sketch assistant : Human in the loop application to drawing sketches

Kumar Bhargav Srinivasan
Department of Computer Science
University of Colorado Boulder
Boulder, CO 80309
kusr7198@colorado.edu

Omar Hammad
Department of Computer Science
University of Colorado Boulder
Boulder, CO 80309
omar.hammad@colorado.edu

Abstract

Sketch Assistant to convert text to image using Generative adveserial network.

1 Introduction

Aim of the project is to generate image of sketch by using text as input. These types of applications can be used in application which invovle modeling or where user is not able to draw sketch. As said in the proposal, we'll be creating application which is Human in the loop based which takes in description of the image as input and generate an image, If the user isn't satisfied with image generated, he/she provides feedback via natural language instruction. This feedback is fed into network to improvise on the image generated. We are concentrating on particular set of images involving birds obtained from sketchy dataset[1] and handcraft corpus involving limited amount of topics.

1.1 Dataset Preparation

This project seemed to be intensive interms of preparing data needed for engineering. Following are the initial steps taken to clean up the data. 1) Obtained data in the svg format and generated strokes out the images. 2) generates images with incremental strokes 3) resize it to fit into the model 4) dataset given was in RGBA format which needed to be converted to grayscale.

We tried couple of approaches to annotate the data, following are the findings

- **Unsupervised learning**
  We tried to cluster strokes[2] by length/pixel density and find the group of strokes.Used Kmeans clustering to group the strokes and annotate groups. As there was no symmetery in the images drawn, the clusters generatated were not of any use. when K was low the stroke were more generic and when K was high smaller number of outlier strokes didn't contribute to drawings.

- **Supervised learning**
  Hand annotate images into 8 set of categories and use them as input to GAN, we annotated around 1000 images into 8 categories.

- **Semi-supervised learning**
  Use Handannotated images to identify classes of unannotated images. Implemented convolutional neural network to classify images into attributes. Accuracy of 0.48 was low as the unnannotated images were diverse interms of strokes.

2 Modeling

We are using Generative Adversarial Text to Image Synthesis paper as reference to device GAN model with incorporating feedback network. Current network consists of following layers:

```
_________________________________________________________________________________________________
generator:
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_4 (InputLayer)         (None, 200, 200, 1)       0
_________________________________________________________________________________________________
conv2d_3 (Conv2D)            (None, 200, 200, 64)      6464        input_4[0][0]
_________________________________________________________________________________________________
activation_5 (Activation)    (None, 200, 200, 64)      0           conv2d_3[0][0]
_________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 100, 100, 64)   0           activation_5[0][0]
_________________________________________________________________________________________________
conv2d_4 (Conv2D)            (None, 96, 96, 128)       204928      max_pooling2d_1[0][0]
_________________________________________________________________________________________________
activation_6 (Activation)    (None, 96, 96, 128)       0           conv2d_4[0][0]
_________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 48, 48, 128)    0           activation_6[0][0]
_________________________________________________________________________________________________
flatten_1 (Flatten)          (None, 294912)            0           max_pooling2d_2[0][0]
_________________________________________________________________________________________________
input_3 (InputLayer)         (None, 9)                 0
_________________________________________________________________________________________________
dense_5 (Dense)              (None, 200)               58982600    flatten_1[0][0]
_________________________________________________________________________________________________
dense_4 (Dense)              (None, 200)               2000        input_3[0][0]
_________________________________________________________________________________________________
concatenate_2 (Concatenate)  (None, 400)               0           dense_5[0][0]
                                                                   dense_4[0][0]
_________________________________________________________________________________________________
activation_7 (Activation)    (None, 400)               0           concatenate_2[0][0]
_________________________________________________________________________________________________
dense_6 (Dense)              (None, 1)                 401         activation_7[0][0]
_________________________________________________________________________________________________
activation_8 (Activation)    (None, 1)                 0           dense_6[0][0]
=================================================================================================
Total params: 59,196,393
Trainable params: 59,196,393
Non-trainable params: 0
_________________________________________________________________________________________________
discriminator
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         (None, 9)                 0
_________________________________________________________________________________________________
input_2 (InputLayer)         (None, 9)                 0
_________________________________________________________________________________________________
dense_1 (Dense)              (None, 400)               4000        input_1[0][0]
_________________________________________________________________________________________________
dense_2 (Dense)              (None, 400)               4000        input_2[0][0]
_________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 800)               0           dense_1[0][0]
                                                                   dense_2[0][0]
```

```
activation_1 (Activation)        (None, 800)           0          concatenate_1[0][0]
dense_3 (Dense)                  (None, 80000)         64080000   activation_1[0][0]
batch_normalization_1 (BatchNorm (None, 80000)         320000     dense_3[0][0]
activation_2 (Activation)        (None, 80000)         0          batch_normalization_1[0][0]
reshape_1 (Reshape)              (None, 50, 50, 32)    0          activation_2[0][0]
up_sampling2d_1 (UpSampling2D)   (None, 100, 100, 32)  0          reshape_1[0][0]
conv2d_1 (Conv2D)                (None, 100, 100, 32)  102432     up_sampling2d_1[0][0]
activation_3 (Activation)        (None, 100, 100, 32)  0          conv2d_1[0][0]
up_sampling2d_2 (UpSampling2D)   (None, 200, 200, 32)  0          activation_3[0][0]
conv2d_2 (Conv2D)                (None, 200, 200, 1)   801        up_sampling2d_2[0][0]
activation_4 (Activation)        (None, 200, 200, 1)   0          conv2d_2[0][0]
model_2 (Model)                  (None, 1)             59196393   activation_4[0][0]
                                                                  input_2[0][0]
```

2.1 Evaluation

Evaluating the quality of these systems is a hard problem. Salimans propose to use an image classifier to evaluate the quality of generated images called Inception score. The intuition behind this that we want out model to generate meaningful objects. We could use top-r metric for selecting r relevant images out of all the images generated by the network.

3 Dataset

Sketch Dataset , which contains pairs of images and sketchs for 125 categories and acquires 75,471 sketches.
Annotations generated for birds includes file head beak wing tail body leg full eyes and image_id

We took category bird from the dataset for our experiment.

4 References

[1] http://sketchy.eye.gatech.edu/
[2] Clustering Hand-Drawn Sketches via Analogical Generalization,Maria D. Chang, Kenneth D. Forbus ,Qualitative Reasoning Group, Northwestern University
[3] Generative Adversarial Text to Image Synthesis, Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran , Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA, 2016