North East University Bangladesh
Department of Computer Science and Engineering

2D Hand Drawn Shape Completion

By
Eftakhar Ahmed Arnob
Reg. No: 190103020028
Ehtimum Rashed Choudhury
Reg. No: 190103020026
BSc(Engg) in CSE
4th year 2nd semester

Supervised By
Khadem Mohammad Asif-uz-zaman
Lecturer
Department of Computer Science and Engineering

25th January, 2023

2D Hand Drawn Shape Completion

A Thesis submitted to the Department of Computer Science and Engineering, North East University Bangladesh, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

By

Eftakhar Ahmed Arnob
Reg. No: 190103020028
Ehtimum Rashed Choudhury
Reg. No: 190103020026
BSc(Engg) in CSE
4th year 2nd semester

Supervised By

Khadem Mohammad Asif-uz-zaman
Lecturer
Department of Computer Science and Engineering

25th January, 2023

Recommendation Letter from Thesis Supervisor

These Students, *Eftakhar Ahmed Arnob & Ehtimum Rashed Choudhury,* whose thesis entitled *"2D Hand Drawn Shape Completion",* is under my supervision and agrees to submit for examination.

Signature of the Supervisor :

Khadem Mohammad Asif-uz-zaman
Lecturer
Department of Computer Science and Engineering
North East University Bangladesh

Qualification Form of BSc(Engg) Degree

Student Name 1 : Eftakhar Ahmed Arnob
Student Name 2 : Ehtimum Rashed Choudhury

Thesis Title : 2D Hand Drawn Shape Completion

This is to certify that the thesis is submitted by the student named above in January, 2022. It is qualified and approved by the following persons and committee.

Head of the Dept.

Tasnim Zahan
Head of The Department
Department of CSE
North East University Bangladesh

Supervisor

Khadem Mohammad Asif-uz-zaman
Lecturer
Department of CSE
North East University Bangladesh

Abstract

In the field of computer vision, image recognition and completion are fundamental problems. Conventional methods rely on a complete image as input, but when working with partially missing shapes, completing those shapes can be a challenging task. Furthermore, real-world hand-drawn sketches often contain missing parts, and the randomly drawn nature of these sketches can make it difficult to take them as input. Previous approaches have employed Generative Adversarial Networks (GANs) to complete incomplete shapes by generating a new shape of the same class and post-processing the generated image to patch it with the input. However, these methods often fail to match the trajectory of the initial shape. To address these limitations, we decided to use an "Encoder-Decoder Network". This type of network is able to follow the trajectory of the initial shape and generate the missing part of it by calculating and minimizing the loss between the "generated complete shape" and "true complete shape". This approach enables the "Encoder-Decoder Network" to develop a human-like understanding to generate the missing parts of a shape or sketch.

Keywords: Hand-drawn, Non-standard Shape, Partial Shape, Shape Completion, Encoder, Decoder, Loss, True shape, Generated shape.

Table of Contents

Table of Figures

Chapter 1

INTRODUCTION

In the field of machine learning, image completion is a technique that generates missing data from input images that often contain partial or incomplete information. One of the main challenges in this area is the prediction and generation of complete shapes from incomplete data. So this is a difficult problem to solve in machine learning.

1.1 Our Goals

- Utilize partially hand-drawn images to generate complete images
- Ensure consistency between the generated image and the partial image
- Investigate the ability of the model to generate output from input that has been rotated at various angles, specifically the ability to maintain rotation angle of the input in the generated output.

During the process of learning to write, children often begin by drawing random shapes, many of which are incomplete. A machine learning system that is capable of detecting these incomplete shapes and completing them could provide valuable guidance for children as they learn to properly draw shapes. Furthermore, advanced applications of this technology could be used to assist self-driving cars in identifying road signs that are not fully visible.

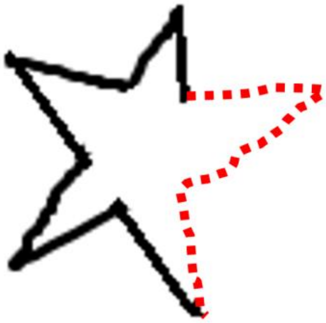


Figure 1: Partial Hand Drawn Star (Our system need to complete the red dotted part)

The key idea of our research is to working with non-standard hand-drawn partial sketches and to complete them.

Non-standard shape: Real hand drawing vary person to person as same shape can be represented in different size, shape and angle. For example, a shape can be drawn in small size in a big paper. Generally, available completion systems work with the image with some missing parts. So, the size, angle, shape remains standard. They are not enough intelligent to catch the variation and to complete them. Some examples of non-standard shapes are given below:

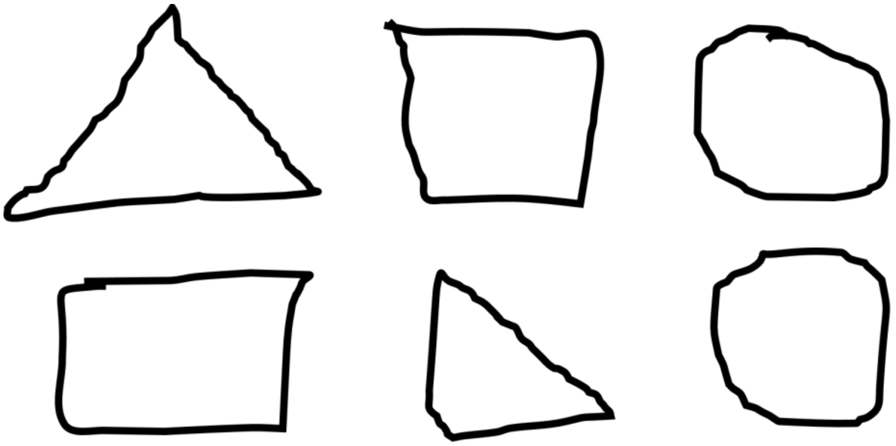


Figure 2: Example of non-standard shapes. There are 2 variants for each 3 shapes in different form factor.

Generative models are a powerful approach to learning the underlying data distribution of a training set and generating new data points with variations. However, upon conducting background research, we discovered that a specific type of generative model, the Generative Adversarial Network (GAN), is not well-suited to our research goal of completing incomplete shapes. GANs can be trained on complete shapes to generate new variations of the shape, but they are unable to properly keep consistency of generated completed image with the partially missing input image. As a result, we decided to use an Encoder-Decoder type neural network, as this architecture has been found to be effective in completing missing parts of shapes.

We will discuss about our background study, proposed methodology, dataset, experiments and results progress and future study in the following chapters. In chapter 2 some of the papers we read in this research which is related and important to our work are presented. In chapter 3, we discussed our methodology and explained all the techniques we are using. In chapter 4 we briefly explained our dataset making procedure. In chapter 5 we included a brief discussion on our experiments. And finally chapter 6 we shown our final results and then on chapter 7 is the conclusion of this research work.

Chapter 2

BACKGROUND STUDY

There are many research papers available on image completion. Some of these papers work with generative models and other work with prior based model. But, none of them worked with real hand-drawn shapes. There are some difficult challenges in working with real hand-drawn shapes which we discussed before. So, after reading all these papers we decided to work with real hand-drawn sketches. Some of the papers I read for this research are described below:

2.1 Joint sketch completion and recognition with GAN [1]:

The SketchGAN architecture shown here, takes the image x as input and then iteratively refines it by regenerating and meshing it with the previous input (if applicable) and generates a more completed image output through multiple stages. At each stage, the generator G takes input x and outputs from previous stage and produces a better completed sketch output G(x) (3 stages are demonstrated in the figure above). The last output G(x) is then judged by the discriminator network D to be real or fake. After that, a sketch recognition network is applied to recognize the sketch.

The Cascade Strategy: A cascade method is used in this research to further refine the contour closure of the sketch. Each cascade stage is built upon a conditional GAN, which does not share network parameters with different cascade stages.

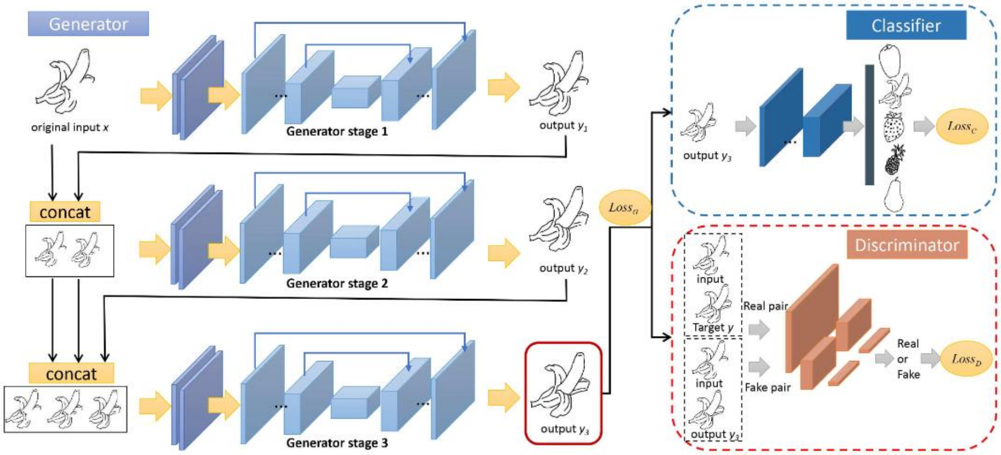


Figure 3: Three Stage SketchGAN Network.

It uses the Sketchy dataset- the first large-scale collection of sketch-photo. It contains total 75471 sketches of 125 classes.

It can complete a partially completed shape, but only from the corrupted sketch of the shape. That means it cannot work with the non-standard hand-drawn shapes.

2.2 2D Shape completion with shape priors [2]:

As the name suggests, it is a research paper based on 2 dimensional shape's contour completion with the shape priors that is detected from a real image which is missing some part of the image. The approach uses two types of object contour or outline datasets, GSD (Generic Shape Database) and CSD (Class-specified Shape Database).

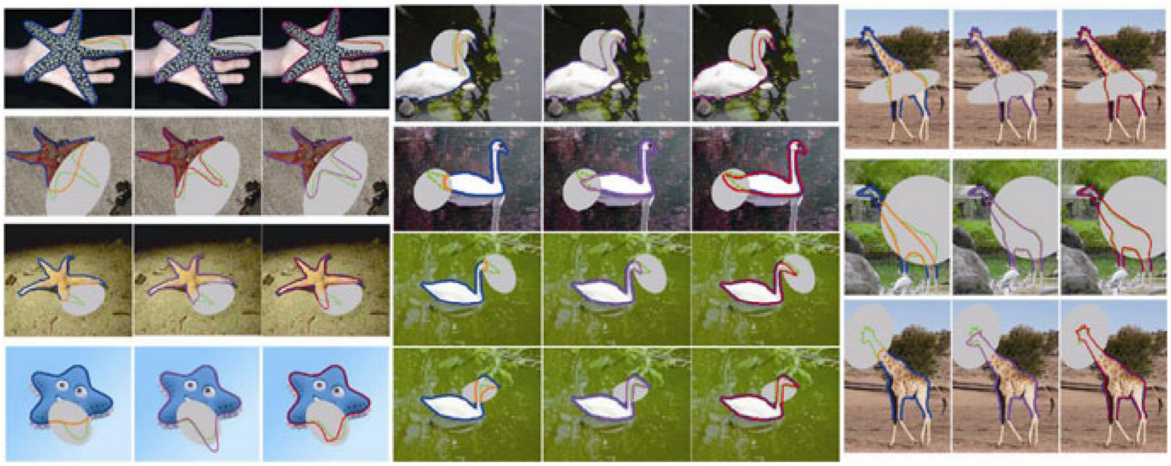


Figure 4: Contour Completion Results.

The GSD contains only one data for each class of shapes and the CSD contains multiple data for a single class. They used a Bayesian model to complete the missing contour using the previously specified dataset.

2.3 Multiclass sketch-to-image translation [3]:

A GAN-based sketch-to-image translation method that helps user to draw sketch with a guiding system. When the user starts to draw sketch the network will try to suggest the user plausible completions, and shows a corresponding synthesized image to the user. Resulting a feedback loop, where the user can edit their sketch based on the network's recommendations, visualizing both the completed shape and final rendered image while they draw.

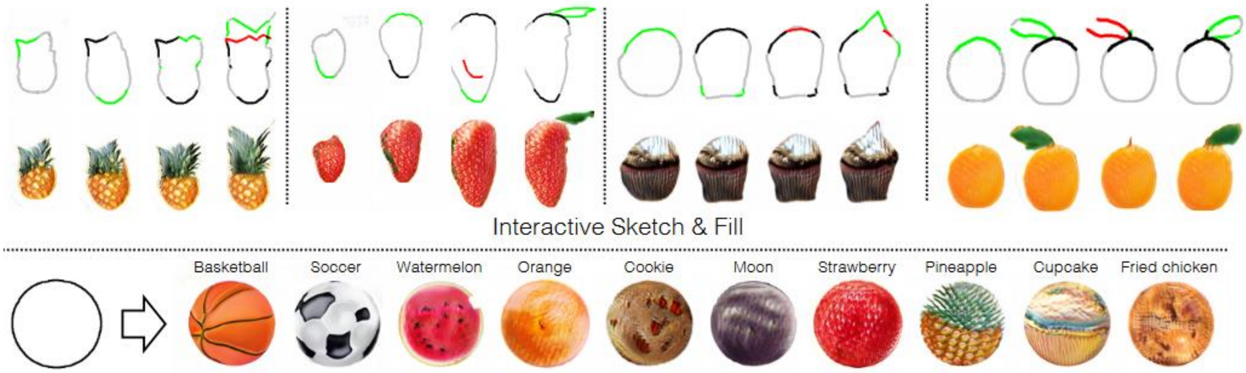


Figure 5: Sketch To Image Translation.

They introduce a gating-based approach for class conditioning, which allows us to generate distinct classes without feature mixing, from a single generator network.

2.4 RL-GAN-Net: A Reinforcement Learning Agent Controlled GAN Network for Real-Time Point Cloud Shape Completion [4]:

Their framework is applied to point cloud shape completion that converts noisy, partial point cloud data into a high-fidelity completed shape by controlling the GAN. While a GAN is unstable and hard to train, the solution they found is by (1) training the GAN on the latent space representation whose dimension is reduced compared to the raw point cloud input and (2) using an RL agent to find the correct input to the GAN to generate the latent space representation of the shape that best fits the current input of incomplete point cloud. The suggested pipeline robustly completes point cloud with large missing regions. Additionally, their pipelines can be used to enhance the classification accuracy of point cloud with missing data.

It uses ShapeNet point cloud dataset which is a large-scale dataset of shapes with 51, 000 objects in 55 categories.

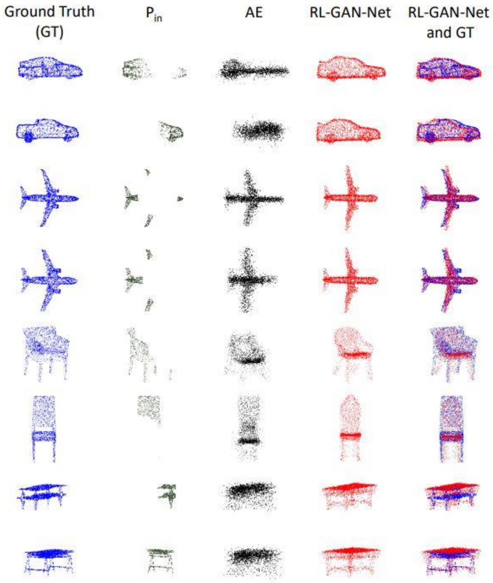


Figure 6: Result of RL controlled GAN network

2.5 Globally and locally consistence image completion [5]:

Their framework is applied to point cloud shape completion that converts noisy, partial point cloud data into a high-fidelity completed shape by controlling the GAN. While a GAN is unstable and hard to train, the solution they found is by (1) training the GAN on the latent space representation whose dimension is reduced compared to the raw point cloud input and (2) using an RL agent to find the correct input to the GAN to generate the latent space representation of the shape that best fits the current input of incomplete point cloud.

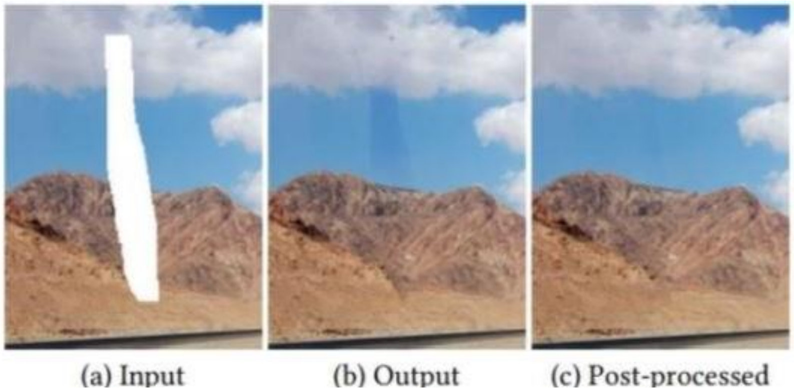


Figure 7: Image completion result

The suggested pipeline robustly completes point cloud with large missing regions. Additionally, their pipelines can be used to enhance the classification accuracy of point cloud with missing data.

The model uses 8,097,967 training images taken from the Places2 dataset. The result shows it can produce better result in objects but sometimes failing to create human face or animal as it completes images based on the area near the missing part.

2.6 Generative Image Inpainting with Contextual Attention [6]:

The propose method is a new deep generative model-based approach which can not only synthesize novel image structures but also explicitly utilize surrounding image features as references during network training to make better predictions. This particular architecture consists of two network one is Coarse network where this network will generate course part in the missing region of the image like a blur version and the refinement network which will take course network's output as input and generates refined parts in the missing region. This approach will result in a much more detail while in painting.

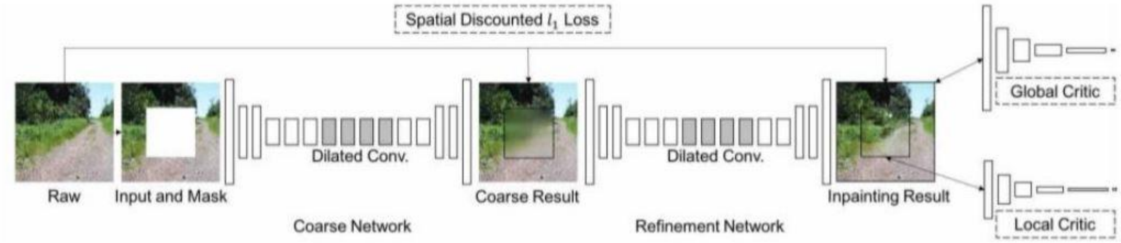


Figure 8: Generative Image In painting with Contextual Attention.

2.7 Generative Adversarial Networks: An Overview [7]:

Generative adversarial networks (GANs) is a Deep Learning architecture that is used to learn deep representations without extensively annotated training data. They achieve this through implicitly modelling high-dimensional distributions of data by training a pair of networks in competition with each other. In GAN there are two networks one is a generator, which generates fake sample of data while the other is a discriminator which recognize the fake data sample from the true data sample.

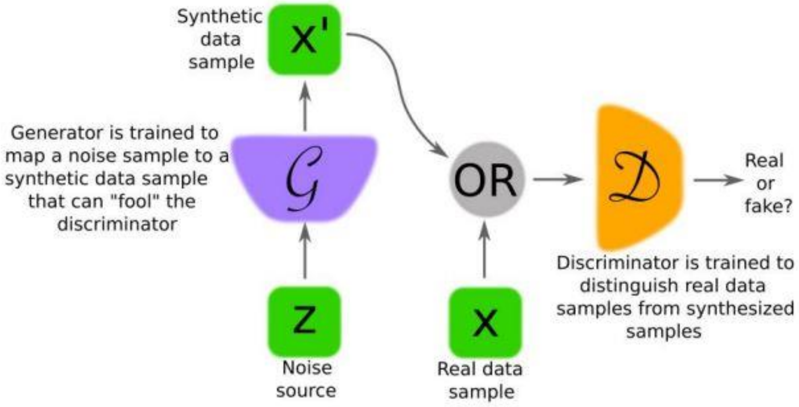

Synthetic data sample
X'
Generator is trained to map a noise sample to a synthetic data sample that can "fool" the discriminator
𝒢
OR
𝒟
Real or fake?
Discriminator is trained to distinguish real data samples from synthesized samples
Z
Noise source
X
Real data sample

Figure 9: : GAN Methodology

The goal of GAN is to gain optimality to the point where the fake data samples are not recognizable by discriminator. GAN is used in many different sectors such as image recognition, image synthesis, semantic image editing, style transfer, image super resolution and classification and image generating. GAN is used to generate fake image that is close to the real image.

2.8 On the Properties of Neural Machine Translation: Encoder–Decoder Approaches [8]:

Encoder decoder neural network model is a technique that takes a sentence or image as input and represents that input as vector. While another network called decoder that takes the vector representation of encoder and convert them into a sentence or image that is close to the original shape.

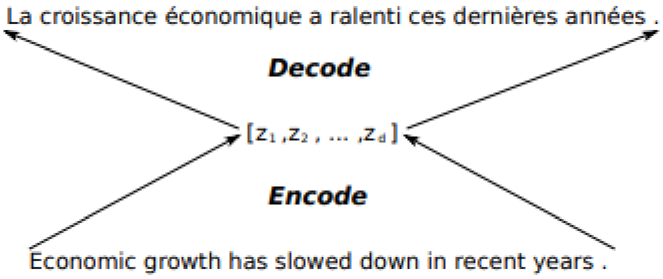

La croissance économique a ralenti ces dernières années .
Decode
[z₁ ,z₂ , ... ,z_d]
Encode
Economic growth has slowed down in recent years .

Figure 10: Encoder Decoder architecture

2.9 Object detection and currency recognition using CNN [10]:

Convolutional neural networks are one of the most popular techniques used to improve images classification accuracy. It is a special type of neural network that works in the same way as a regular neural network except that at the beginning it has a convolution layer and pooling layer.

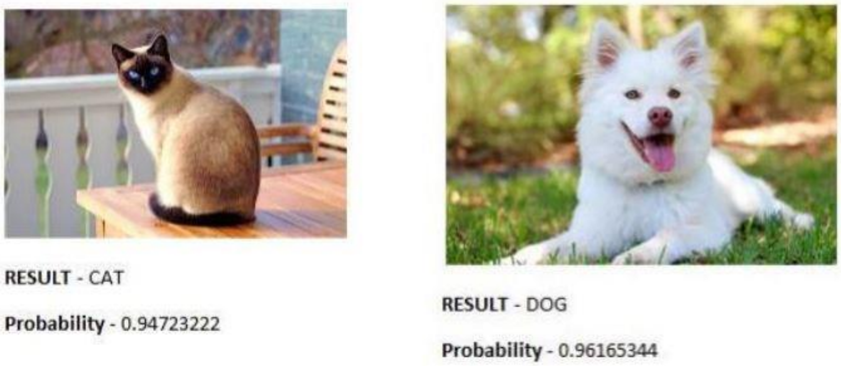


Figure 11: : Object detection using CNN

The system uses the Cifar-10 dataset which has 32 x 32 pixels of 60,000 images and 10 classes. Then it splits the dataset into 2 parts- 50,000 images for training and 10,000 images testing. It artificially put some noise in data to get variation. After training the model it got 87.4% accuracy in testing.

2.10 Drawing trajectory generation for partial drawing by 2D shape completion [11]:

Hand drawn sketch recognition is a fundamental problem in computer vision. It's used in sketch based image and video retrieval, editing and reorganization. But there is not enough research previously that suggests any well enough method to recognize and complete the incomplete hand drawn shapes of various kind and angles. In this paper, A CNN is used for shape recognition and A GAN is used to further complete the missing part of it. This method can be used on retrieving hand drawn sketches and writings that missing some part of it.

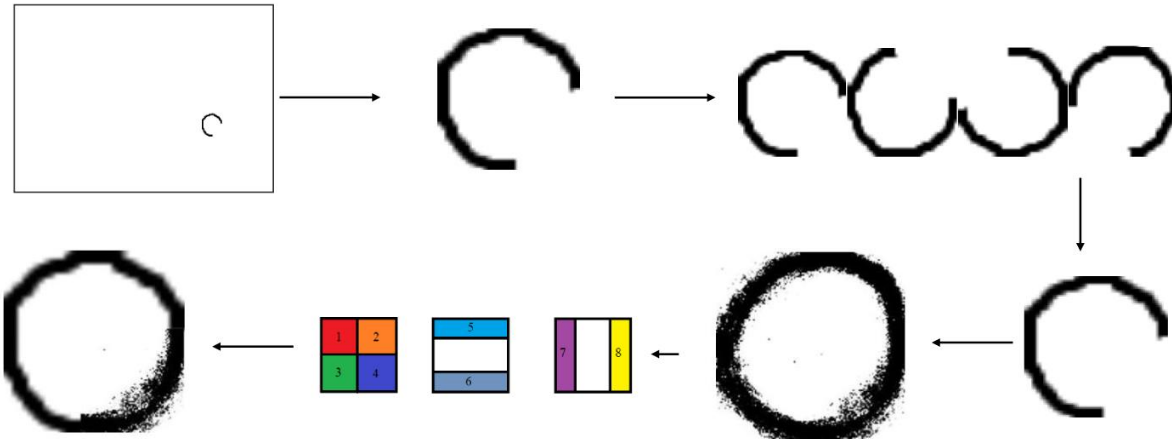


Figure 12: Working Procedure of Shape Completion

2.11 SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation [12]:

The article introduces SegNet, an efficient architecture for pixel-wise semantic segmentation that is motivated by road scene understanding applications. SegNet's encoder network is topologically identical to the convolutional layers in VGG16 and it's decoder network uses max-pooling indices to perform non-linear up sampling of the input feature maps, improving boundary delineation and reducing the number of parameters. The paper also analyzes the trade-offs of designing segmentation architectures and compares SegNet's performance to other approaches on two scene segmentation tasks, CamVid and SUN RGB-D.

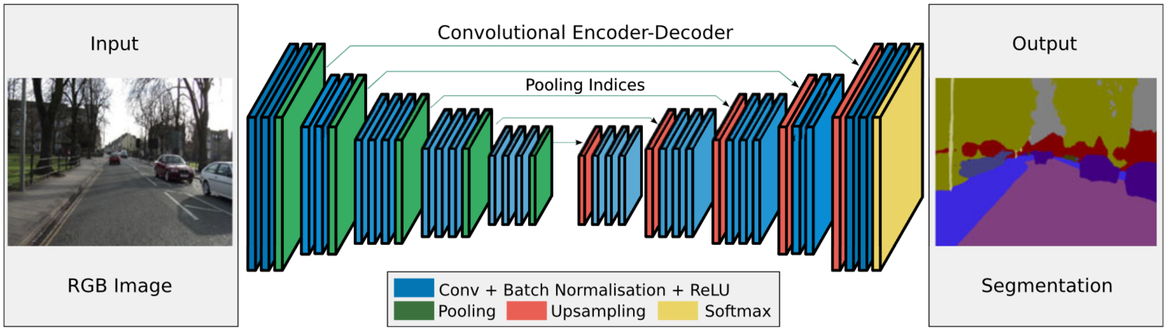


Figure 13: An Illustration of SagNet Architecture

Chapter 3

METHODOLOGY

The methodology we applied was trained and tested on Google Colab.

All the algorithms were implemented as python programs. The training & testing took around 2.5 hours (around 30 minutes for each class - Encoder Decoder Generation Model).

3.1 Dataset

We are using a customized dataset for this project as there is no proper dataset we found which is hand-drawn as well as partial. We collected some data from Khadem Mohammad Asif-uz-zaman's hand-drawn shape dataset and labeled some of the partial data in order to use with our methodology. In our system there are two part of our dataset for every class (shape). One contains complete representation of a shape and another contains incomplete. In Encoder-Decoder based system to get best performance we need a large number of input data for each class. So, we build a web application using php, html canvas, css & javascript to generate datasets easily. Using that application, we made about 900 images for each class. Some samples of our dataset are shown below.

3.1.1 Incomplete and Complete Star Shapes

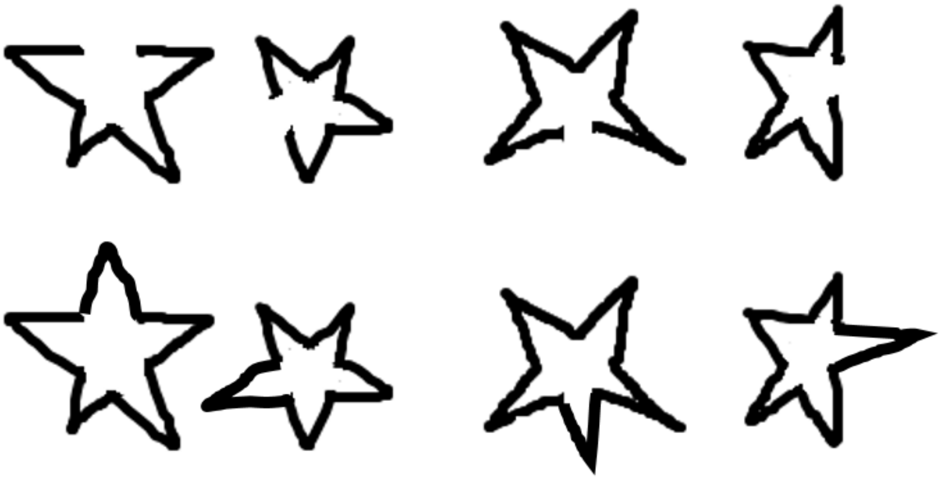


Figure 14: Some Samples of Incomplete and Complete Star Shapes

3.1.2 Incomplete and Complete Circle Shapes

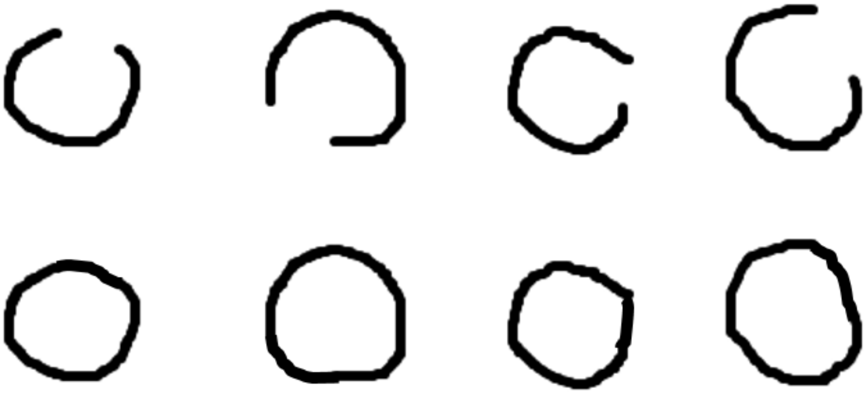


Figure 15: Some Samples of Incomplete and Complete Circle Shapes

3.2 Getting Input

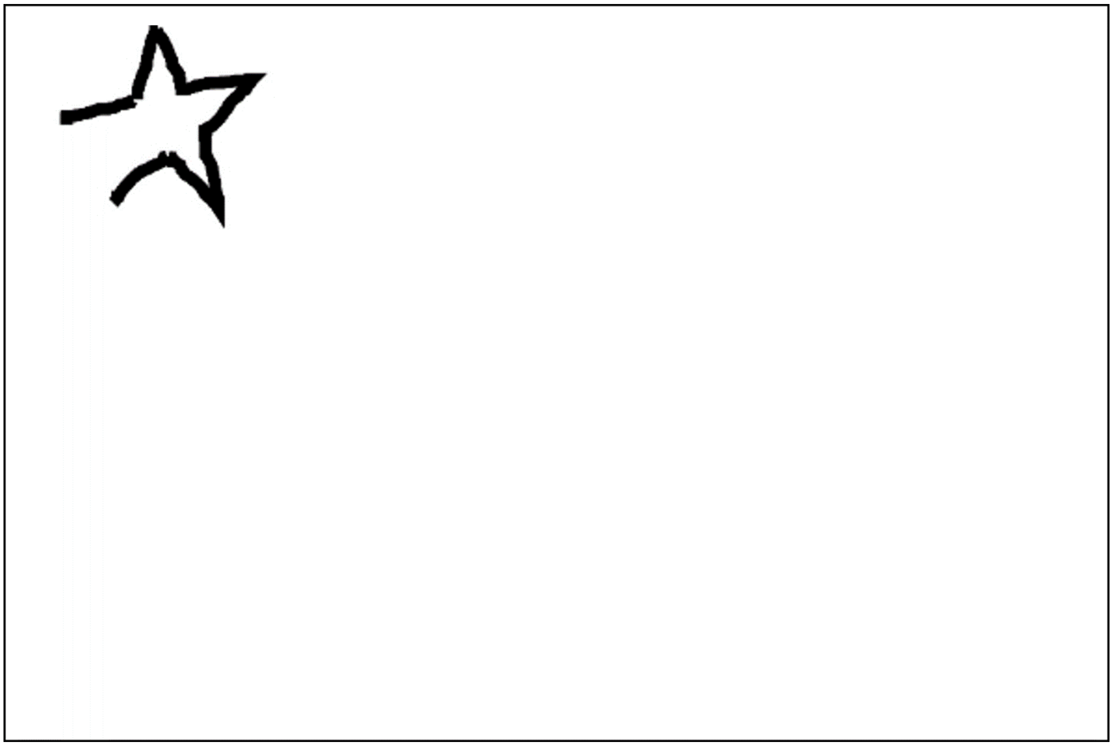


Figure 16: Getting Incomplete Shape from User

The Input can be in any size, shape and in any position. Here in this case the input image is in shape of 320x444

3.3 Bounding The Image

After getting the image, our first task is to bound and crop the image to keep the actual shape. We used a pixel to pixel bounding system to bind the object. After bounding the object, the image looks like:

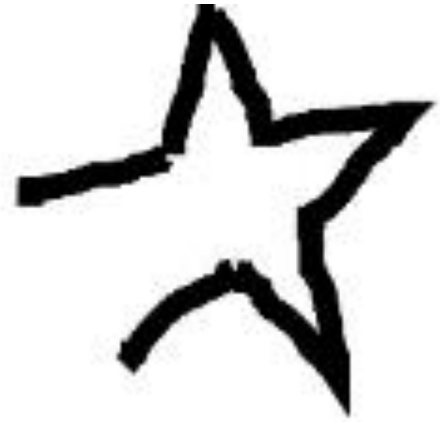


Figure 17: The Input After Bounding

Then we converted the image into 56 X 56 pixels as our detection and completion will be in this shape. In this example the bounded shape was 39 X 35 pixels.

3.4 Encoder-Decoder Network Model for Image Generation

In our system, we employed an Encoder-Decoder Network Model to generate complete images from partial images. The model was initially trained using our custom dataset. During the training phase, we utilized a binary cross entropy loss function for calculating and minimizing loss. Initially, we attempted to use a custom loss function, however, after conducting experiments, we determined that binary cross entropy loss was more effective for our model. As our dataset consisted of only two colors (black as the foreground/shape and white as the background), we found that binary cross entropy loss was well-suited for our training model.

After the model was trained, the knowledge acquired during the training phase was utilized to generate complete shapes from partial, user-defined unknown images by the network model. We defined and utilized one network model for each class.

As an example, we present one sample of the generated complete images from an unknown dataset from a user for each class:

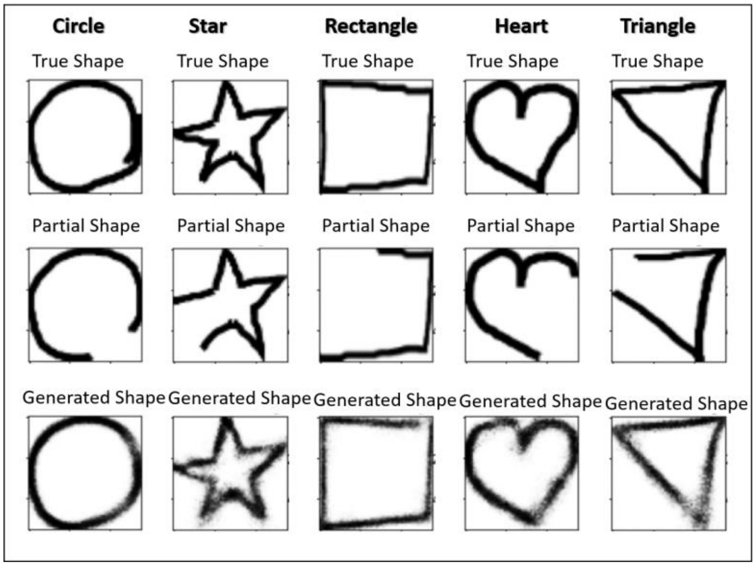


Figure 18: Sample images of Generation For Each Class

3.5 Reverting The Complete Image to Its Initial Format

The positional and dimensional parameters that were stored while bounding the partial image is used to restore the generated image to its initial form factor.

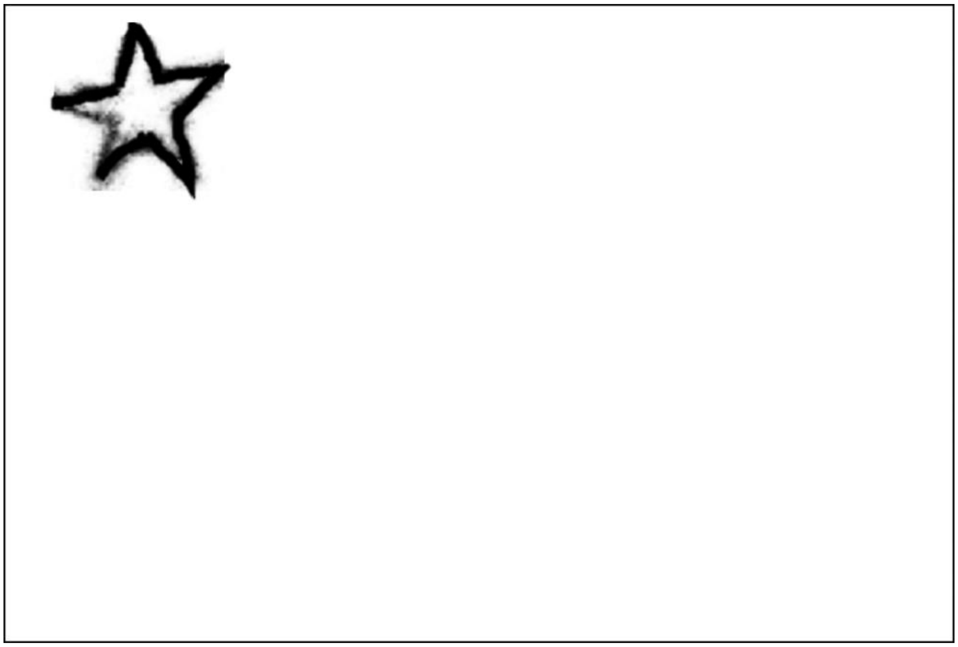


Figure 19: Reverting Generated Completed Image to Initial Form

3.6 Our Network Model

Our Encoder-Decoder model is a sequential model and consists of 6 dense layers on the encoder side along with 5 dense layers on the decoder part. Here's a brief summary of our model architecture and important parameters:

- **Encoder:**
  - Layer 1 - Type: Dense - Shape: 2500 - Activation: LeakyRelu
  - Layer 2 - Type: Dense - Shape: 2000 - Activation: LeakyRelu
  - Layer 3 - Type: Dense - Shape: 1500 - Activation: LeakyRelu
  - Layer 4 - Type: Dense - Shape: 1000 - Activation: LeakyRelu
  - Layer 5 - Type: Dense - Shape: 500 - Activation: LeakyRelu
  - Layer 6 - Type: Dense - Shape: 100 - Activation: LeakyRelu

- **Decoder:**
  - Layer 1 - Type: Dense - Shape: 300 - Activation: LeakyRelu
  - Layer 2 - Type: Dense - Shape: 500 - Activation: LeakyRelu
  - Layer 3 - Type: Dense - Shape: 1000 - Activation: LeakyRelu
  - Layer 4 - Type: Dense - Shape: 2000 - Activation: LeakyRelu
  - Layer 5 - Type: Dense - Shape: 3136 - Activation: Sigmoid

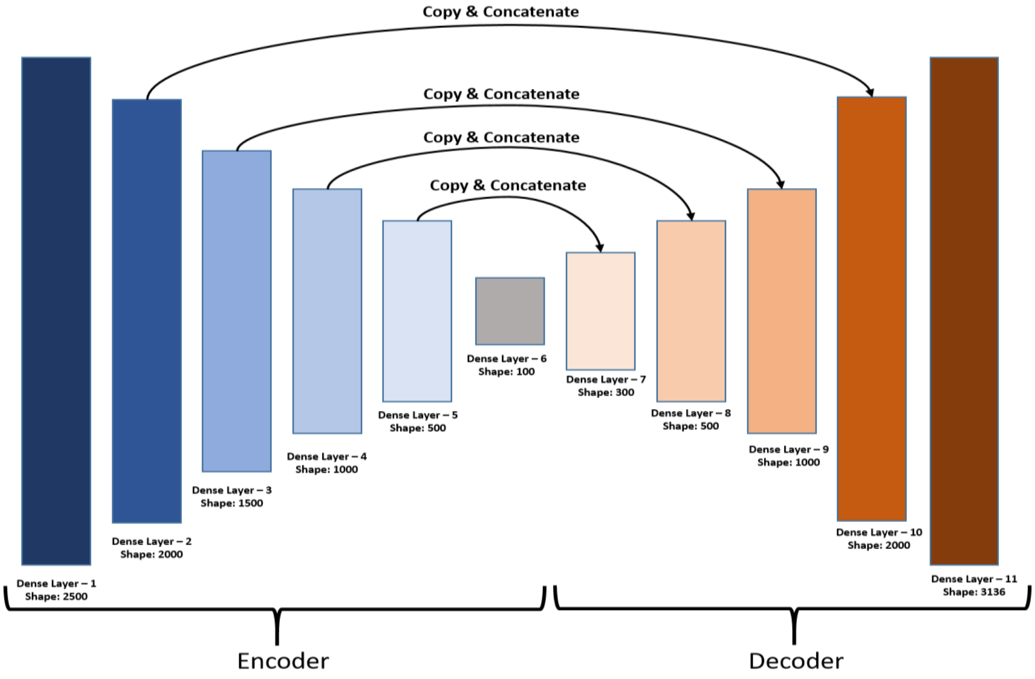


Figure 20: Visual Representation of Our Network Model

3.7 Overview of Our System: Training Phase of Network Model with Training Dataset

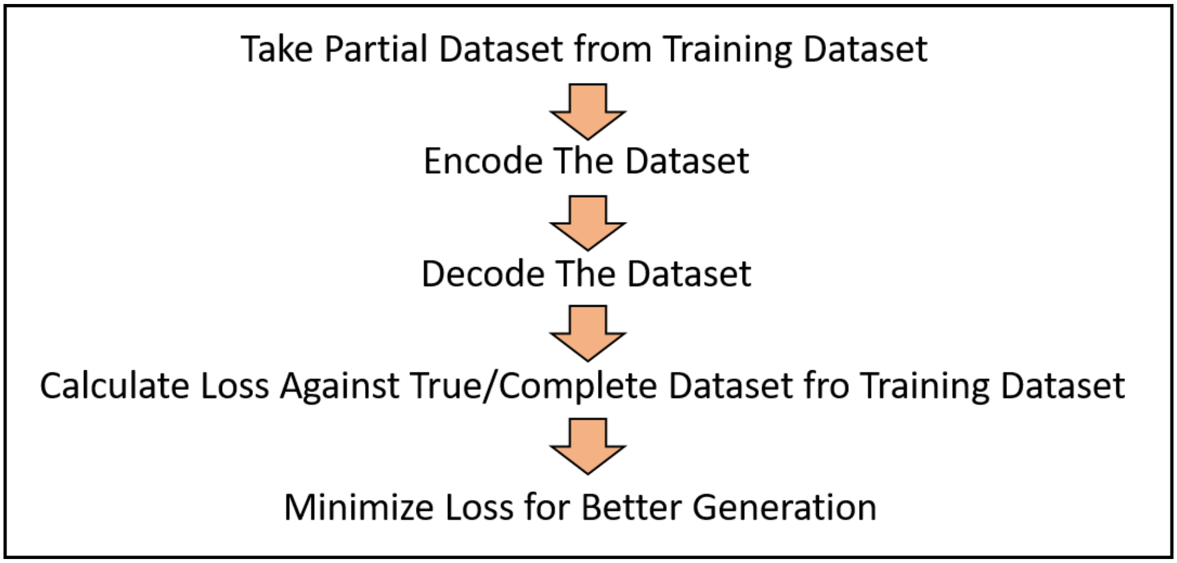

Take Partial Dataset from Training Dataset
Encode The Dataset
Decode The Dataset
Calculate Loss Against True/Complete Dataset fro Training Dataset
Minimize Loss for Better Generation

Figure 21: Training Phase of Network Model from Training Dataset

3.8 Overview of Our System: Testing Phase of Network Model from Unknown User Data

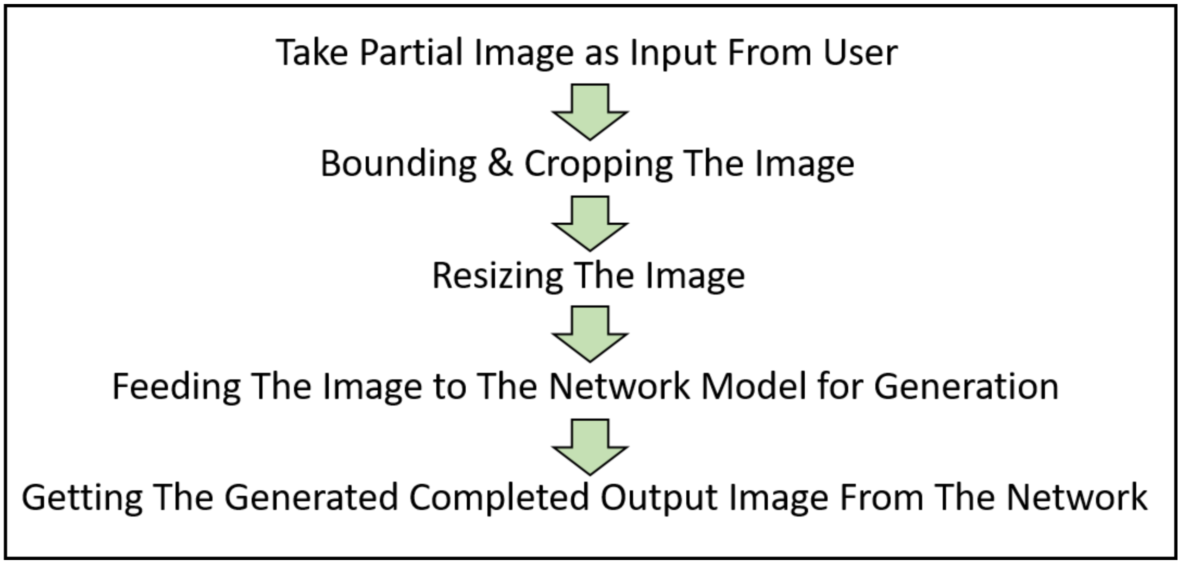

Take Partial Image as Input From User
Bounding & Cropping The Image
Resizing The Image
Feeding The Image to The Network Model for Generation
Getting The Generated Completed Output Image From The Network

Figure 22: Testing Phase of Network Model from Unknown User Data

3.9 Binary Cross Entropy Loss Function and Our Loss Calculation

In the context of image completion, the binary cross-entropy loss function is used to compare the predicted image (also known as the output image) with the true image (also known as the target image). The goal is to minimize the difference between these two images, which is measured using the binary cross-entropy loss.

The predicted image is generated by our neural network, which is trained to complete missing or corrupted parts of an input shape. The network takes the input image and outputs a predicted image with the same dimensions. The predicted image is then compared to the true image using the binary cross-entropy loss function.

The binary cross-entropy loss function is used because the output of the network is a probability map, where each pixel represents the probability of being the correct color of that pixel. The true image, on the other hand, is a binary image, where each pixel can be either 0 or 1, indicating the true color of that pixel.

The function calculates the loss by comparing the predicted probability map with the true binary image for each pixel, and sums up the negative logarithm of the predicted probability for the true color of each pixel. The final loss is a scalar value that represents how well the network is doing at completing the image. The goal is to minimize this value during training, so that the network can learn to generate images that are as similar as possible to the true images.

3.10 A Brief on Encoder-Decoder Network Model

An encoder-decoder model is a type of neural network architecture that is commonly used for image generation tasks, such as image completion. The encoder is a neural network that takes in an input image and "encodes" it into a compact representation, often called a bottleneck or latent code. This compact representation captures the most important features of the input image, and is then used as the input for the decoder, which is another neural network that "decodes" the compact representation back into an output image.

The encoder and decoder are typically trained together, using a dataset of complete images. The encoder learns to extract the most important features of the input images, and the decoder learns to use these features to generate new images that are similar to the input images.

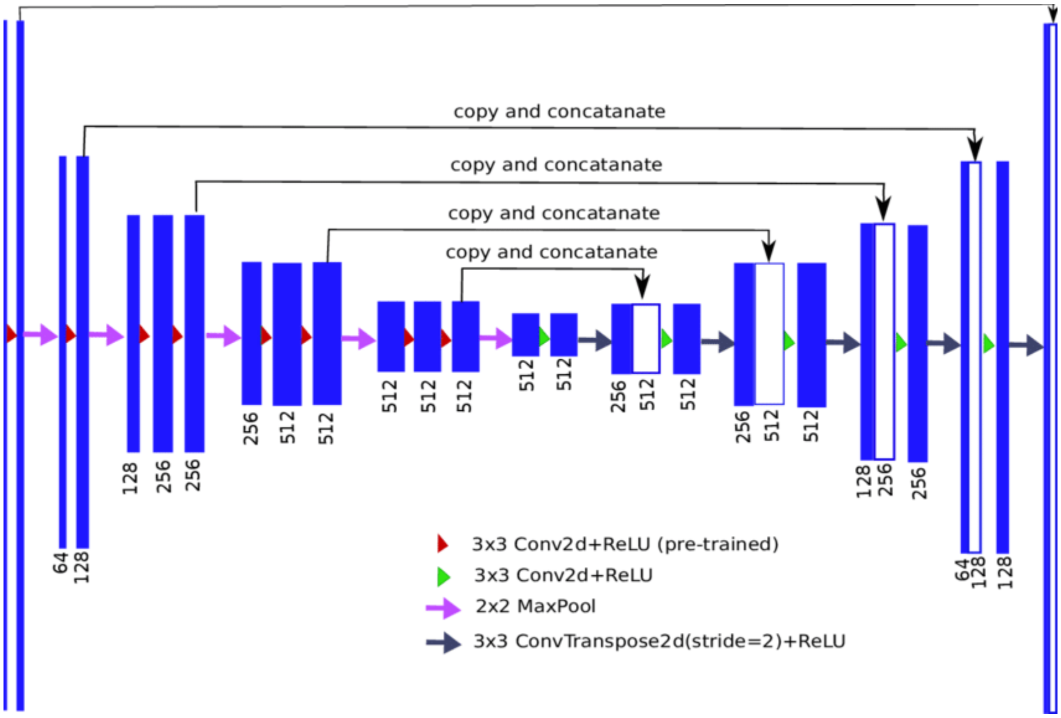


Figure 23: Generalized Architecture of Encoder-Decoder Network Model (Source: ResearchGate)

The encoder typically consists of several layers of convolutional or fully connected neural network layers. These layers are trained to extract important features from the input, and to reduce the dimensionality of the input. The final layer of the encoder is called the bottleneck, it holds the compact representation of the input.

The decoder, on the other hand, typically consists of several layers of transposed convolutional and/or fully connected neural network layers. These layers are trained to "expand" the compact representation back into the original input space. The final layer of the decoder produces the output, which should be similar to the original input.

During the inference stage, the encoder takes in an input and encodes it into a compact representation. This compact representation is then passed to the decoder, which generates the output.

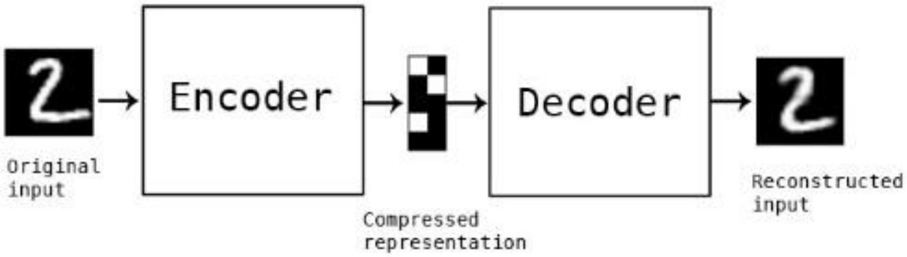


Figure 24: Working Mechanism of Encoder Decoder (Source: KerasBlog)

In summary, an encoder-decoder neural network model is a two-part architecture consisting of an encoder and a decoder. The encoder takes in the input and encodes it into a compact representation, while the decoder takes in this representation and decodes it back into an output. The model is trained to learn the relationship between the input and the output, and can be used for tasks such as image generation, image captioning, and machine translation.

3.11 Encoder-Decoder vs GAN

Encoder-decoder network and Generative Adversarial Network (GAN) are both neural network architectures that can be used for image generation tasks, but they have some key differences in their approach and architecture.

Encoder-decoder networks consist of two main components: an encoder and a decoder. The encoder takes in an input image and encodes it into a compact representation, which is then passed to the decoder to generate the output image. The encoder and decoder are trained together to learn the relationship between the input and output images. The encoder learns to extract the most important features of the input image, and the decoder learns to use these features to generate an output image that is similar to the input image.

On the other hand, GANs consist of two main components: a generator and a discriminator. The generator takes in a random noise and generates an output image, while the discriminator takes in an image and classifies it as either real or fake. The generator and discriminator are trained together in an adversarial way, where the generator learns to generate images that can fool the discriminator, and the discriminator learns to correctly classify real and fake images.

Encoder-decoder networks are generally considered to be more stable and easier to train than GANs, and they often produce high-quality images. However, GANs are capable of producing more diverse and realistic images, as they are able to learn the underlying probability distribution of the data.

In summary, Encoder-Decoder networks and GANs are both neural network architectures that can be used for image generation tasks, but they have different architecture and approach. Encoder-Decoder networks are easier to train and produce high-quality images, while GANs can produce more diverse and realistic images by learning the underlying probability distribution of the data.

3.12 Why We Used Encoder-Decoder rather than GAN

Generative Adversarial Networks (GANs) have been observed to produce outputs that are less closely aligned with the input shape, as they are based on the training dataset and random noise. But the encoder-decoder architecture has been shown to be particularly effective for generating images that are closely aligned with the input shape.

In this research, we were interested in generating complete images from partial inputs. To accomplish this goal, we employed an encoder-decoder network model. Our results demonstrate that this model was able to generate outputs that were closely aligned with the input shape.

In addition to its improved performance, the encoder-decoder architecture has the added advantage of being less resource-intensive than GANs. GANs are known for their resource-intensive nature, due to their complex architecture and working principle. In contrast, the encoder-decoder model requires minimal resources and time to train & generates output based on input on perspective of the learning from training, making it a more practical choice for our research application.

Chapter 4

DATASET

Generating huge training image dataset that consists one partial shape and its corresponding completed shape for each sample is difficult. That's why we developed a web application utilizing the technologies of PHP, HTML canvas, CSS, and JavaScript. The purpose of this application is to facilitate the collection of a large training dataset consisting of samples containing one partial shape and its corresponding completed shape.

4.1 Dataset Description

Using our web application, we create a dataset that consists of 900 partial and 900 completed corresponding images per class, total of 5 classes. In total we have made (900 + 900) x 5 = 9000 images.

Our dataset classes:

1. **Circle - (900+900 images)**
2. **Heart - (900+900 images)**
3. **Rectangle - (900+900 images)**
4. **Triangle - (900+900 images)**
5. **Star - (900+900 images)**

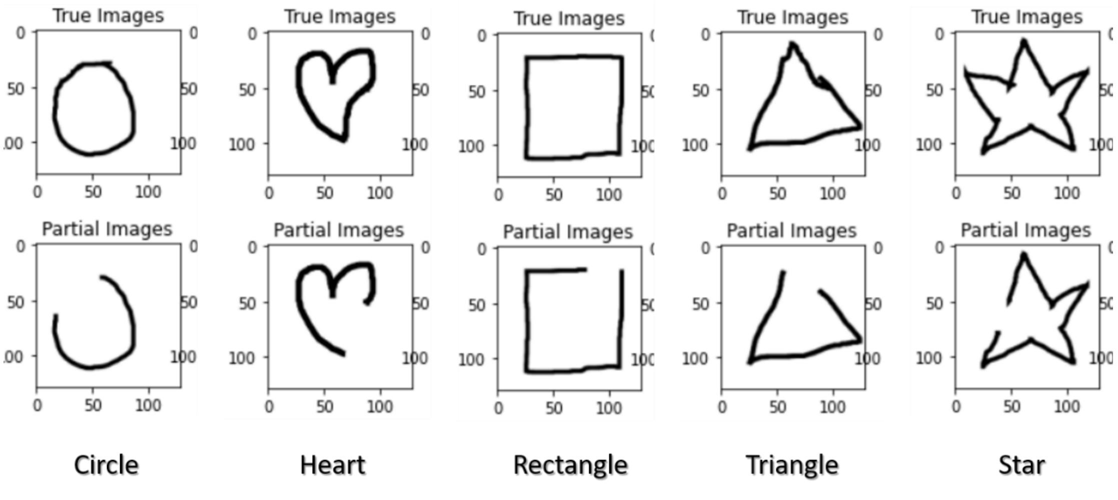
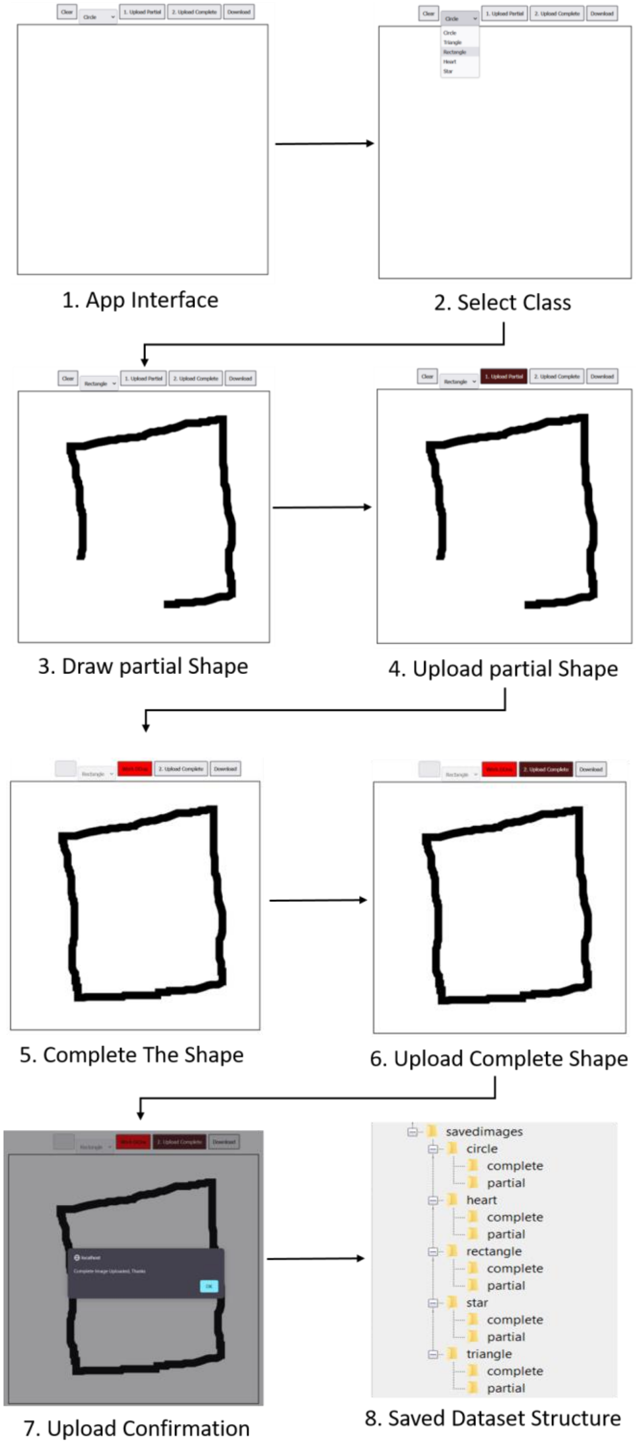


Figure 25: Preview of the Dataset Samples

4.2 Web Application for Building Dataset

Here’s an overview of our dataset builder application and it’s working procedure:



1. App Interface

2. Select Class

3. Draw partial Shape

4. Upload partial Shape

5. Complete The Shape

6. Upload Complete Shape

7. Upload Confirmation

8. Saved Dataset Structure

Figure 26: Overview Of Dataset Builder Web App

Chapter 5

EXPERIMENTS

In the process of conducting research for this thesis, a comprehensive examination of various aspects of machine learning was undertaken. This included an in-depth analysis of different types of machine learning models, loss functions, datasets, and approaches. The goal of this research was to gain a thorough understanding of the current state of the field and to identify potential areas for further study.

5.1 Initial Experiments and Training

In the initial stages of our research, we intended to utilize a Generative Adversarial Network (GAN) model for our system. However, after conducting a thorough literature review, it was determined that utilizing an Encoder-Decoder Network would be more appropriate for the specific field of study. During the initial experimental training phase, a basic model was trained utilizing estimated parameters with a substantial number of epochs. Each training session required a significant amount of time, ranging from 2 to 6 hours per class. The results of this initial experimentation were used to inform the development of the final model and optimize its performance.

5.2 Final State of Training

After conducting a comprehensive examination of various features and optimization parameters, and experimenting with encoder-decoder models, a significant advancement in the training time of the model was achieved. Specifically, the model was able to be trained within a time period of 20 minutes & 1000 epoch per class, for a total of 100 minutes for five classes.

The model's learning from this brief training period resulted in high-accuracy predictions of complete shapes from partial shapes on unknown datasets. Specifically, the accuracy was in excess of 90%. In certain cases, for specific classes and input types, the model's predictions of completed shapes surpassed those of the user-defined true/completed shapes.

To clarify this, here’s an example image below:

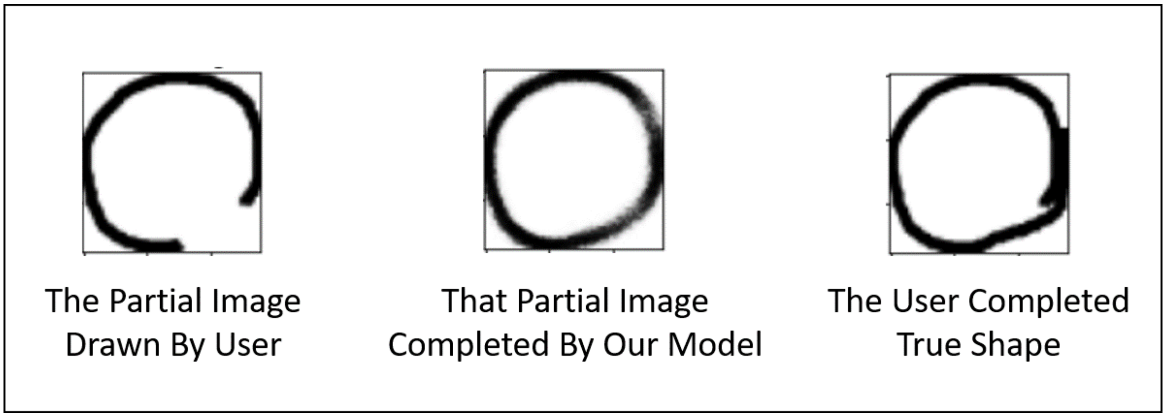

The Partial Image Drawn By User
That Partial Image Completed By Our Model
The User Completed True Shape

Figure 27: Model Generates Better Result than True Shape

From this image, we can see that the image generated by our model is even better than the one which is completed by the user.

Chapter 6

RESULTS & DISCUSSION

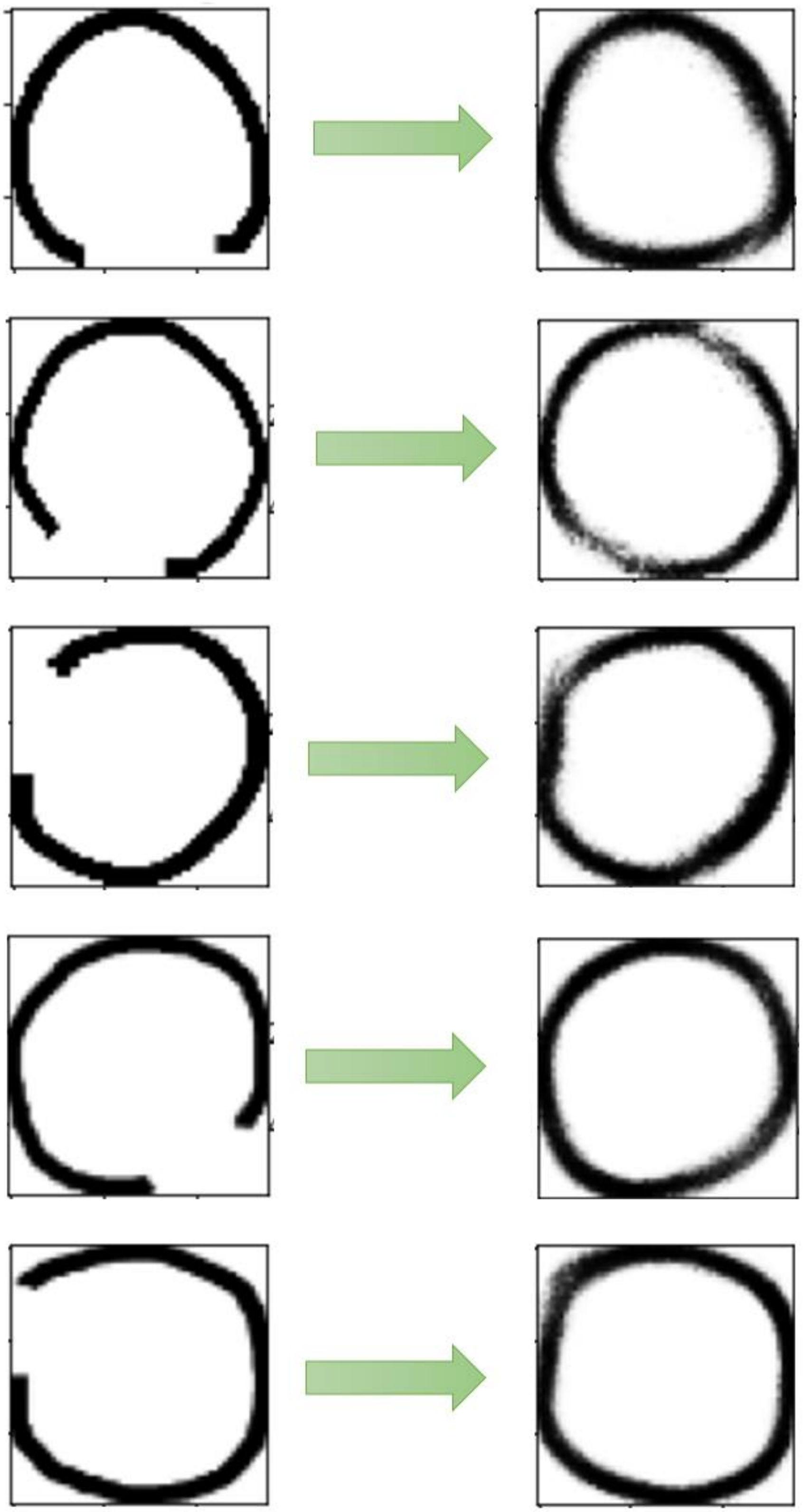
Our machine learning model, to be specific our Encoder-Decoder model generates almost near perfect results by using minimum resources, processing power and time.

6.1 Result

We conducted our system's image generation and completion capability using a dataset of 100 partially drawn images from various individuals. The images varied in size and shape trajectory, including some very odd shapes.

Of these images, 73 were found to have been completed with high accuracy, with an average completion rate of 90-95%. 15 images were determined to have an accuracy of approximately 50%, while the remaining 12 images were deemed to have insufficient accuracy and were classified as "garbage generation." The overall average accuracy of our image completion system was calculated to be 78%.

Here's the demonstration of our system's image/shape completion:



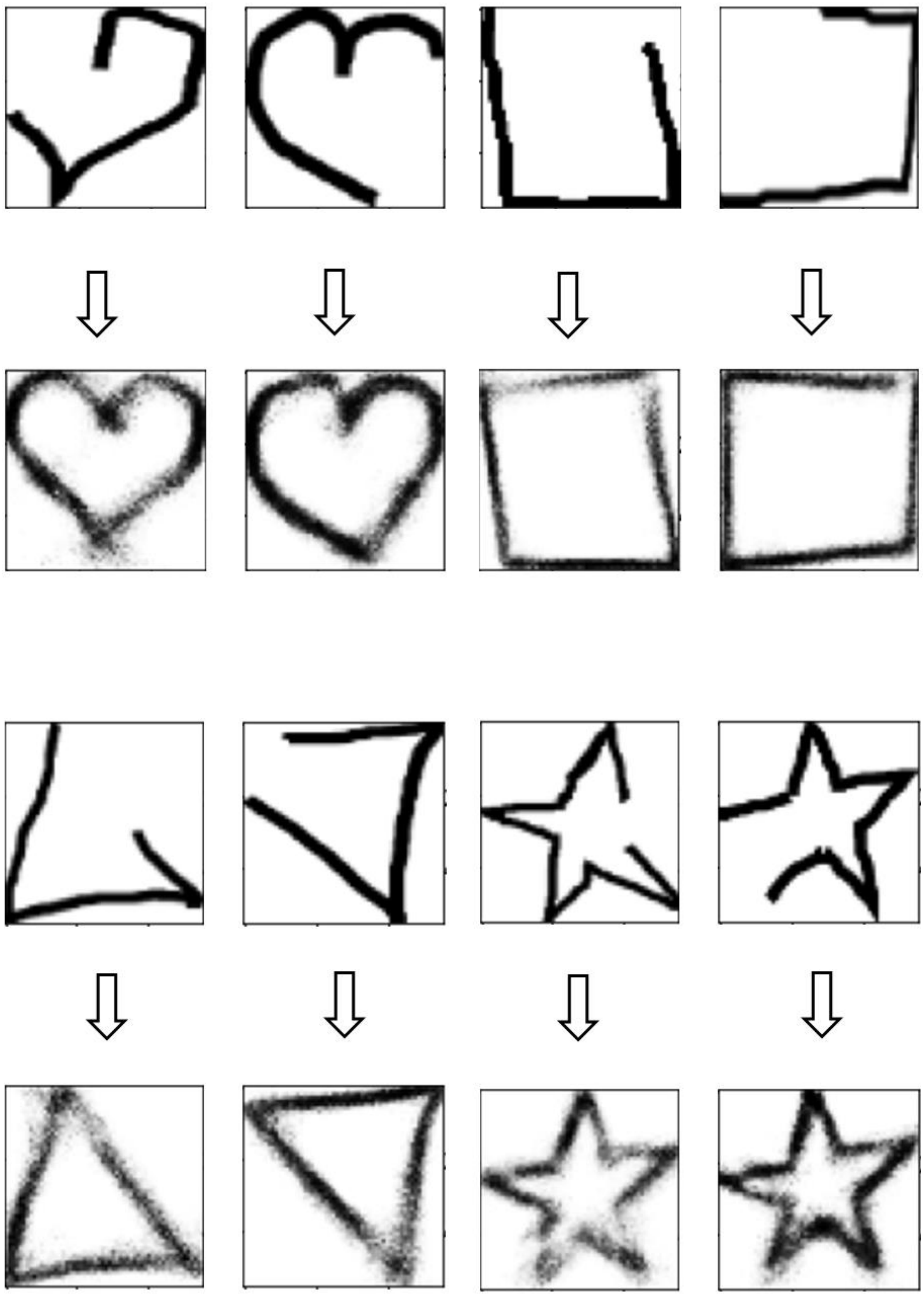


Figure 28: Completed Shapes from Partial Non-Standard Shapes Drawn By Random Users

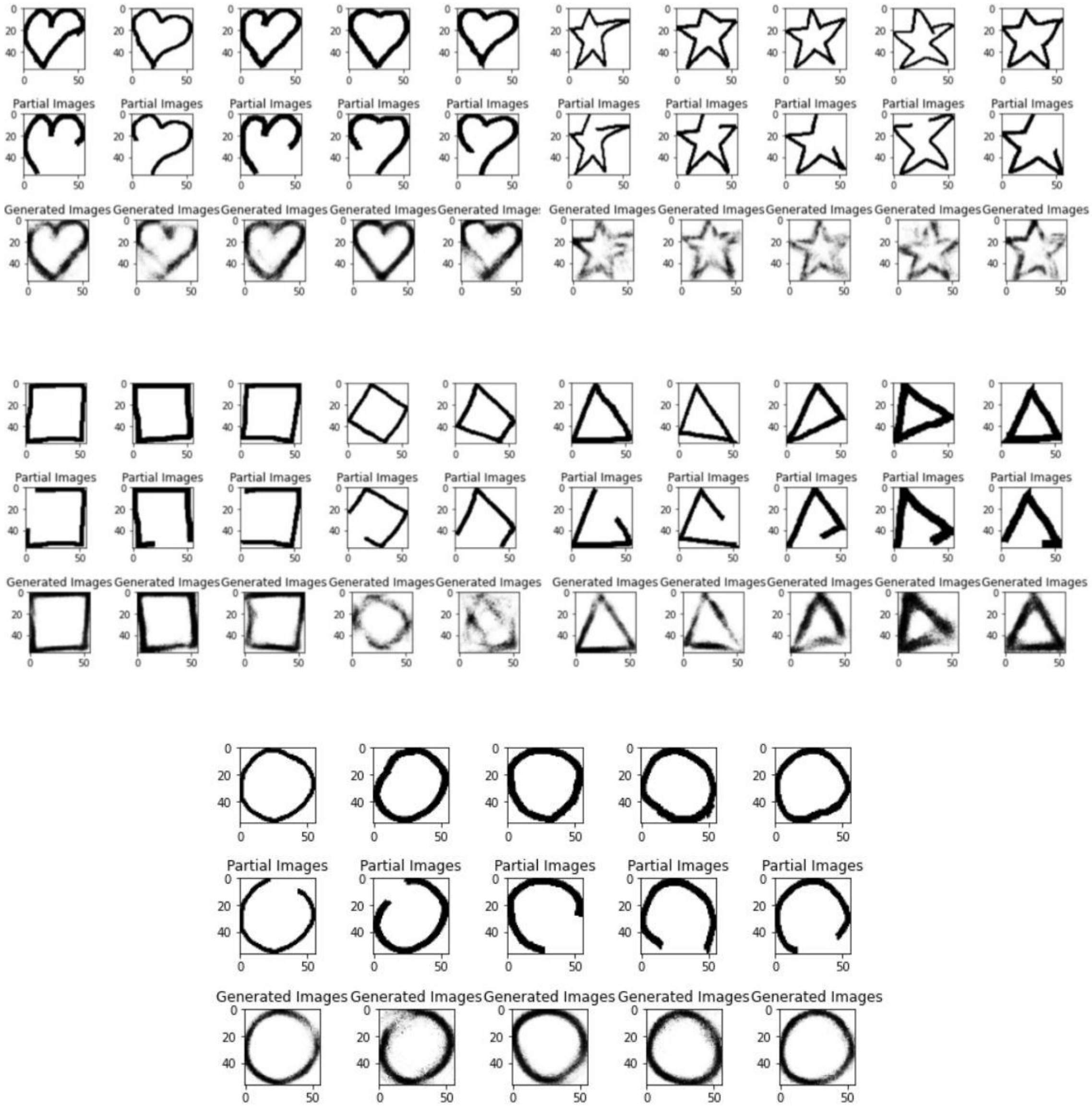


Figure 29: Some More Generation & Completion Results on Unknown User Dataset

6.2 Discussion on Result

In previous studies, the use of GANs and other advanced machine learning network models for image generation and completion has been observed. These systems, however, have been found to be resource-intensive, requiring a significant amount of time and data for training. Despite these efforts, they often produce noisy and imperfect results.

But we thought outside the box and planned to simplify the system of image generation and completion by using simpler machine learning model to increase accuracy & efficiency. And we got success in our research, we finally developed a machine learning model that is:

- **Way Less Time Consuming**
- **Generates way more accurate results**
- **Simple architecture so easier to implement**

6.3 Future Work

The machine learning model that we developed in our research, can be further used in various fields that requires image generation and completion such as Self-Driving Technology for reading roadside signs while driving, teaching children basic shapes in their initial learning phase.

This type of machine learning model that works with non-standard shapes and images can be used in robot learning fields with a huge potential.

We will try to use this system in similar types of fields in future.

Chapter 7

CONCLUSION

Hand-drawn shape and sketch completion is a critical area of computer vision with a wide range of practical applications. Despite its importance, there has been limited research in this field. Previous efforts have utilized Generative Adversarial Networks (GANs), but these solutions have been found to be inadequate. In our research, we have replaced the traditional GAN approach with an "Encoder-Decoder Network" and developed a comprehensive dataset to train our system. Our system is able to work with hand-drawn sketches of any form factor, making it useful in fields such as education, art, and robotics. We plan to continue our research to further improve the system and our understanding of this topic

References

[1] Liu, Fang, et al. "Sketchgan: Joint sketch completion and recognition with generative adversarial network." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[2] G uo, Ge, et al. "2-D shape completion with shape priors." Chinese Science Bulletin 58.27 (2013): 3430-3436.

[3] Ghosh, Arnab, et al. "Interactive sketch & fill: Multiclass sketch-to-image translation." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[4] Sarmad, Muhammad, Hyunjoo Jenny Lee, and Young Min Kim. "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[5] Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion." ACM Transactions on Graphics (ToG) 36.4 (2017): 1-14.

[6] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang. "Generative Image inpainting with Contextual Attention". CVPR 2018.

[7] Creswell, Antonia, et al. "Generative adversarial networks: An overview." IEEE Signal Processing Magazine 35.1 (2018): 53-65.

[8] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". 3 Sep 2014 21:03:41 UTC.

[9] Pawar, Mahesh, et al. "Object Detection and Currency Recognition Using CNN." Cikitusi Journal For Multidisciplinary Research 6.4 (2019): 357-362.

[10] Khadem Mohammad Asif-uz-zaman. "Drawing trajectory generation for partial drawing by 2D shape completion." North East University Bangladesh Thesis Research Publication (2022).

[11] Vijay Badrinarayanan, Alex Kendal. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.” EEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 39, NO. 12, DECEMBER 2017.

[12] www.developers.google.com

[13] www.machinelearningmastery.com