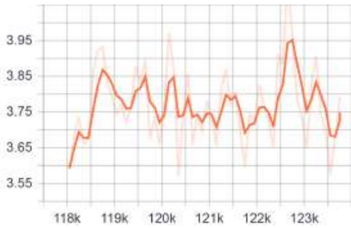
5.2.3 Training with *Enhanced Resolution* and Output

After training the model with the *Boundary* → *Image* approach we tried the model with the *Sketch* → *Image* approach. We prepared the dataset with motif extracted from the 'Traditional Jamdani' book. Our dataset size was 1983. The training started with a discriminator loss 0.4101, generator loss 3.592 and $L1$ loss 0.1043 and ended the training with the values 0.379, 3.748 and 0.1099 respectively. See figure 5.4.

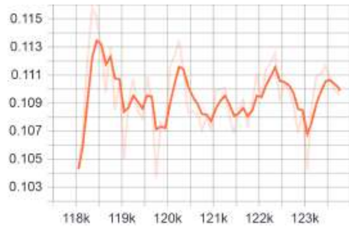
In skeleton-based approaches, we're providing much fewer details in the input. Despite having lower information in the input to condition the output of the generator, generated images look quite good in the target domain. Sample outputs are shown in figure 5.5.



(a) discriminator_loss_1



(b) generator_loss_GAN



(c) L1_loss

Figure 5.4: Loss graph for training on Enhanced Resolution

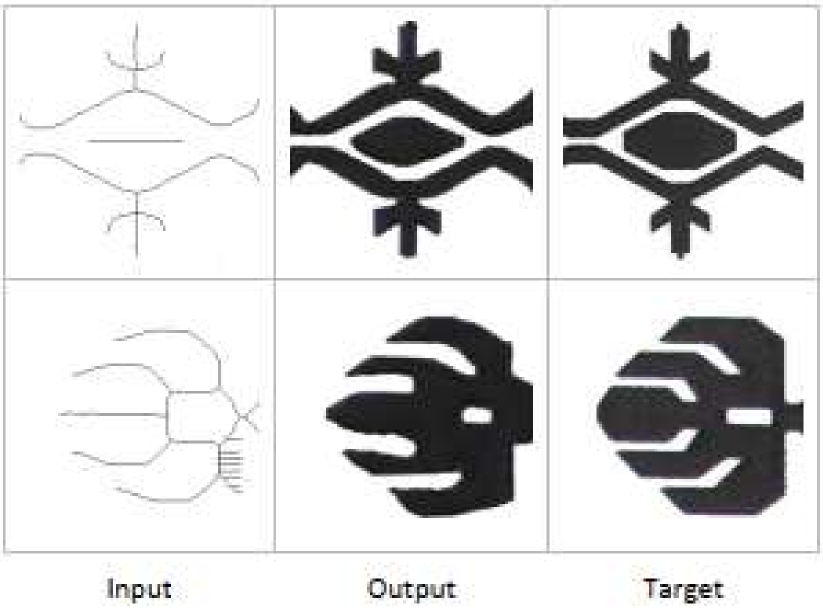


Figure 5.5: Sample output (*middle column*) for model trained on *Jamdani Noksha*'s enhanced version, compared to ground truth (*right column*). *Left column* shows input strokes from user.

After assessing the generated images by both the approaches it is found that *Boundary* $\rightarrow$ *Image* produces more plausible images than *Skeleton* $\rightarrow$ *Image* approach. The loss graphs generated by the tensor board also shows that *Boundary* $\rightarrow$ *Image* based training has lower loss metrics than *Skeleton* $\rightarrow$ *Image*. This is quite obvious since the learning required to produce sharp Jadmdani motif is pretty little while training on *JamdaniNoksha*'s Boundary version. We aim to produce more plausible images concerning the problem domain rather than produce more accurate image-to-image translation i.e. producing images that look like real Jamdani motifs.

Taking these facts into account, as well as considering the ease of taking inputs i.e. boundary/sketch of the images from the user for generating the desired Jamdani motif as a key factor, we opted to go for a skeleton-based Jamdani motif generator model with several variations in our dataset 3.1. The training statistics for different versions of the dataset are shown in table 5.1

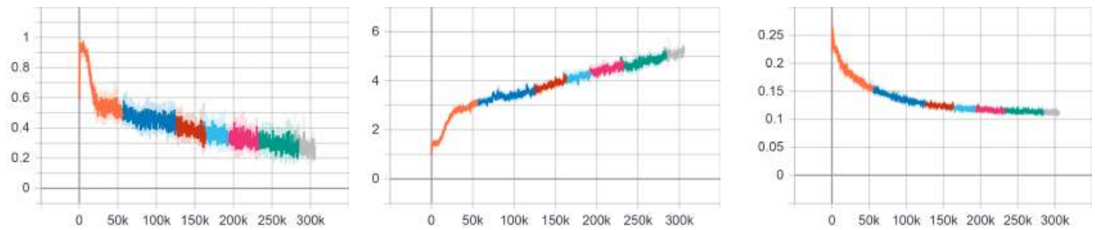
Model trained on dataset	Number of Epochs	D Loss	G Loss	L1 Loss
Boundary	100	0.8952	2.104	0.975
Enhanced Resolution	100	0.379	3.748	0.1099
Reduced Branch	150	0.2205	5.165	0.1095
Sketch	150	-1.362	-0.01646	0.0882
Skeleton	150	-1.385	-1.5×10^{-4}	0.08269

Table 5.1: Loss for model trained on different version of *Jamdani Noksha* dataset

5.2.4 Training with *Reduced Branch* and Output

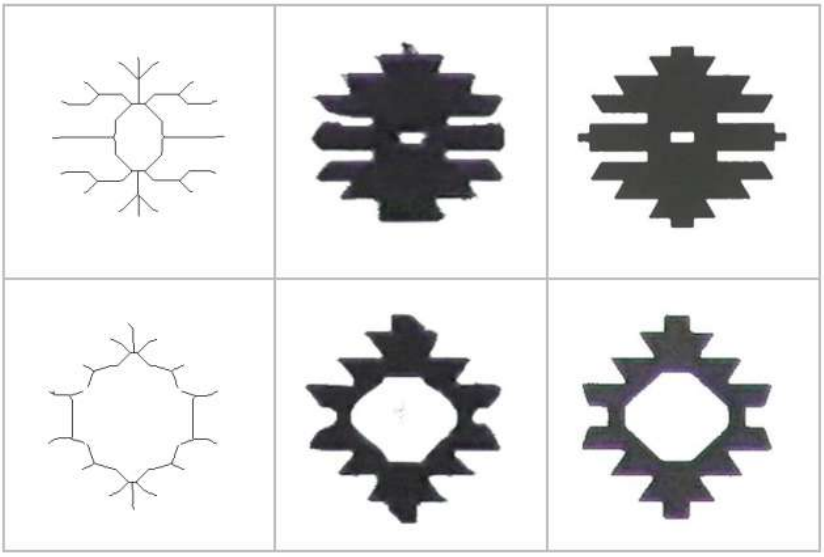
To make the learning of the model more efficient, we decided to chop off the branching of the skeletons. The training started with a discriminator loss 0.9196, generator loss 0.9882 and $L1$ loss 0.2189 and ended the training with the values 0.2205, 5.165 and 0.1095 respectively. See figure 5.6.

In the previous version of the dataset, the input skeletons are consist of complex branching. In a real-world scenario, the user won't draw such complex skeletons. That's why we choose to reduce branching of the skeletons and trained the model with this version of the dataset. Though information in the inputs reduced more, the generated motifs look plausible translation of the target image. Sample outputs are shown in figure 5.7.



(a) discriminator_loss_1 (b) generator_loss_GAN (c) L1_loss

Figure 5.6: Loss graph for training on Reduced Branch



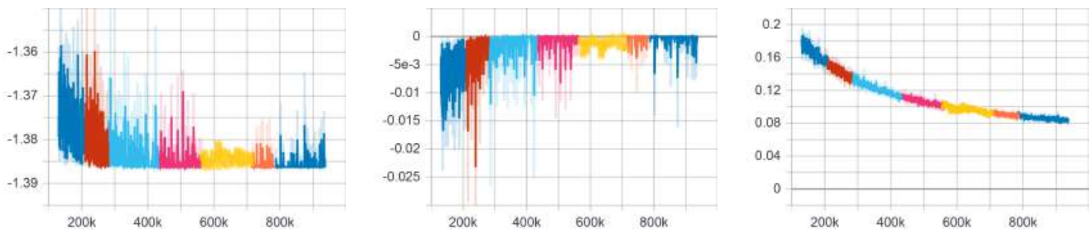
Input Output Target

Figure 5.7: Sample output (*middle column*) for model trained on *Jamdani Noksha*'s reduced branch version, compared to ground truth (*right column*). *Left column* shows input strokes from user.

5.2.5 Training with *Skeleton* and Output

We all know that Generative Adversarial Networks are so much data-hungry. The larger the dataset size, the better the learning of the model. For better learning of the skeleton to motif mapping, we choose to increase the dataset size by applying different data augmentation techniques like flip, rotation, pixel-shifting, etc.

Since the model is trained on almost four times larger dataset compared to the former one, hence the outputs are quite sharp and crisp. The l1 loss value is also minimum among all the model's loss, indicating more accurate pixel to pixel translation. The training started with a discriminator loss -1.379, generator loss -5.886×10^{-3} and L1 loss 0.1734 and ended the training with the values -1.385, -1.5×10^{-4} and 0.08269 respectively. See figure 5.8. Sample outputs are shown in figure 5.9.



(a) discriminator_loss_1 (b) generator_loss_GAN (c) L1_loss

Figure 5.8: Loss graph for training on Skeleton version

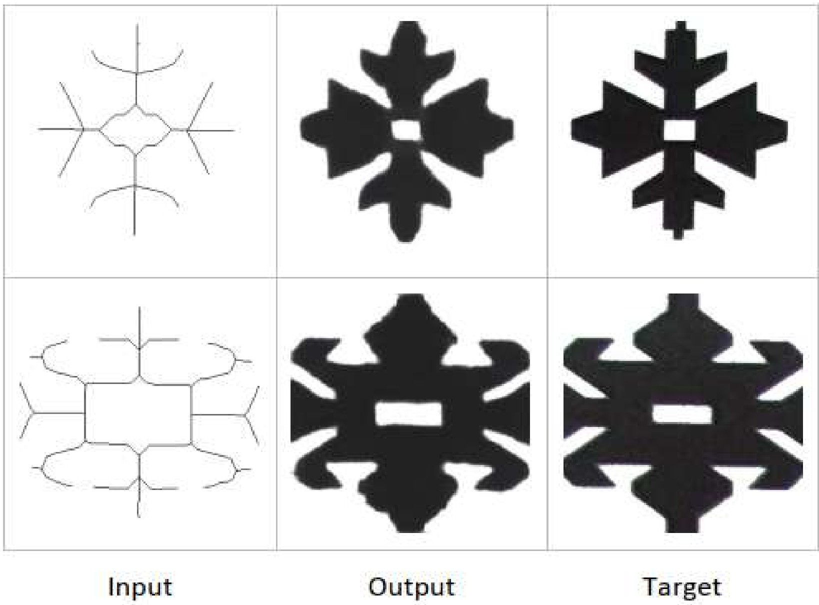
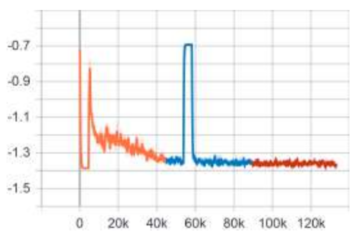


Figure 5.9: Sample output (*middle column*) for model trained on *Jamdani Noksha*'s skeleton version, compared to ground truth (*right column*). *Left column* shows input strokes from user.

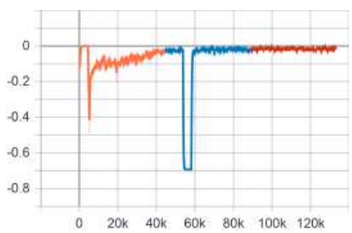
5.2.6 Training with *Sketch* and Output

Finally, we choose to draw handcrafted sketches so that the model gets to learn the actual input sketch to output mapping. We created 250 sketches corresponding to our target output with the digital drawing application and increase the dataset size four times by applying data augmentation. The training started with a discriminator loss -0.7185, generator loss -0.1069, and $L1$ loss 0.2526 and ended the training with the values -1.362, -0.01646, and 0.0882 respectively. See figure 5.10.

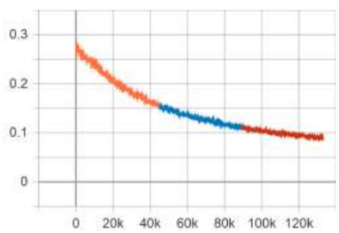
Since the dataset size is small, the generated output produced by the model trained on this version of the dataset is not quite as sharp as the previous ones. Still, it's one step forward in our mission to generate motifs that would come closer to real Jamdani motifs. Sample outputs are shown in figure 5.11.



(a) discriminator_loss_1



(b) generator_loss_GAN



(c) L1_loss

Figure 5.10: Loss graph for training on Sketch version

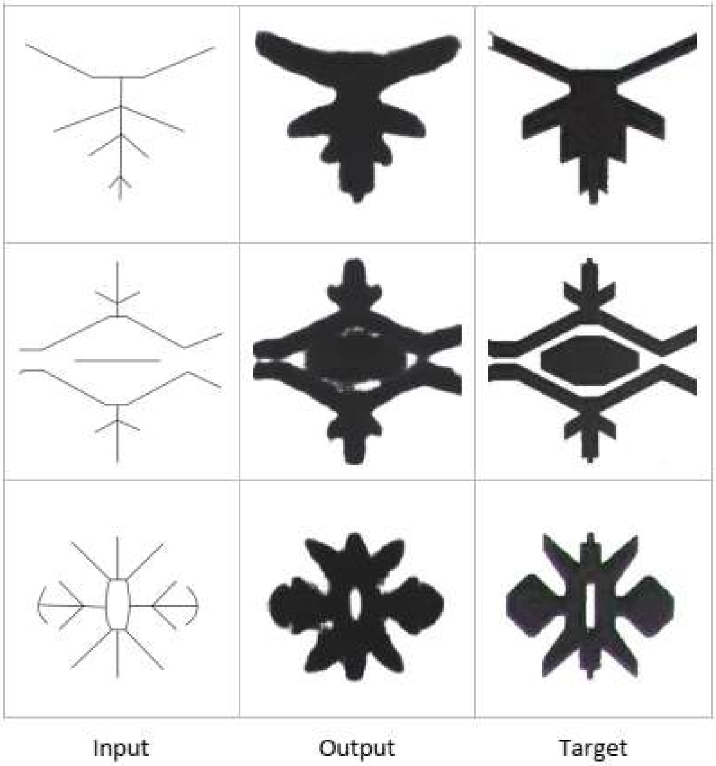


Figure 5.11: Sample output (*middle column*) for model trained on *Jamdani Noksha*'s hand-drawn sketch version, compared to ground truth (*right column*). *Left column* shows input strokes from user.

5.2.7 Loss Graph Analysis

It is well known that the loss graph of Generative Adversarial Network is very non-intuitive. Mostly it happens due to the fact that the generator and discriminator are competing against each other, hence improvement on the one means the higher loss on the other until this other learn better on the received loss, which screws up its competitor. Specifically, the discriminator is learned to provide the loss function for the generator. The two networks compete in adversary, where simultaneous improvements are made to both generator and discriminator models that compete.

We typically seek convergence of a model on a training dataset observed as the minimization of the chosen loss function on the training dataset. In a GAN, an equilibrium between generator and discriminator loss is sought, instead of convergence.

As equation 4.4 depicted that, the discriminator seeks to maximize the average of the log probability of real images and the log of the inverse probability for fake images and the generator seeks to minimize the log of the inverse probability predicted by the discriminator for fake images. This has the effect of encouraging the generator to generate samples that have a low probability of being fake.

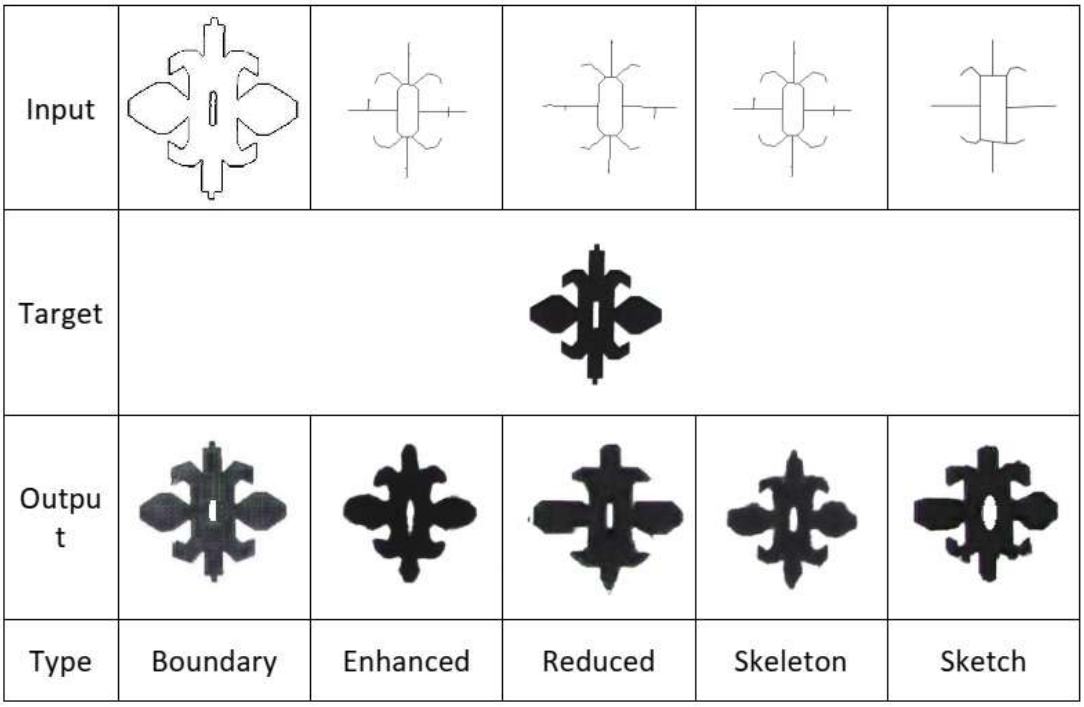


Figure 5.12: Comparison of sample output produced by model trained on different version of dataset.

From the loss graphs 5.2, 5.4, 5.6, 5.8, 5.10 we can see that, as the number of iterations increases the discriminator loss reduces, while the generator loss increases. The $L1$ loss i,e: mean absolute pixel to pixel error tends to get lower as well. These indicate the fact that the discriminator gets better at detecting fake images, while the generator becomes more competent fooling the discriminator. These two adversarial objective drives, both the discriminator and the generator network to reach an equilibrium. As such the generator produces plausible images in the target domain. The $L1$ loss forces the generator to perform more accurate pixel to pixel translation.

The comparison among outputs generated by the model trained on different versions of the dataset is illustrated in figure 5.12

5.3 Web Application using the Generative-Jamdani Engine

We also developed a web application to generate Jamdani motifs in real-time using our model trained on **Jamdani Noksha** dataset. We named this trained model as *Generative-Jamdani Engine*. The application is built using ml5.js library. The library provides access to machine learning algorithms and models in the browser, building on top of *TensorFlow.js* with no other external dependencies.

Our web application provides a user interface to produce Jamdani motifs from the sketch

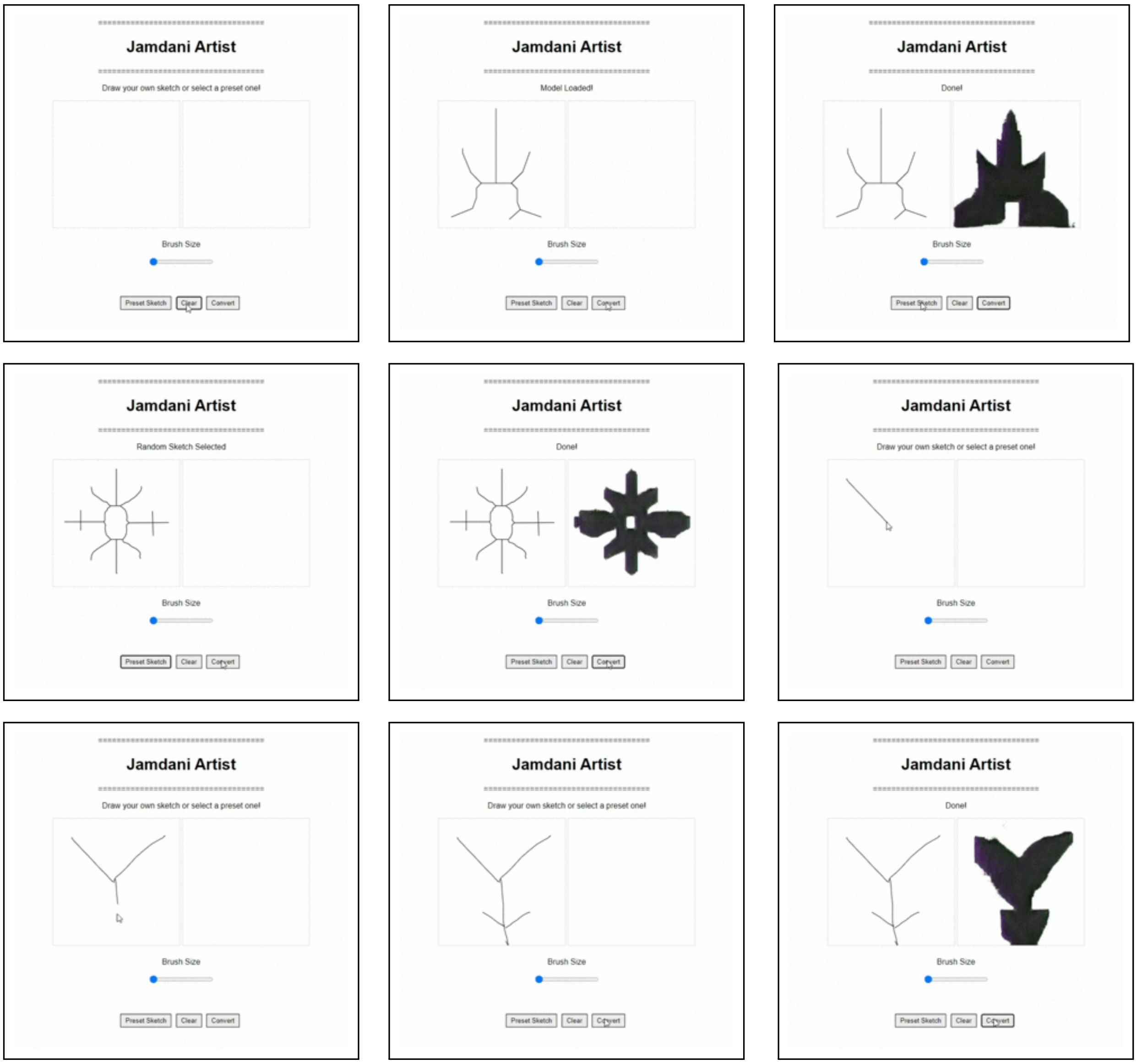


Figure 5.13: Sequential snapshots of the web application. The **Preset Sketch** button loads a random sketch from the library, **Clear** button clears the canvas and the **Convert** button generates Jamdani Motif from the input sketch.

drawn by the user into the canvas, as well as from some preset sketch. Since, it runs on the web browser, the model used at the back-end is made lightweight for low resource consumption. Thus, the generated output is not as crisp as the original model. Yet, the application facilitates the opportunity of experiencing the genius **Generative-Jamdani** in an interactive manner. The application is available at https://raihan-tanvir.github.io/generative-jamdani/. Some snapshot of the application is shown in figure 5.13.

5.4 Evaluation

5.4.1 Human Evaluation

As our work is on image to image translation, the output motif generated by our conditional Generative Adversarial (GAN) Model should be evaluated against the real motif given in each data. Our domain of work is an artistic pattern which is Jamdani Motif. The best way of measuring the accuracy of an artistic phenomenon can be the open eye evaluation by the experts of that domain. As we are working during a time when the COVID-19 virus is dominating our world, it is not easy for us to find experts and take reviews from them. Authenticity in collecting the Jamdani motifs was maintained very carefully as we did not take any design from the internet. The book "Traditional Jamdani Designs" by National Craft Council Bangladesh [1] and the direct observation from Sonargaon were the two sources of the motifs that we are working on. As the authenticity is maintained, we just tried to measure the open-eyed accuracy of the generated motifs against the real ones.

5.4.1.1 Response Collection

We prepared a google form which was used to collect the responses of some individuals. Although any person can tell the similarities or differences between two images with an open eye, we tried to send the form mostly to the persons related to the field of Computer Science for evaluation assuming that they know the basics about accuracy measurement. The form we prepared was completely unbiased as we just wanted the individuals to rate every sample out of 10 in terms of accuracy. A snapshot of our google form is attached here in figure 5.14

We mainly targeted the female individuals for collecting responses as Jamdani motifs are mainly used in saree which is a very celebrated fashion product among the women in Bangladesh. So, women have more knowledge about Jamdani motifs than men. Among the 88 responses of the google form 49 are from female respondents and 39 are from the male respondents. A pie graph describing the ratio of male and female respondents is given in figure 5.15

We selected two of our models which were trained on *Skeleton* and *Sketch* version of our dataset for human evaluation. We have selected the model trained on *Skeleton* version as the version of the dataset has the most number of data and the outputs of this model are quite accurate to us. Again the model trained on *Sketch* version was selected as it is the most realistic approach to generate a motif. Five samples from each model were selected randomly for evaluation. The samples are given in figure 5.16 and 5.17.

Survey on the Accuracy of Motif Generated by the Generative-Jamdani Model

Jamdani is a unique century-old handloom creation embedded in Bangladeshi history having a great socio-economic impact. The exclusive geometric motifs woven on the fabric are the most attractive part of this craftsmanship having a remarkable influence on textile and fine art. In this project, we have developed a technique based on Generative Adversarial Network that can learn to generate entirely new Jamdani patterns from a collection of Jamdani motifs that we assembled, the newly formed motifs can mimic the appearance of the original designs. Users can input the skeleton of a desired pattern in terms of rough strokes and our system finalizes the input by generating the complete motif which follows the geometric structure of real Jamdani ones.

We would like to request you all to view and compare Jamdani motifs artificially generated by our system with the real ones in order to measure the visual accuracy of our proposed system.

* Required

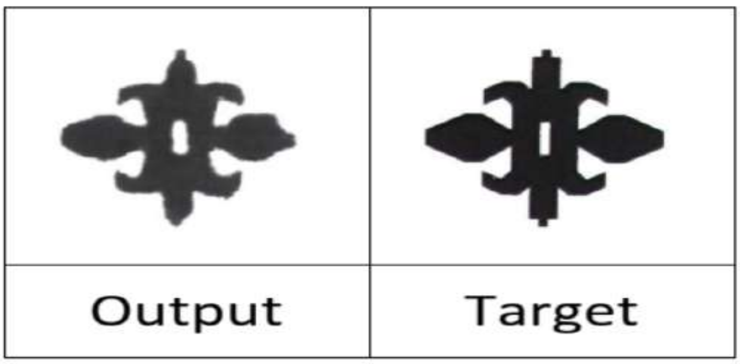
Email address *

Your email

Gender *

- Female
- Male
- Other

How accurate the generated motif (on left-labeled as Output) compared to the real motif (on right-labeled as Target)? *



	1	2	3	4	5	6	7	8	9	10	
Not accurate at all	○	○	○	○	○	○	○	○	○	○	Very accurate

Figure 5.14: A snapshot of the form used to collect the response for human evaluation.

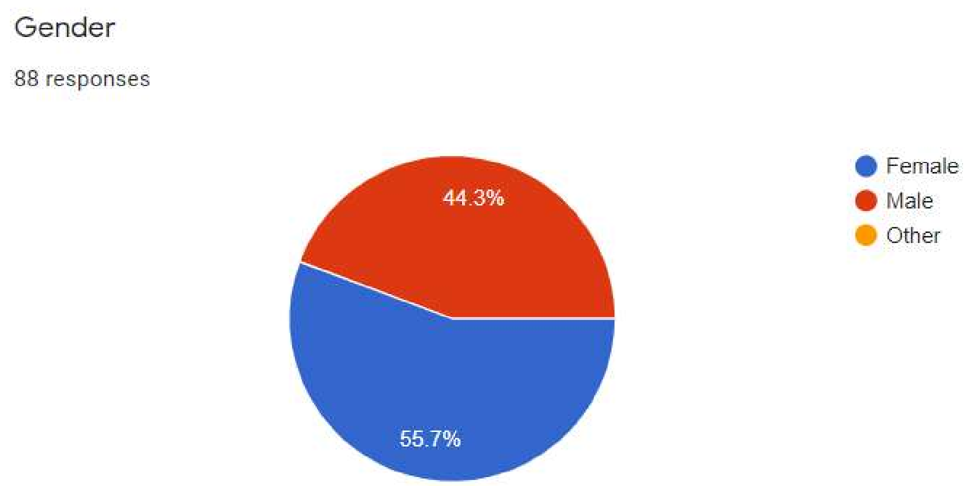


Figure 5.15: Pie graph describing the male female ratio of the respondents generated on the google form summary.

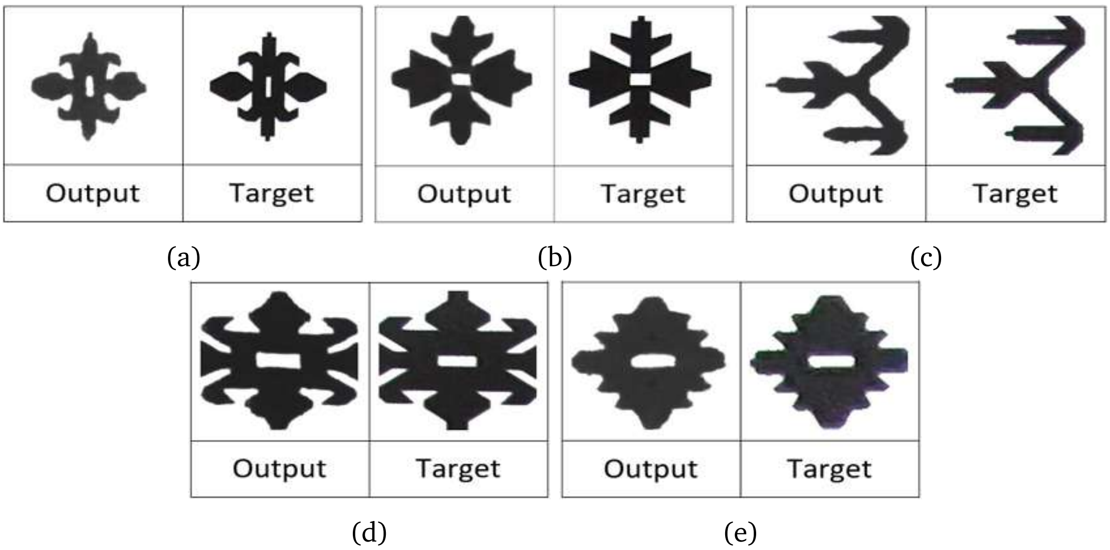


Figure 5.16: Samples randomly selected from the output of the model trained on *Skeleton* version of our dataset for evaluation. In each pair from left to right, the output generated by the model and the corresponding real motif is shown respectively.

Average score of each individual sample out of 10 for the model trained on *Skeleton* version of our dataset is given in table 5.2

The average score given by the individuals for this model is 7.45 out of 10. So, the accuracy is 74.5%.

Average score of each individual sample out of 10 for the model tarined on *Sketch* version of our dataset is given in table 5.3

No. of Sample	Score (Out of 10)
5.16a	6.92
5.16b	7.17
5.16c	7.25
5.16d	8.07
5.16e	7.83

Table 5.2: Average score of each individual samples of fig 5.16. The score is rated on 10.0

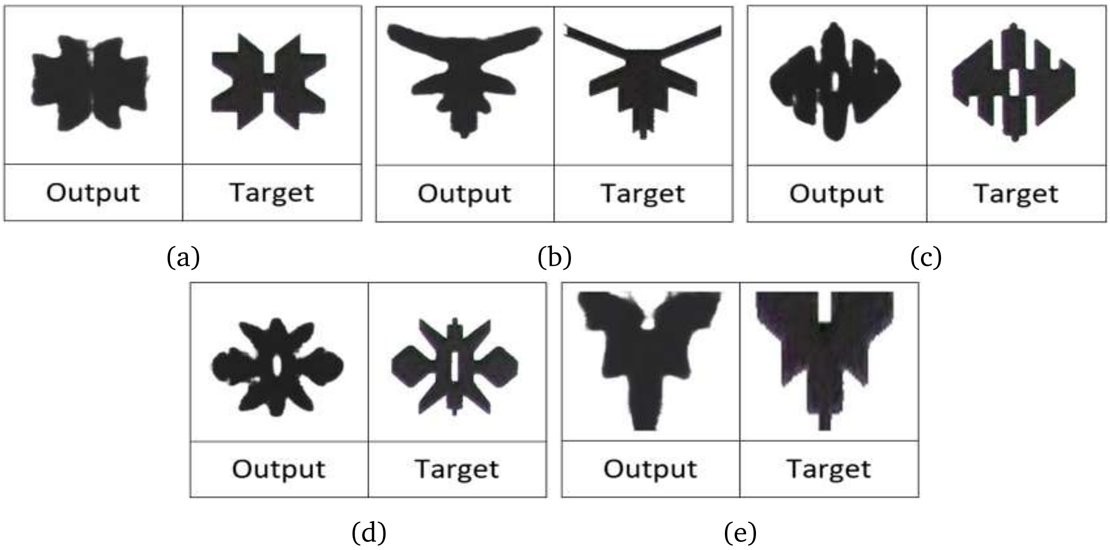


Figure 5.17: Samples randomly selected from the output of the model trained on *Sketch* version of our dataset for evaluation. In each pair from left to right, the output generated by the model and the corresponding real motif is shown respectively.

No. of Sample	Score (Out of 10)
5.17a	5.01
5.17b	6.44
5.17c	5.56
5.17d	5.89
5.17e	5.97

Table 5.3: Average score of each individual samples of fig 5.17. The score is rated on 10.0

The average score given by the individuals for this model is 5.77 out of 10. So, the accuracy is 57.7%.

5.4.1.2 Result Analysis

Generative Adversarial Network (GAN) is a data-hungry architecture. Its performance is proportional to the size of the dataset. The *Skeleton* version is the largest version of our dataset consisting of 7932 data. So the output is quite good in terms of other trained models. The human evaluation also projects this outcome. The model trained on *Skeleton* version of our dataset has got the accuracy of 74.5% which is very satisfactory.

On the other hand, The *Sketch* version is the smallest version of our dataset which has only 910 data samples. This is the most realistic approach among other versions and the model which should be considered as the ultimate model for *sketch* → *motif* generation. The insufficiency of the data is the main reason behind the low accuracy of 57.7% for the model trained on the *sketch* version of our dataset.

So, analyzing the results of the evaluation we can come to a conclusion that the larger the number of data will be the more visually accurate the produced output will seem. This fact becomes evident if we consider the output of the model trained on the *Skeleton* version, and compare it with the output of the model trained on the *sketch* version of our dataset. If the number of data in the *sketch* version of the dataset is increased then it will be possible to get more accurate output while using the system in real life.

Chapter 6

Future Work and Conclusion

6.1 Limitations And Future Work

As our research work is a trailblazer in the area of computer vision, there were remarkable challenges we had to overcome. At the same time, this pioneering research has potentials that can be driven and explored in ways to unlock opportunities in the long run. The research we have carried out so far is not free of limitations. We move forward with a view that our unique data set which represents our cultural heritage and the cutting-edge technology of GAN will intersect and bring out a new dimension in both the areas of computer vision-related research and the Century-old tradition of Bangladesh. We hope our contribution adds to the industry and economy at the greater end. To the best of our knowledge, the research that we have initiated is a unique and one of a kind contribution to this field and it holds the prospective to go further. But as said earlier that opportunities are not without limitation. We now try to reflect on the huddles and try finding out ways to solve them. Below the burning barriers are discussed :

- The biggest constraint is the insufficiency of data. To have a more structured and efficient model more data is needed. But there is a shortage of data. Specially if we look at the *Sketch* version of our data set we will see it has only 910 samples.
- Time is another blocked factor. Sketching the hand drawn skeletons is a time consuming process and hence producing more data of that kind demands more time.

We have to keep in mind that these are the initial obstacles we are facing right now. New challenges will arise as we inspect more options and try adding new features to our system. Nevertheless, there lie numerous scopes to explore and ideas for improvement which are analyzed and stated below:

- The appearance of the outputs are quite good compared to the quantity of input data. But if we can collect more data it would be easier to make the outputs more realistic and flawless.
- Jamdani is the proud representative of the century-old handloom of Bangladesh. But nowadays several unethical fake factories have been established producing machine-made Jamdani sarees and labeling and selling them as original DHAKAI JAMDANI at high prices. Without proper knowledge and expertise, it is hard for general people to distinguish between the original and the fake ones. There is no technical solution to identify authentic Jamdani designs. This is not only threatening to Jamdani industry of Bangladesh but adding more to the misery of helpless weavers of the country. To solve this issue, a model can be implemented which will classify the genuine Jamdani design from the fake ones, and hence the identification of the original Jamdani design can be done in a digitized form. So building a system to classify the original Jamdani motifs/ patterns from the fake designs and other non-Jamdani designs can be great future work. The Jamdani Noksha Dataset along with other handloom datasets such as the ref..... can be used to serve the intended purpose.
- We are working with the Jamdani Motifs which are the building blocks of a complete design. Generating larger designs/ patterns by combining the newly generated motifs can help the industry to prepare designs maintaining the authenticity of the motifs.
- We have only generating the motifs here. But as stated earlier the motifs can be used in making non-textile products as well. So a domain transfer feature can be added as an extension of our research work. By using that feature anyone creating a new motifs/ design can try the motif on different products. This will be a great way of visualizing the appearance of the motifs on different products through trials. See figure 6.1
- As stated earlier the Jamdani motifs are the geometric expression of various natural elements found on the banks of Shitalakshya river. The contemporary motifs these days are made taking inspiration from elements beyond that boarder. Anything artistic can be made into a geomitric Jamdani Motif. And Hence a system which can convert different objects into an geometric pattern that resembles the hand loomed Jamdani designs can be made in the future trailed from the research performed today.

With proper strategy, the system we have created can become a dedicated tool for generating and using Jandani Motifs to serve all sorts of creative purpose. If this tool can cross the international border the Jandani Industry of Bangladesh along with the weavers here will get an international exposure that is sure to help this industry grow and sustain.

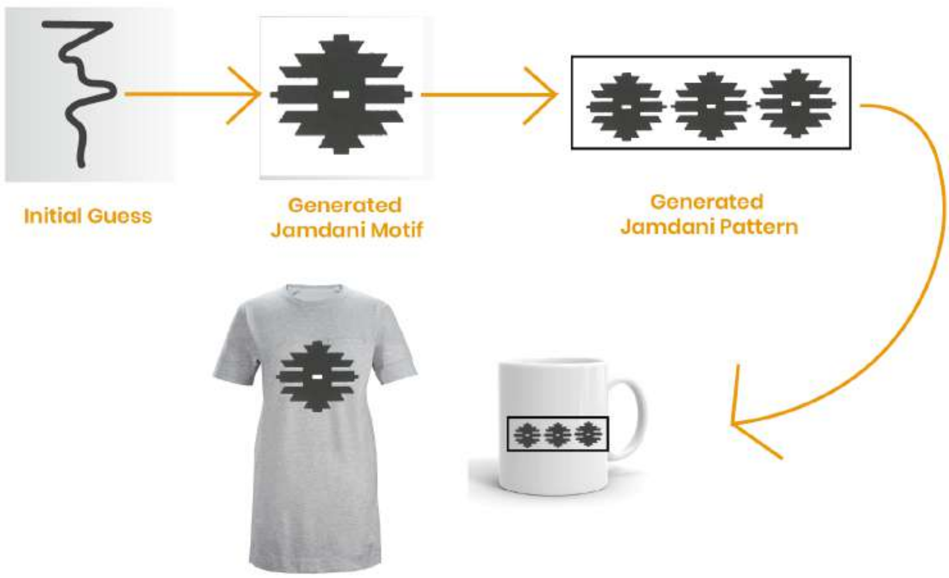


Figure 6.1: Jamdani patterns on different products.

6.2 Conclusion

Jamdani is treated as the most advanced hand-weaving technique in the world. But the charm lies in its remarkable geometric motifs. Going forward, we see that if the next generation isn't enthused by handloom business prospects leaving with a handful of weavers. So in order to contribute to the continuation of the Jamdani tradition of Bangladesh we believe the most reliable solution is to digitize this industry in an artificially intelligent manner. In this thesis work, we presented an approach to generate Jamdani motifs based on initial guesses from users' strokes. Our motto is to preserve the thousand years old heritage of Bangladesh with the help the today's groundbreaking technology. This will not only ensure the preservation of the most complicated and oldest motifs but will also help the Artists involved in sector work in a more efficient and easier way. We believe our research will serve as a foundation for more dynamic and creative work in this domain that will assure the development of the Jamdani industry by helping weavers sustain in an enriched economic condition. The core identity of Jamdani tradition has to be kept unchanged. Our **Jamdani Noksha** Dataset is not only a collection of motifs of different eras but a rich source for the researchers enabling them to unfold the unknown. We move forward with a view that our unique dataset which represents our culture and the cutting-edge technology of GAN will intersect and bring out a new dimension in both the areas of research and tradition. We are hopeful the fully developed system of ours will bring a change, and create a new sense of revival.

References

[1] C. S. Shaha, *Traditional Jamdani Design*. Bangladesh National Museum, March 1, 2018.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[4] D. Yoo, N. Kim, S. Park, A. S. Paek, and I.-S. Kweon, "Pixel-level domain transfer," in *ECCV*, 2016.

[5] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[6] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," *CoRR*, vol. abs/1801.02753, 2018.

[7] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *CoRR*, vol. abs/1604.07379, 2016.

[8] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Sketch-to-image generation using deep contextual completion," 11 2017.

[9] I. Salian, "Gaugan turns doodles into stunning, realistic landscapes: Nvidia blog." https://blogs.nvidia.com/blog/2019/03/18/gaugan-photorealistic-landscapes-nvidia-research, Oct 2019.

[10] R. K. Bhattacharjee, M. Nandi, and A. Jha, "Handloom design generation using deep neural networks." https://github.com/rajatkb/Handloom-Design-Generation-using-Deep-Neural-Networks, 2018.

[11] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *CoRR*, vol. abs/1703.05192, 2017.

[12] *Jamdani Festival*, 2019 (accessed August 23, 2020). https://jamdanifestival.com.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[14] "Traditional art of jamdani weaving." https://ich.unesco.org/en/RL/traditional-art-of-jamdani-weaving-008, 2013 (accessed September 5, 2020).

[15] Z. Wan, B. Zhang, D. Chen, P. Zhang, D. Chen, J. Liao, and F. Wen, "Old photo restoration via deep latent space translation," *arXiv preprint arXiv:2009.07047*, 2020.

[16] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018.

[17] L. Yang, D. Oyen, and B. Wohlberg, "A novel algorithm for skeleton extraction from images using topological graph analysis," 06 2019.

[18] J. Ma, T. Yurevich, and V. Kanapelka, "Image skeletonization based on combination of one- and two-sub-iterations models," *Informatics*, vol. 17, pp. 25–35, 06 2020.

[19] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved sobel edge detection," 08 2010.

[20] H. Yang, Y. Li, X. Yan, and F. Cao, "Contourgan: Image contour detection with generative adversarial network," *Knowledge-Based Systems*, vol. 164, 10 2018.

[21] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[25] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," 2016.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Thursday 6th July, 2023 at 7:23pm.