Doodle Translator - A Computer Vision Project

Manish Kumar, Utkarsh Sharma, Prutyay Gautam
2018047, 2018114, 2018403
Indraprastha Institute of Infromation Technology, Delhi

Abstract

The aim of our project is to develop a tool that identifies hand drawn doodles. This includes developing a algorithm that can extract a clean doodle from images of hand drawn doodles on various types of paper and classifying them into different categories. We developed a heuristic algorithm to solve the first challenge. We used Google's Quickdraw dataset to train our CNN classifier. We made a dataset of approximately 90 images of hand drawn doodles and tested our algorithm on it. We were able to achieve satisfactory conversion in most images and classification accuracy comparable to images from Quickdraw dataset.

1. Introduction

It's highly likely that we all have doodled at one point in our lives. Some might consider doodling as just something that a child might do. Although, according to some researchers, doodling helps relieve boredom and frustration and the urge to doodle gets stronger as stress levels rise. Doodling is like a safety valve that allows pressure to be dispelled in a playful and creative way. For several researchers, it is an interesting topic of study. They interpret the doodles of individuals and try to formulate a correlation between the doodle and the state of mind of a person. Therefore, being intrigued by this field of doodling we sought to find a way to decipher and translate these doodles. The ability to detect and classify hand-drawn doodles has far-reaching implications for artificial intelligence research in general. For example, the development of a robust classifier on high-noise datasets will considerably aid research in computer vision and pattern recognition, particularly in subfields like Optical Character Recognition (OCR).

The aim of our project is to develop a tool that identifies hand-drawn doodles. Our tool will input the images of drawings with doodles and tell us the category in which it lies, for example, an airplane, arm, ear, apple, axe, bird etc.

Our project is majorly divided into two tasks:-

1. Making an accurate image classification model on a subset of the QuickDraw dataset.

2. Extracting the doodle from an image of a drawing on a piece of paper or a notebook for real-life usage and demonstration.

2. Related Works

One of the most fundamental tasks in computer vision is sketch recognition [1], which seeks to predict the class label of a given sketch. It can be used for a variety of purposes, such as interactive drawing systems that provide feedback to users [16], sketch-based science teaching [2], games [3], [4], and so on.

There are two types of sketch recognition settings: offline and online. Offline recognition algorithms use the entire sketch as an input and anticipate a class label based on it. During sketching, online recognition algorithms use the accumulated drawing strokes to anticipate the class label in real time. Although offline identification methods are more prevalent, online recognition methods can be utilised in more interactive real-time applications such as real-time drawing guidance [5], tracing, and interactive sketch retrieval.

Year	Model	Architecture	Layers	Params	Ensemble	Pretrain	Input	Preprocess	Dataset	Accuracy
2015	Sketch-a-Net [10]	CNN	5 conv.	8.8M	✓		picture	augm. [10]	TU-Berlin [6] 250 cat.	0.7490
2016	AlexNet-FC-GRU [99]	CNN-to-RNN cascaded	–	–			stroke accu. pic.	–	TU-Berlin 160 cat.	0.8510
2018	SketchMate [19], [100]	RNN	2 GRU	–			stroke vector	tru. & pad. [19]	QuickDraw 3.8M [19]	0.7788
2018	SketchMate [19], [100]	CNN-RNN dual-branch	5 conv. & 2 GRU	–			picture & stroke vector	tru. & pad. [19]	QuickDraw 3.8M [19]	0.7949
2017	Jia *et al.* [101]	RNN-RNN dual branch	–	–	✓	✓	CNN features of stroke accu. pic.	reflection, rotation, etc.	TU-Berlin	0.9220
2017	DVSF [102]	R-FC and RNN dual branch	–	–		✓	CNN features of stroke accu. pic.	–	TU-Berlin	0.7960
2018	FBin DAB-Net [103]	binary CNN	–	–			picture	–	TU-Berlin	0.7370
2018	RNN−CNN [104]	RNN-to-CNN cascaded	2 LSTM & 5 conv.	–		✓	stroke vector	augm. [3]	TU-Berlin	0.7849
2019	multi-graph tran. [23]	GNN	4 tran.	10M			stroke vector	–	QuickDraw subset [23]	0.7070

Figure 1. Comparison of representative sketch recognition networks

Figure 1 lists various networks that are engineered for freehand sketches and capable of sketch recognition. We next introduce some representative networks. The first deep CNN built for free-hand sketching was Sketch-a-Net [6], [7].

Sarvadevabhatla et al. [8] proposed a sketch recognition network that takes advantage of the sequential process of sketching by plotting each training sketch as a continuous sequence of cumulative stroke pictures and sending the corresponding AlexNet [9] based deep features into a Gated Recurrent Unit (GRU) [10] network in order.

He et al. [102] proposed the deep visual sequential fusion (DVSF) net for simultaneously capturing spatial and temporal patterns of sketching.

In 2017, Ha and Eck developed the ground-breaking SketchRNN [11], which uses a Sequential Sketch Generation Model based on Variational Inference (VI) [12] to conduct representation learning.

Xu et al. proposed the SketchMate [13] sketch hashing network, which features a CNN-RNN dual branch network as its backbone, which uses CNN to extract abstract visual concepts and RNN to mimic human temporal stroke order.

In topological space, a drawing can be represented as sparsely linked graphs. The Multi-Graph Transformer (MGT) [14] is a GNN model that uses sketch graphs to learn both geometric structure and temporal information.

3. Methodology

The aim of this project was to provide a method to detect paper drawn doodles.

We needed a dataset of hand drawn doodles in different kinds of papers for the project. As no dataset was available, we made the dataset ourselves. We made a total of approximately 90 images of 10 different objects on 3 kinds of paper. The number of doodles in each category is available in Figure 2. The dataset is now available on our project's Github page.

Airplane	9
Apple	9
Arm	9
Axe	9
Banana	9
Bat	7
Bicycle	9
Bird	11
Bowtie	8
Broom	8

Figure 2. Distribution of hand drawn doodles over 10 categories

We also needed a dataset of clean doodles for training our DL model. We used Google's Quickdraw dataset for this. The Quick Draw Dataset is a collection of 50 million drawings across 345 categories, contributed by players of the game Quick, Draw!. We took a sample of 200,000 images over 10 categories for our project.

We used a multi-layer CNN model to train our classifier with Quick Draw dataset. A CNN model is used extensively in modern Machine Learning applications due to its ongoing record breaking effectiveness [17]. We referred to different implementations of CNN on the QuickDraw dataset and experimented with model parameters like number of layers. The final architecture which was performing the best was chosen for the set task. Final CNN architecture is shown in Figure 3.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_32 (Conv2D)           (None, 28, 28, 16)        160

 conv2d_33 (Conv2D)           (None, 28, 28, 16)        2320

 max_pooling2d_16 (MaxPoolin  (None, 14, 14, 16)        0
 g2D)

 conv2d_34 (Conv2D)           (None, 14, 14, 32)        4640

 conv2d_35 (Conv2D)           (None, 14, 14, 32)        9248

 max_pooling2d_17 (MaxPoolin  (None, 7, 7, 32)          0
 g2D)

 conv2d_36 (Conv2D)           (None, 7, 7, 64)          18496

 conv2d_37 (Conv2D)           (None, 7, 7, 64)          36928

 max_pooling2d_18 (MaxPoolin  (None, 3, 3, 64)          0
 g2D)

 dropout_6 (Dropout)          (None, 3, 3, 64)          0

 flatten_6 (Flatten)          (None, 576)               0

 dense_12 (Dense)             (None, 512)               295424

 dense_13 (Dense)             (None, 10)                5130

=================================================================
Total params: 372,346
Trainable params: 372,346
Non-trainable params: 0
```

Figure 3. CNN Model Layers

To get a clean 28*28 pixels doodle out of the clicked pictures of hand drawn doodles, we used a combination of HSV processing, OTSU, Morphological operations, Bounding box detection, and image compression techniques. The approach is further expanded upon in Figure 4.

We converted the image to HSV space as it is advantageous to use HSV space when you have to match colors, or problematically determine if one color is similar to another color. [18] As evident in Figure 5, the doodles are clearly highlighted in either the Hue or Transparency space. For choosing between the two spaces we applied a self made heuristic approach. Both these spaces were treated as a greyscale image and difference between the frequency of black and white pixels were computed. If the ratio of minimum of the two by the maximum of the two was greater than 0.8 i.e. had a similar difference, then the space having the smaller difference was taken forward else the space having greater difference was considered. Through this approach we were successfully able to choose the space which comparatively highlighted the doodle better for the extraction.

For the chosen space we treat it as a greyscale image and applied 'open' morphological function(erosion followed by dilation) for noise reduction as it erases the salt and pepper noise in our image without affecting the doodle too much.

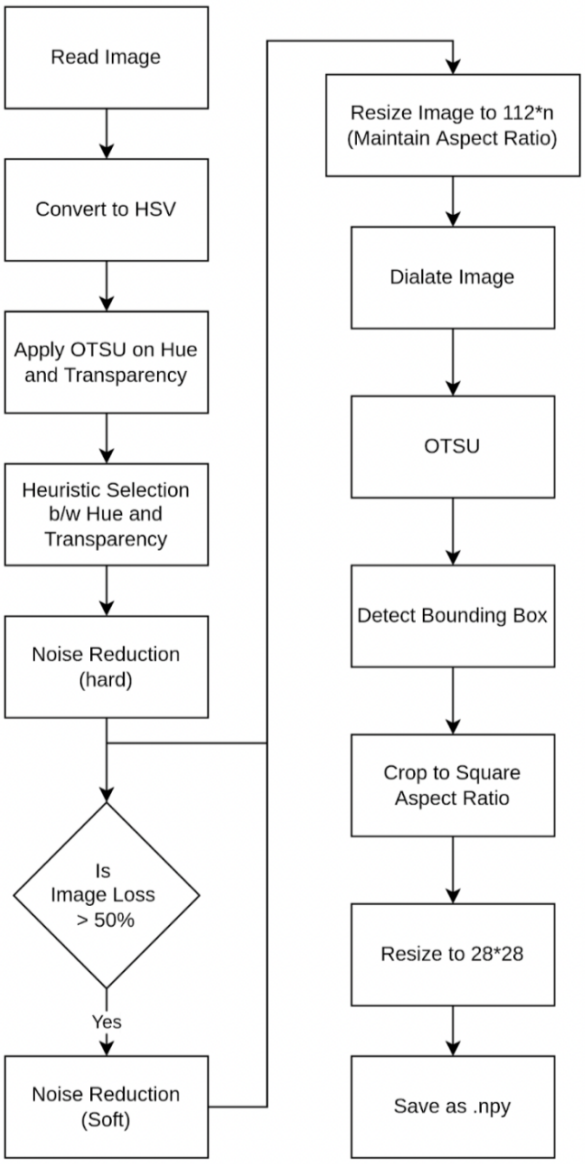

Figure 4. Processing Workflow on Hand drawn doodle images to get a 28*28 clean numpy array

We also use the 'dilate' morphological function when resizing the image to preserve connected components.

4. Experiments

4.1. CNN Model Performance:

The dataset for our project consists of 20,000 samples each of 10 different doodle categories for the QuickDraw database. Upon using 70 percent data for training and 30 percent data for testing the performance metrics for our final model are shown below in Figure 6.

Upon analysing the results we can infer that the model

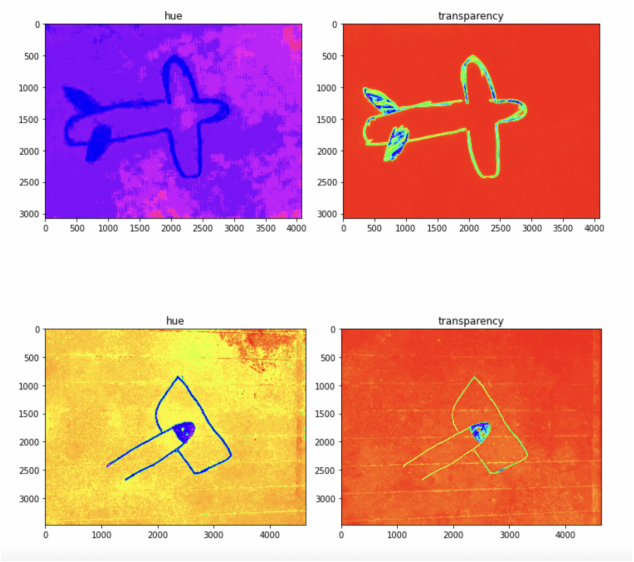

Figure 5. Hue and Transparency of Airplane and Axe

is 90 percent accurate with the balanced accuracy exactly being 90.85 percent. The Bat doodle category is the one the model is facing the most difficulty in detecting. While it is easily able to detect doodles of Apples. Overall, the model is performing efficiently as inferred from the MCC score which is almost 90 percent and refers to the results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives). Also the model is reliable and valid according to the Cohen Kappa Score which is 89.32 percent.

	precision	recall	f1-score	support
0	0.83	0.96	0.89	5259
1	0.97	0.98	0.98	5933
2	0.90	0.82	0.86	6643
3	0.94	0.91	0.92	6132
4	0.95	0.89	0.92	6417
5	0.88	0.76	0.81	7050
6	0.96	0.99	0.97	5796
7	0.75	0.89	0.82	5038
8	0.92	0.94	0.93	5849
9	0.95	0.94	0.94	5883
accuracy			0.90	60000
macro avg	0.90	0.91	0.90	60000
weighted avg	0.91	0.90	0.90	60000

Cohen Kappa Score: 0.8932750035207859

MCC Score: 0.8937071619288989

Balanced Accuract: 0.9085444803749034

Category	Code
Airplane	0
Apple	1
Arm	2
Axe	3
Banana	4
Bat	5
Bicycle	6
Bird	7
Bowtie	8
Broom	9

Figure 6. CNN Model Evaluation

4.2. Doodle Extraction

For this experiment we made 87 doodles using different sheets of paper (Having different linings), different pens and markers in order to get the performance of our model on real hand-drawn doodles. Our first task was to extract these doodles from the sheets of paper. Using the preprocessing steps as mentioned in the methodology 84 out of the 87 doodles

were extracted with acceptable results. The extracted doodles were then resized to the resolution of 28x28(Same as that of the QuickDraw database) as shown in Figure 7.

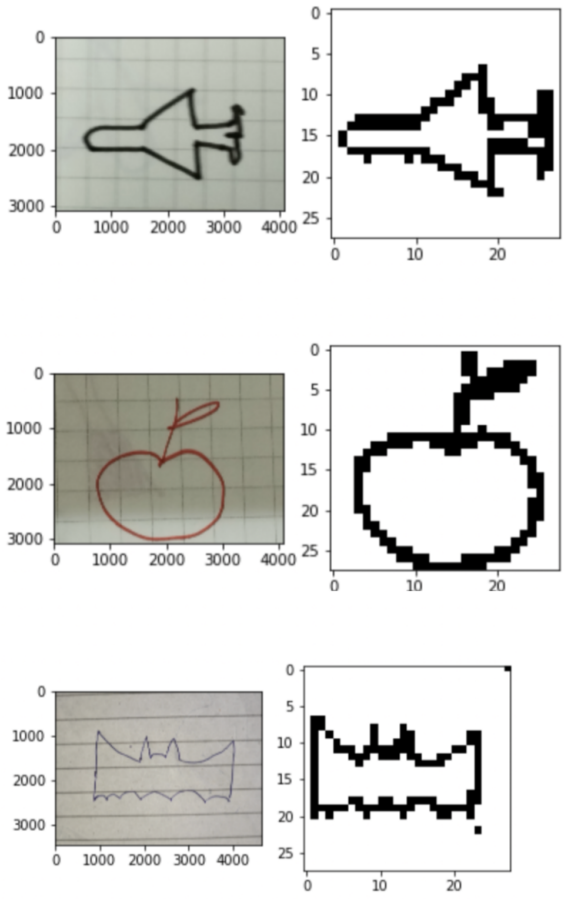

Figure 7. Examples of Hand drawn images and their 28*28 doodle counterparts

4.3. Hand-Drawn Doodle Classification

When these hand-drawn doodles were classified using our CNN model the obtained metrics were as shown in Figure 8.

Our metrics show us that the model is 78 percent accurate with the balanced accuracy exactly being 79.73 percent. The Airplane doodle category is the one the model is facing the most difficulty in detecting the hand-drawn doodles. While it is easily able to detect doodles of Bowtie and Broom doodles. The MCC score also tells us that the model has an acceptable overall performance with the score being 76.11 percent. Also, the model is acceptably reliable and valid according to the Cohen Kappa Score which is 75.76%.

The comparison of Hand drawn images and Quickdraw

	precision	recall	f1-score	support
0	0.33	0.60	0.43	5
1	0.89	1.00	0.94	8
2	0.75	0.55	0.63	11
3	0.78	0.88	0.82	8
4	0.78	0.78	0.78	9
5	0.86	0.50	0.63	12
6	0.89	0.80	0.84	10
7	0.64	0.88	0.74	8
8	1.00	1.00	1.00	8
9	1.00	1.00	1.00	8
accuracy			0.78	87
macro avg	0.79	0.80	0.78	87
weighted avg	0.81	0.78	0.78	87

Cohen Kappa Score: 0.7576601671309192

MCC Score: 0.7611204817080797

Balanced Accuract: 0.7973232323232323

Category	Code
Airplane	0
Apple	1
Arm	2
Axe	3
Banana	4
Bat	5
Bicycle	6
Bird	7
Bowtie	8
Broom	9

Figure 8. Hand-Drawn doodle classification evaluation

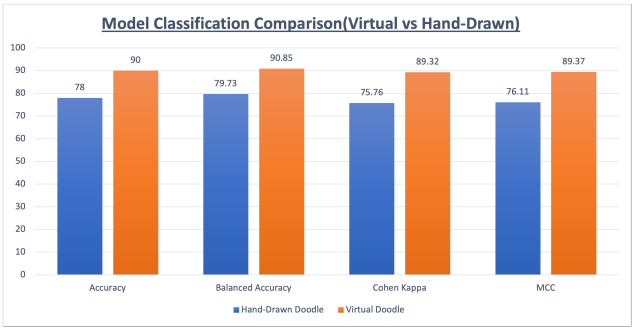

Figure 9. CNN Model Classification Comparison

images can be seen in Figure 9. We can observe that our model performs comparative to base metrics (metrics on Quickdraw datasets) and thus our extraction algorithm is effective.

5. Conclusion

We started with the aim of classifying images of hand drawn doodles on different types of papers. In this paper, we proposed a heuristic algorithm to extract the doodle from images and classify them using a CNN classifier. We were able to achieve satisfactory conversion in most images and achieve classification accuracy of our dataset comparable to images from Quickdraw dataset.

In future, we will expand the classifier to include more categories of doodles and train them on more samples. We would also like our algorithm to cover more types of paper like multi-color lined papers. The possibility of making a gamified application for the algorithm can also be explored.

6. References

[1] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" TOG, 2012.

[2] K. D. Forbus, B. Garnier, B. Tikoff, W. Marko, M. Usher, and M. McLure, "Sketch worksheets in stem classrooms: Two deployments," in AAAI, 2018.

[3] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," IJCV, 2017.

[4] R. K. Sarvadevabhatla, S. Surya, T. Mittal, and V. B. Radhakrishnan, "Pictionary-style word-guessing on hand-drawn object sketches: dataset, analysis and deep network models," TPAMI, 2020.

[5] J. Choi, H. Cho, J. Song, and S. M. Yoon, "Sketch-helper: Real-time stroke guidance for freehand sketch retrieval," TMM, 2019.

[6] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" TOG, 2012.

[7] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net that beats humans," in BMVC, 2015.

[8] R. K. Sarvadevabhatla, J. Kundu et al., "Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition," in MM, 2016.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton,"Imagenet classification with deep convolutional neural networks," in NeurIPS, 2012.

[10] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, ¨ F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[11] D. Ha and D. Eck, "A neural representation of sketch drawings," in ICLR, 2018.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[13] A. Kotani and S. Tellex, "Teaching robots to draw," in ICRA, 2019.

[14] P. Xu, C. K. Joshi, and X. Bresson, "Multigraph transformer for free-hand sketch recognition," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[15] Peng Xu, Timothy M. Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang, "Deep Learning for Free-Hand Sketch: A Survey", in IEEE TPAML 2022.

[16] Y. Matsui, T. Shiratori, and K. Aizawa, "Drawfrom-drawings: 2d drawing assistance via stroke interpolation with a sketch database," TVCG, 2016.

[17]Hussain, M., Bird, J.J. and Faria, D.R., 2018, September. A study on cnn transfer learning for image classification. In UK Workshop on computational Intelligence (pp. 191-202). Springer, Cham.

[18] Cardani, D., 2001. Adventures in hsv space. Laboratorio de Robótica, Instituto Tecnológico Autónomo de México.