International Journal of Scientific & Engineering Research Volume 10, Issue 8, August-2019
ISSN 2229-5518

NEURAL SKETCHING

C Vijaykumaran, Prakhar Dixit, and V R Abhinav

***Abstract*—Humans have used drawing to depict the ocular world since earliest times. Even on the present, sketching is possibly the only rendering technique readily available to all humans. Sketch-rnn, it is a special type of neural network known as the Recurrent Neural Network which is able to sketch stroke based pictures of day to day things. The model of this recurrent neural network is trained on a number of human drawn sketches which shows different classes of objects. It showcases a Structure for conditional and unconditional pictures generation, and depicts the latest robust and accurate training algorithms for producing same pictures in a vector format.**

***Index Terms*—Recurrent Neural Network, GANS, Variational Autoencoders, Generative Models, Deep Learning, AutoRegressive Models**

I. INTRODUCTION

Free hand curves have been a fundamental form of pictorial representation of ideas and objects in both historic and modern world. Sketching conveys details quickly and is capable of conveying abstract ideas easily. In today's day and age, computers and mobiles are the primary means of communication,and integrating sketching as a form of communication with a machine increases the scope of interaction between humans and computers. Our project focuses on bridging this gap between a machine and the user. For a machine to understand and identify what a human draws, it first needs to identify curves and their orientation. Furthermore, these curves need to be mapped against shapes and labels. The overall structure of the figure determines the label of the sketch. Presesnting sketch recurrent neural network (RNN) able to build stroke-based sketches of common objects. This is trained on many thousands of human-drawn drawings representing many categories. We have a urge to define a frame-work for conditional and unconditional drawing generation, and elaborate new sturdy coaching techniques for generating coherent sketch drawings in an vector format.

II. OBJECTIVE OF THE PROJECT.

The objective of our project is to establish an alternative approach to sketch recognition using neural networks. In doing so, our model aims to:

1. To build a recognizer that can effectively recognize from the noisy data
2. To Remove Outliers in order to get proper training of the classifier
3.To improve upon the current recognition system which would further help in various applications like OCR, handwriting recognition etc.

By ensuring the above stated parameters in the project, we will be able to produce an alternative method to neural sketching. All models produced have a certain level of accuracy and efficiency, we aim to use new algorithms to tackle this problem and increase accuracy by using neural networks which can handle large data sets and label data accurately. Recurrent neural networks and fiddling with their KBL loss functions along with the application of GAN and Autoencoder, we propose a system to recognize sketch drawings using neural networks.

III. DATASET

Quick, Draw!" was released as a fun game to make the public educate in a interactive way about how AI works. The game prompts users to draw an image depicting a certain category, such as " banana," "table," etc. The game generated more than 1B drawings, of which a subset was publicly released as the basis for this competition's training set. That subset contains 50M drawings encompassing 340 label categories.

```
Total number of categories:  340
Few Example Categories ['airplane', 'alarm clock', 'ambulance', 'angel', 'animal migration']
```

Fig. 1. DATASET

IV. LITERATURE SURVEY

Generative Adversarial Networks(GANs):GANs is a type of neural networks, which is combined with genetic algorithms. It is a form of Unsupervised learning. There are two main components of a GAN – Generator Neural
Network and Discriminator Neural Network. The Generator Network takes an random input and tries to generate a sample of data.It then generates a data which is then fed into a discriminator network D(x).The task of Discriminator Network is to take input either from the real data or from the generator and try to predict whether the input is real or generated. The generator tries to reduce the entropy and the discriminator tries to do the exact of the generator. They both fight with each other and finally resulting in the best solution.

Convolutional Neural Networks(CNN):They are type of neural networks which are used for mainly image classification, they take an image as input and output the trained weights in the form of vectors . They calculate the accuracy by calculating the loss of every step by backpropogation. A CNN consists mainly of convolution layer, pooling layer and a fully connected layer . It is a form of supervised learning.

V. ALGORITHM

The training algorithm follows the method of Variational Autoencoder, in there the loss function is the addition of two terms: the Reconstruction Loss, LR, and the Kullback- Leibler Divergence Loss, LKL. The training of the model to improve

this two-part loss function. The Reconstruction loss term, maximizes the log-likelihood of the generated probability distribution to explain the training data S.

VI. ARCHITECTURE DIAGRAM

1). Our model is a Sequence-to-Sequence Variational Autoencoder (VAE).
2). Our encoder is a bidirectional RNN that takes in a sketch as an input, and outputs a latent vector of size Nz.
3). We feed the sketch sequence, S, and also the same sketch sequence in reverse order, Reverse, into two encoding RNNs that make up the bidirectional RNN, to obtain two final hidden states.

VII. IMPLEMENTATION

We had 340 label categories of dataset , in which we only choose to work with few of them because of the limit of computational power and lack of storage.
We designed classifiers that could recognize the existing dataset more efficiently and in a robust manner .We build a CNN model and a LSTM model for the implementation.

Numpy: It is a framework which is used to calculate and perform mathematical and scientific process in the code.

Tensorflow: TensorFlow is a library built by the Google Brain Team to speed up machine learning and deep neural network research.

2) Steps:

1) Data Cleaning and Pre-Processing: The first step in the dataset is it has to go through a cleaning process to remove duplicate records. Next a Pre-processing step will be done has to be taken in place because the dataset contains numerical and non-numerical instances. Generally the estimator (classifier) defines in the scikit-learn works well with numerical inputs, so a one-of-K or one-hot encoding method is used to make that transformation. This technique will transform each categorical value with m possible inputs to n binary features.

```
#It is used to convert the images into numpy array so that it can be trained for model.
def drawing_to_np(drawing, shape=(28, 28)):
    # evaluates the drawing array
    drawing = eval(drawing)
    fig, ax = plt.subplots()
    # Close figure so it won't get displayed while transforming the set
    plt.close(fig)
    for x,y in drawing:
        ax.plot(x, y, marker='.')
        ax.axis('off')
    fig.canvas.draw()
    # Convert images to numpy array
    np_drawing = np.array(fig.canvas.renderer._renderer)
    # Take only one channel
    np_drawing =np_drawing[:, :, 1]
    # Normalize data
    np_drawing = np_drawing / 255.
    return cv2.resize(np_drawing, shape) # Resize array


def plot_metrics_primary(acc, val_acc, loss, val_loss):
    fig, (ax1, ax2) = plt.subplots(1, 2, sharex='col', figsize=(20,7))

    ax1.plot(acc, label='Train Accuracy')
    ax1.plot(val_acc, label='Validation accuracy')
    ax1.legend(loc='best')
    ax1.set_title('Accuracy')

    ax2.plot(loss, label='Train loss')
    ax2.plot(val_loss, label='Validation loss')
    ax2.legend(loc='best')
    ax2.set_title('Loss')

    plt.xlabel('Epochs')
```

Fig. 2. **Data Cleaning and Pre-Processing:**

2.) **Feature Selection:**Feature Selection also known as variable selection or subset extraction plays an important part here. After preprocessing of data takes place, Elimination of redundant and irrelevant data by selecting a subset of relevant features. We only used 5 classes for our model because of the limit of computational power and space resource.

3.) Build the Model CNN Model : In CNN model we got an accuracy of 84% after 60 epochs of training. LSTM Model: In LSTM Model we got an accuracy of
71.7% after 50 epochs

4.) **Prediction & Evaluation (validation):** Using the test data to make predictions of the model, here they found the Accuracy score, recall, f-measure separately for each label.

```
Colocations handled automatically by placer.
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_1 (Batch (None, None, 3)           12
_________________________________________________________________
conv1d_1 (Conv1D)            (None, None, 48)          768
_________________________________________________________________
dropout_1 (Dropout)          (None, None, 48)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, None, 64)          15424
_________________________________________________________________
dropout_2 (Dropout)          (None, None, 64)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, None, 96)          18528
_________________________________________________________________
dropout_3 (Dropout)          (None, None, 96)          0
_________________________________________________________________
cu_dnnlstm_1 (CuDNNLSTM)     (None, None, 128)         115712
_________________________________________________________________
dropout_4 (Dropout)          (None, None, 128)         0
_________________________________________________________________
cu_dnnlstm_2 (CuDNNLSTM)     (None, 128)               132096
_________________________________________________________________
dropout_5 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_6 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 2565
=================================================================
Total params: 351,153
Trainable params: 351,147
Non-trainable params: 6
_________________________________________________________________
```

Fig. 3. Model

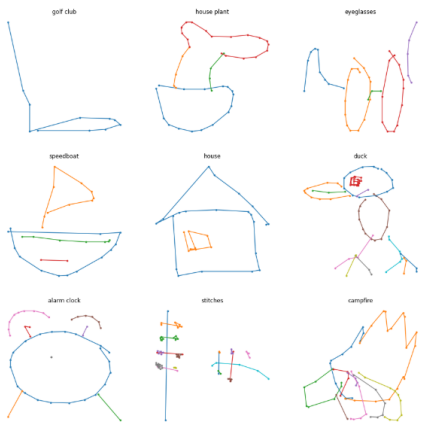

Fig. 4. OUTPUT

VIII. EXPERIMENTAL RESULTS AND DISCUSSION

A comparison study between the Convolution Neural Network and Long Short Term Memory is given below:

TABLE I
EXPERIMENTAL RESULTS

Type of model	Accracy	F SCORE	PRECISION	RECALL
CNN	84.46	0.78	0.77	0.75
LSTM	71.7	0.68	0.65	0.67

IX. CONCLUSION

In this work, we develop algorithm which could better classify the pictures. The CNN and LSTM model are able to generate possible ways to finish an existing, but unfinished sketch drawing. The model can also encode existing sketches into a latent vector, It shows how to insert in between two different sketches by adding between its latent space, and also show that we can adjust properties of an image by increasing the latent space.By making available a large dataset of sketch drawings, we hope to encourage further research and development in the area of generative vector image modeling. These methods aid in improving the accuracy of the model and making it more reliable.

X. REFRENCES

[1] kamra 2017 deep, Deep generative dual memory network for continual
learning, Kamra, Nitin and Gupta, Umang and Liu, Yan, arXiv preprint
arXiv:1710.10368, 2017
[2] Sattigeri, Prasanna, et al. "Fairness gan." arXiv preprint arXiv:1805.09910 2018
[3] Ha, David, and Douglas Eck. "A neural representation of sketch drawings." arXiv preprint arXiv:1704.03477 2017.
[4] Ha, David, and Douglas Eck. "A neural representation of sketch drawings." arXiv preprint arXiv:1704.03477 2017.
[5] Yu, Qian, et al. "Sketch-a-net: A deep neural network that beats humans."
International journal of computer vision 122.3 2017.
[6] Güçlütürk, Yagmur, et al. "Convolutional sketch inversion." European ˘
Conference on Computer Vision. Springer, Cham, 2016.
[7] Huang, Yinghsiu. "Investigating the cognitive behavior of generating idea
sketches through neural network systems." Design Studies 29.1 2008.
[8] Fu, Luoting, and Levent Burak Kara. "Recognizing network-like handdrawn sketches: a convolutional neural network approach." ASME 2009
International Design Engineering Technical Conferences and Computers
and Information in Engineering Conference. American Society of Mechanical Engineers, 2009.
[9] Perrett, David I., et al. "Frameworks of analysis for the neural representation of animate objects and actions." Journal of experimental Biology 146.1 1989.