Teach the computer to recognize freehand sketching

Fengdi Li, Yarou Xu, Yifan Wu

1. Introduction

The story of image recognition begins in 2001, the year an efficient algorithm was invented by Paul Viola and Michael Jones to identify human figures through their facial traits [1]. After that several new algorithms have been created and improved. Until 2012, deep learning algorithms became the mainstream in computer vision with its resounding success at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In this competition, Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton raised this new deep learning algorithm ensuring an 85% level of accuracy. Later, all winning entries were based on deep learning and in 2015 multiple Convolutional Neural Network based algorithms improved the whole level of accuracy in image recognition to 95%. CNN is used as the default method for dealing with images. But nowadays, some researches that have mentioned the use of Recurrent Neural Network for the image recognition [2]. Traditionally RNNs are being used for text and speech recognition because the idea behind RNNs is to make use of sequential information. RNNs perform the input sequential data by capturing and memorizing information about what has been calculated so far.

In this project, our goal is teaching the computer to recognize doodling. Unlike other images, doodling or sketching is only with and black and is composed of continuous lines. Pixels in the flattened image vector can be considered as continuous information. For each category of sketching, strokes should be similar to each other which means that there are patterns among the pixels. Therefore, we proposed a new model combining CNNs and RNNs together to classify doodling images. After training on pre-labeled sketches, we expect that the model will recognize any drawing that belongs to these pre-trained categories. Our further focus is to investigate whether the style of doodling is different between players from the U.S. and China.

2. Data

We used the sketching data provided by the Google AI game "Quick, Draw!" [3]. This dataset is a collection of 50 million drawings across 340 categories which are all contributed by the player of this game from all over the world. Each hand-drawn drawing data is a vector that tagged with metadata such as what the players are asked to draw, the player's country, and stroke positions of this drawing. We preprocessed stroke data into image matrices and uniformly rendered them into 28 ×28 grayscale bitmaps.

Since the "Quick, Draw!" gamed was built on these drawings successfully, it is the appropriate dataset to use for our new network structure. Furthermore, in order to investigate whether the style of doodling is different between players from the U.S. and China, we selected drawings which have been successfully recognized by the Google AI and only from the U.S. and China. Then train two models using U.S. and China dataset respectively. The China dataset contains 5139 drawings for 340 categories. For each category, the number of drawings is unbalanced. The whole dataset corresponding to China region was covered for training the China model. The U.S. dataset contains more than 20 million of drawings for 340 categories. Due to the computing and storage memory limits, for each category in the U.S. dataset, we trained our model on 100 and 1000 samples.

3. Approach

The proposed model has a CNN+LSTM architecture. This architecture was implemented using Keras[4]. The CNNs contain two convolutional layers with kernel size of 3 by 3, one MaxPooling layers with size of 2 by 2, one LSTM layer and two fully connected layers. It takes in a freehand sketching graph in the format of a matrix of pixel points and decrease the dimension of features by MaxPooling to extract major information. Then after flattening into one single vector, it connects to LSTM units to learn the continuous information since LSTM performs excellently in dealing with sequential data. Considering that strokes in the freehand sketching are always interdependent with each other, LSTM is a suitable tool for processing

freehand drawing data. In each step of the LSTM layer, each LSTM unit is expected to capture and memorize relationships between neighbor pixel points, where the model can better understand the sketching.

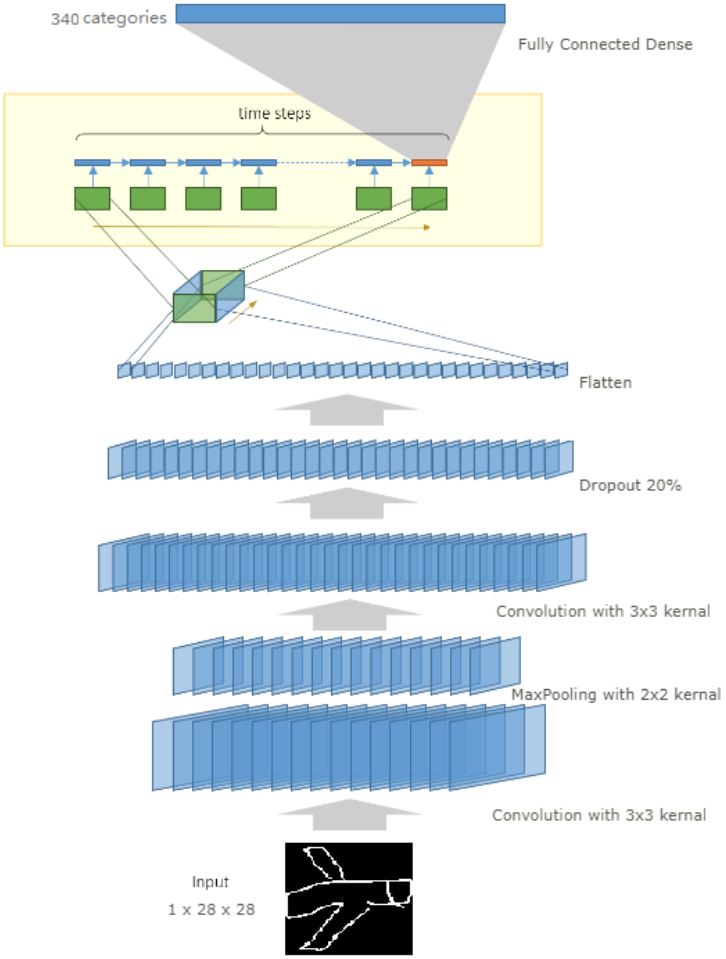

Fig 1. Structure of the CNN+LSTM model

Layer (type)	Output Shape	Param #
time_distributed_1 (TimeDist	(None, None, 26, 26, 64)	640
time_distributed_2 (TimeDist	(None, None, 25, 25, 64)	0
time_distributed_3 (TimeDist	(None, None, 23, 23, 32)	18464
time_distributed_4 (TimeDist	(None, None, 23, 23, 32)	0
time_distributed_5 (TimeDist	(None, None, 16928)	0
lstm_1 (LSTM)	(None, None, 128)	8733184
dense_3 (Dense)	(None, None, 256)	33024
dense_4 (Dense)	(None, None, 340)	87380

Total params: 8,872,692
Trainable params: 8,872,692
Non-trainable params: 0

Fig 2. Summary of the CNN+LSTM model

4. Assessment Metrics

Multi-class cross-entropy loss function integrating with the Adam optimization algorithm was used as the loss metric: Because the output of our model after the SoftMax layer is the probability of each category which ranges from 0 to 1, when the actual outputs are close to the desired outputs for all training inputs, the cross-entropy will be close to zero. Otherwise, the larger differences between the actual outputs and desired outputs, the larger cross-entropy loss. On the other hand, compared with other loss functions, such as the squared error cost, the cross-entropy cost can avoid the problem of learning rate slowing down. By computing the derivative of the cross-entropy cost with respect to the weights, we know that the derivatives are controlled by the output error. Therefore, the larger the error, the faster the model will learn.

Convolution neural net baseline model: We trained a basic CNN base model which is composed of two Convolutional layers and generated our final product by adjusting several hyperparameters, such as the kernel size, the pooling size, nodes numbers in the full connection layers. Also, we added one MaxPooling layer to decrease input dimensions and added drop-out layers to avoid over-fitting and force hidden units to learn more information.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
conv2d_2 (Conv2D)	(None, 24, 24, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 32)	0
dropout_1 (Dropout)	(None, 12, 12, 32)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 256)	1179904
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 340)	87380

Total params: 1,276,852
Trainable params: 1,276,852
Non-trainable params: 0

Fig 3. Summary of the basic CNN model

Training-Validation Split & Cross Test between countries: In this project, we trained two models for U.S. and China respectively using data of corresponding country. During the training process, the ratio of the validation set over the training set is 2:8. Then we cross tested these two models with the data of the other county which means that we tested the U.S model using the China images and vice versa.

5. Results

In order to obtain the best validation and test performance of both CNN and CNN+LSTM models, early stopping was not used. For the China model, the CNN+LSTM model has much better performance on training accuracy than the basic CNN model. However, its validation and testing accuracy is lower than the basic CNN model.

For the U.S. model, with 100 drawings per category data, it has similar results as the China model. However, for both the basic CNN and CNN+LSTM model, the validation and testing accuracy increased. When the training sample size increased to 1000 drawings per category, our LSTM+CNN model outperforms the

basic CNN model with 54% validation accuracy and 49% testing accuracy. Considering that there are 340 categories, this result is acceptable.

Model	**U.S.**				**China**	
Training Size	100/Category		1000/Category		All	
Accuracy	**CNN**	**CNN+LSTM**	**CNN**	**CNN+LSTM**	**CNN**	**CNN+LSTM**
Training	54%	99%	33%	62%	89%	99%
Validation	41%	33%	47%	54%	17%	11%
Testing set	China data				U.S. 100/Category data	
Cross Test	38%	31%	43%	49%	14%	12%

Table 1. Results for the basic CNN model and LSTM+CNN model

6. Discussion

By visualizing the aircraft as an example category of doodle images from two countries, we observed that China's doodling is more variant than U.S., which indicates that Chinese people has different drawing styles from Americans. Using China model to test on the U.S images will produce a slightly lower accuracy, vice versa.

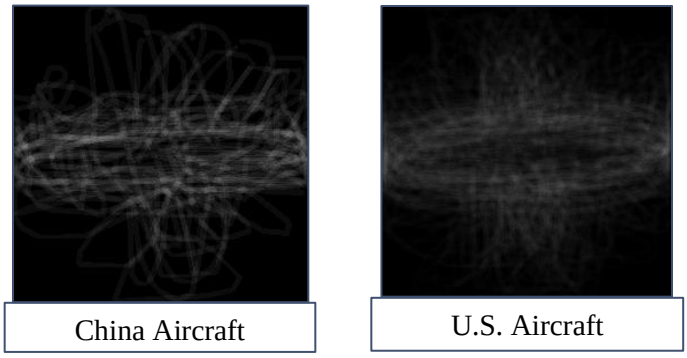

Fig 4. Mean images for the China and U.S. aircraft doodling

Since this model only trained on black-and-white freehand sketching, therefore, it may not be able to classify other types of images accurately, such as realistic photographs with RGB colors. Besides, either the basic CNN model or the CNN+LSTM model has the over fitting problem. But with increasing training data size, the overfitting problem ameliorates, especially for our CNN+LSTM model.

The performance of proposed model is deeply dependent on the quality and size of the training data. Compared with the Chinese drawing data, the US drawing data is much balanced, therefore, the overfitting problem is alleviated, and the validation accuracy increased.

Due to the computation and storage limitation of Google Colab, the U.S. model only trained on a subset of examples of each image categories. However, when the sample size increased from 100 to 1000, the validation and test accuracy greatly improved. We suppose that with further larger training size, the LSTM+CNN model will have even better performance.

The converging speed of CNN+LSTM is a lot faster than basic CNN model, which achieved a comparably high training accuracy within few epochs in all tested circumstances. For example, when training the U.S.

model with 1000 drawings per category data, the CNN model used 20 epochs to get 30% training accuracy, which was obtained at the 2nd epoch for the LSTM+CNN model.

7. Conclusion

The CNN-LSTM model, built in this study, can be used to identify U.S. and Chinese doodle images belongs to 340 common categories with pretty high training accuracy and worse in validation and cross test accuracy. However, with increasing training data size of U.S. data, the CNN+LSTM model performs better and converge much faster than the basic CNN model.

There are slightly different between U.S. and China in doodling drawing styles. Therefore, training on one country dataset will has slightly lower testing accuracy on the other country dataset.

8. Acknowledgements

We would like to thank every participant playing the 'Quick, Draw!', which form the foundation of this study. Also, we appreciate Dr. Joshuah Touyz for the guidance of deep learning theories and technique applications.

Reference:

[1] Vikas Gupta (2017, December 26). Keras Tutorial: Using pre-trained imagenet models. Retrieved from https://www.learnopencv.com/keras-tutorial-using-pre-trained-imagenet-models/

[2] Recurrent Neural Networks for Drawing Classification | TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/tutorials/sequences/recurrent_quickdraw

[3] Googlecreativelab. (2018, November 26). Googlecreativelab/quickdraw-dataset. Retrieved from https://github.com/googlecreativelab/quickdraw-dataset

[4] Brownlee, J. (2017, July 19). CNN Long Short-Term Memory Networks. Retrieved from https://machinelearningmastery.com/cnn-long-short-term-memory-networks/