Sketch to Image Translation

Nice Maria, Jayalakshmi V[1], J Ananthakrishnan[2], Tania Sam[3], Deepa P L[4]

[1-4]*Mar Baselios College of Engineering & Technology, Trivandrum, Kerala, India*

---***---

Abstract – *Sketch identification and colouring of the image together is an unaddressed issue so far. In our project the half-drawn sketch is identified first and then completed using neural networks. The half-drawn sketches are completed as full drawn doodles as for now. The identified sketches are of birds, flowers, common objects, anime faces etc. The completed doodling is then given to a generative adversarial network which is trained using several datasets. The sketches are then resized, reshaped and then coloured. The GAN network consists of a generator circuit and a discriminator circuit. The generator of GAN contains a cascade network, where each stage is consisted of a conditional GAN. The first stage of generator uses an incomplete hand drawn sketch as the input, and then gradually completes the contouring. For the later stages with the original input cascaded with the output of the previous stage is given, and is fed to the generator of the current stage. The discriminator circuit is used to find the coherence between the generated images with the original one which is already present in the dataset. The final output of the circuit will be a completed coloured image of the given half drawn sketch.*

1. INTRODUCTION

Sketch is a common metaphor for communicating abstract ideas in a straightforward manner, commonly used in computer vision, reorganization, and communication with human computers. Many studies have been made about sketch recognition problem in recent years. Deep learning algorithms are used widely for sketch recognition and translation. Deep learning is a technique of machine learning that trains system to do what naturally enters to people. In deep learning, a system-based model studies directly from pictures, text, or sound to perform classification tasks. Models are skilled by using a large labelled dataset and neural network frameworks that consists of many layers. The neural networks may modify to evolving input: therefore, the circuit produces the best available output after redesigning the performance criteria. Neural networking concept, which is related to artificial intelligence, is rapidly achieving acceptance in trading systems development. In this project we make use of "**Recurrent Neural Networks (RNN)**" and "**Generative Adversarial Networks (GAN**). RNN is a set of artificial neural network, where the connections between nodes form a directed graph along a temporal sequence. RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario. They are mainly used in 'Natural Language Processing' and speech recognition. GAN is also a deep learning, unsupervised machine learning technique. It consists of 2 networks. A generator network and a discriminative one. The generator generates images which looks like the original one. The discriminator's job is to identify the fake image generated by the generator with the help of datasets in the discriminator. Our goal is to generate a system which could generate a full completed image if an incomplete sketch is given. RNNs are used for the sketch completion purpose and GANs are used for image processing.

GANs are widely used for image completion and image processing vigorously. SketchGAN is a method used to complete a given sketch. Apart from the usual generator and discriminator units of GAN sketchGAN also consists of a classifier unit. It is a convolutional neural network model used for freehand sketch identification. The circuit is trained by using a cross-entropy loss along with the rest of SketchGAN. The main important features of SketchGAN are: The cascade network of the generator and the auxiliary sketch identification circuit. The cascade structure helps for better completion results. It helps to keep the finished sketches close to the contour. The auxiliary sketch identification circuit. The sketch completion appears to be blurred without the auxiliary sketch recognition network. It helps to improve the sketch completion quality when multiple images are present. The method of finishing the sketch using SketchGAN works efficiently up to an incomplete ratio of about forty percent. The network has to extend to cope with large missing ratio. The method for sketch completion used in this paper is **sketch rnn** method which is the same technique used in [2]. Here the datasets are taken from the Quick Draw! which is an online game. We use the pretrained datasets from the sketchrnn mentioned in [2]. GANs have been updated in recent years for different purposes. The different types of GANs which are widely used for image processing purposes are conditional Gan, deep convolutional GAN etc. Here we use Pix2pix GANs [3]. **Pix2pix GANs** are the most advanced GAN for image completion. Pix2pix was first introduced by 'Philip Isola', 'Jun-Yan Zhu', 'Tinghui Zhou' and 'Alexei A Efros' in their paper "Image-to-Image Translation with Conditional Adversarial Networks" but here they use CNNs. But later on with advancement in neural networks pix2pix GANs are also available which have already proven their efficiency in image processing.

2. PROPOSED METHOD

The first procedure is completion of incomplete sketch. This is done by the sketch-rnn module. The completed sketch is then fed to the GAN network.

2.1 Sketch Completion

The block diagram for sketch completion network is given in the figure 1.

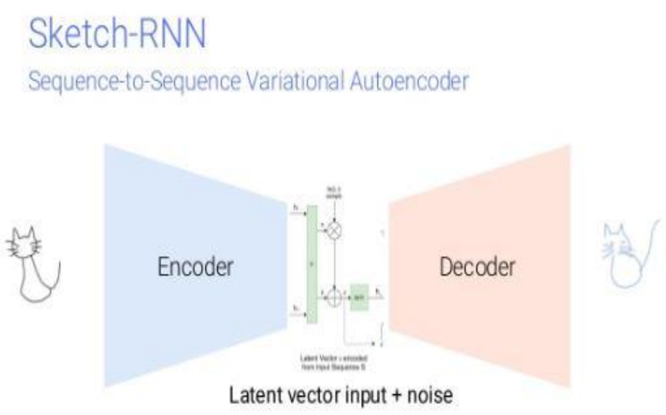

Fig 1: Encoder and decoder of sketch rnn [6]

The encoder of the network is a bidirectional RNN. The sketch sequence S and its reverse sequence S_r are given to two encoding RNNs simultaneously which makes the bidirectional network. We get two hidden states as the result of two networks which are h and h_r. These hidden states from each of the encoder blocks are projected to two vectors μ and $\hat{\sigma}$. An arbitrary parameter called z is formulated with μ and σ were

$$\sigma = \exp(\hat{\sigma}/2) \quad (1)$$

where $\mu = W_\mu h + b_\mu$ and $\sigma = W_\sigma h + b_\sigma$

The random vector z is conditioned on the input sketch. The decoder is an autoregressive RNN. The first hidden state of RNN is h_0, here

$$h_0 = \tanh(W_z z + b_z). \quad (2)$$

We feed S_0, h_0 and z together as the first input for the decoder circuit. For the second block we feed the output of first one ie; S_1 as the input. This process continues.

2.2 Image formation from sketch

We use pix2pix GANs for the image formation from the sketches. It is a type of **cGAN**. In cGANs the generator and discriminator receive some extra conditional input data. In GANs, the output image that is generated with the generator network is random. That is, it might generate images of any object that was there in the data set. But, with a cGAN, we can generate images what we want. If we want it to generate a person, it'll generate an image of a person. This is achieved by conditioning the GAN. Fig 2 gives the training of cGAN.

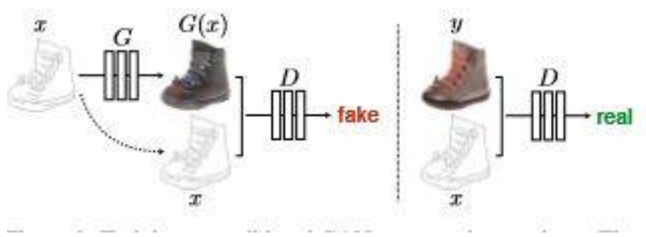

Fig 2: Training of cGAN [3]

cGAN learn the mapping from the detected image x and the noise vector z, to the output y, G: {x, z} → y.

3. NETWORK ARCHITECTURE

The incomplete sketch is given to the RNN module which translates it to completed sketch which is given to the GAN network and a completed image is produced. Fig 3 shows the complete block diagram of the network.

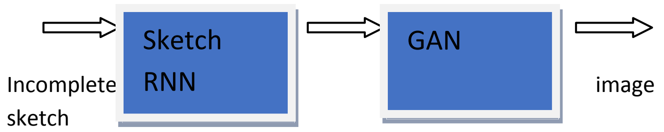

Fig 3: Block diagram of the network

The incomplete sketch is given to the RNN module which translates it to completed sketch which is given to the GAN network and a completed image is produced.

3.1 Sketch RNN

For the completion of incomplete sketch, the RNN decoder is used as a stand-alone model. When we remove the encoder the RNN will become an autoregressive model without any latent variables. The decoder RNN is used to first convert the sketch to a hidden state h. Later, the other points of the sketch are found by keeping h as the initial hidden state.

3.2 Pix2pix GAN

The pix2pix gan uses U-net architecture for generator which uses skip connections between the encoder and decoder layers of same size. This is repeated for each and every layer of the encoder and decoder which forms the U-net architecture. Fig 4 shows the U-net architecture.

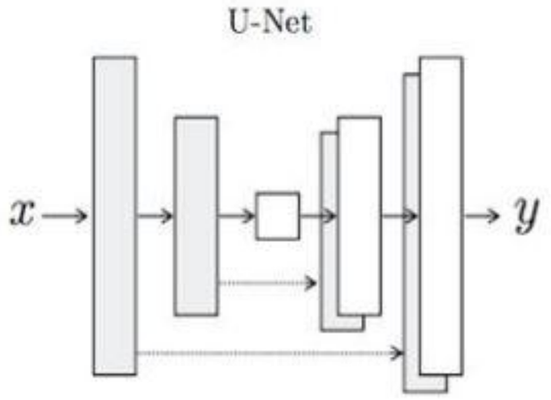

Fig 4: U-net architecture [3]

For discriminator patchgan model is used. The difference between patchgan and normal model is that for patchgan patches of an image is used for comparing real and fake rather than the whole image. The U-net architecture and the patchgan model increases the efficiency of pix2pix gan.

4. LOSS FUNCTION

The total generator loss of cGAN is given by,

$$\text{generator loss} = \text{gan\_loss} + \lambda \times L_1\text{\_loss} \quad (3)$$

were, L_1_loss is the mean absolute error between generated image and target image.

λ is an arbitrary value which is taken as 100.

The total discriminator loss is,

$$\text{discriminator Loss} = \text{real\_loss} + \text{generated\_loss} \quad (4)$$

where, real_loss is a sigmoid cross entropy loss of the real images and an array of ones. Generated_loss is a sigmoid cross entropy loss of the generated images and an array of zeros.

5. RESULT

Our project consists of two parts as mentioned earlier.

5.1 Sketch Completion

The RNN creates the networks with loops in them, which allows it to persist the information. It is this loop structure that enables the RNN to take the sequence of inputs. Since we require the content from previous input, we make use of RNN. RNN feeds the data from previous input into all stages, thereby helping in the sketch completion process. At the output of the first stage, that is the RNN sketch completion, the inputted sketch is completed, its boundaries are well defined and the sketch is contoured. The completed sketch is received as the output

The dataset of birds was already pretrained [2]. We drew a circle in the left-hand side. The circle was converted to the body for some images whereas head for some other. We can select the best image from the drawn doodles to give as the input for the GAN.

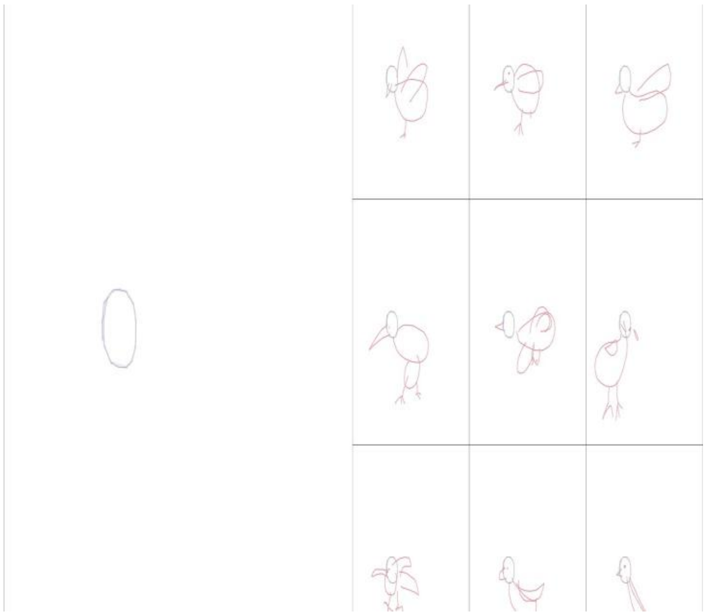

Fig 5: The output of sketchRNN

Fig 5 shows the output of sketch rnn network when we input a circle.

5.2 Image formation from sketch

For image processing we first trained the GAN network with the images and sketches of apple. We trained 100 images of apple. Later a sketch of pear was given to the network which was reshaped and resized to the shape of an apple. Fig 6 shows the image of the input given and fig 7 shows the reshaped output.

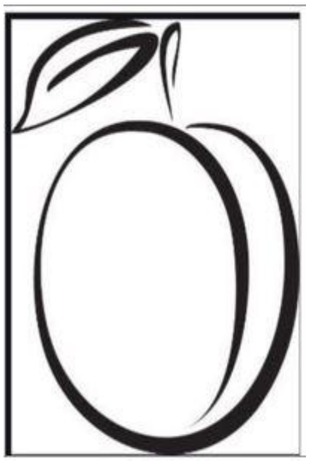

Fig 6: Input given

Fig 7: output received

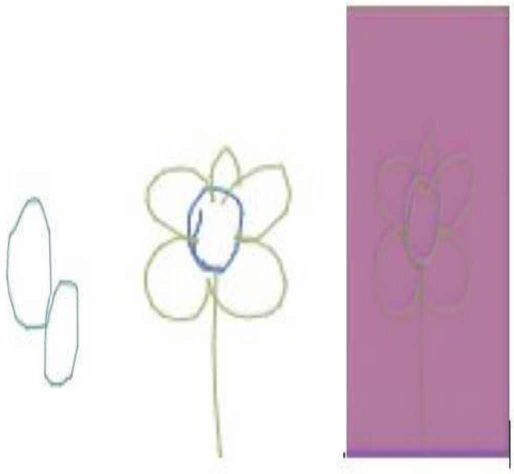

Fig 8: Output received using a flower dataset

The figure 8 shows the output received of a flower after all the 2 processes of sketch completion and image processing.

6. LIMITATION

Although the concept of our project was image processing when an incomplete sketch is given, we could not incorporate colour coding for our images. With more processed training dataset we could include the colour coding also. For this we need a graphics card implemented system.

7. CONCLUSION

In recent years, several experiments have been carried out on the issue of sketch identification. Our model illustrates the image processing if incomplete sketch is given. Later on this project can be improved by more accurate sketch completion and image processing units. For sketch completion instead of doodles accurate sketches could be produced and colourful images can be made by the GAN module by using a GPU.

REFERENCES

[1] Fang Liu, Xiaoming, Deng Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma1, Hongan Wang "SketchGAN: Joint Sketch Completion and Recognition with Generative Adversarial Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[2] DavidHa, Douglas Eck, "A Neural Representation of Sketch Drawings", arXiv:1704.03477v4 [cs.NE], April 2017.

[3] Phillip Isola, Jun-Yan, Zhu Tinghui Zhou, Alexei A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", arXiv:1611.07004v3 [cs.CV] 26 Nov 2018.

[4] Shangzhe Wu, Yu-Wing Tai, "Sketch-to-Image Generation Using Deep Contextual Completion", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[5] X.Wang and A. Gupta, Generative image modeling using style and structure adversarial networks, in Proc. Eur. Conf. Comput. Vis. (ECCV), Amsterdam, The Netherlands Oct. 2016.

[6] Ta Virot Chiraphadhanakul, "Music and art with Machine Learning", GDG DevFest Bangkok, Oct 7,2017.

8. APPENDIX

8.1 Sketch Completion

The network architecture and the pretrained dataset for sketch completion was adapted from [2].

The code for the same is available in https://github.com/magenta/magenta-demos/tree/master/sketch-rnn-js

8.2 Image formation from sketch

The code for pix2pixGAN is available in

https://github.com/anshuman73/sketch_to_image