International Conference on Innovative Research in Engineering, Management and Sciences
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 4 | Issue 9 | ISSN : 2456-3307

Recognition of Labels for Hand Drawn Images

Teena A James*[1], Darshan Kothawade[2]

*[1] Computer Science Dept. New Horizon College of Engineering, Karnataka, India
teenasunny12@gmail.com[1]
[2] Computer Science Dept. Pillai College of Engineering, Maharashtra, India
darshan169@student.mes.ac.in [2]

ABSTRACT

Freehand sketch drawings are highly abstract and sparse in structures. Due to the diversity, highly iconic and intra-class deformations of these sketches, automatic recognition is more a challenging task. This paper, sheds light on developing an efficient recognition scheme of freehand sketch, based on Convolutional Neural Networks (CNNs). Furthermore, this paper seek to classify Google's 'Quick, Draw!' dataset sketches which contains more than 50 million drawings across 345 categories by creating a Keras model. It aim to integrate a custom model to an Android app using Tensor flow Lite. Such a system will outperform for variety of applications, such as human-computer interaction, sketch-based search, game design, and education.

Keywords : Component, Formatting, Style, Styling, Insert (Key Words)

I. INTRODUCTION

Google's experimental game Quick, Draw! Doodle was framed to educate public about artificial intelligence. In this game, user are asked to immediately draw an image of depicted category and at the other end neural network attempts to effectively recognize the image that is either incomplete or doesn't match with any existing labels. Other significant areas whereby doodle can span includes computer vision and pattern recognition, sketch-based search, game design, and education, which works on highly noisy datasets etc. This project focuses on classification of doodles which is in as such a tedious task because of numerous categories, variations and same resemblance at large and whose output is the predicted category for the depicted object.

II. RELATED WORK

Variant outlook to achieve efficiently better results has been made by computer vision in different applications. Eitz et al. [1] successfully demonstrated achievement of classification rates for computational sketch recognition by using feature vectors, bag of words sketch representation and SVMs to classify sketches. Schneider et al. [4] then modified the benchmark by making it more focused on how the image should like, rather than the original drawing intention, and they also used SIFT, GMM based on Fisher vector encoding, and SVMs to achieve sketch recognition.

In general, the majority of the earlier systems used to pull out sketch features which is thereby fed to a classifier. Convolutional neural CNN) have emerged as a vigorous framework for feature representation and recognition [3]. These type of neurons

biologically inspired feed-forward artificial neural network composed of multiple layers of neurons, and the neurons in each layer are then collected into sets. At the input layer, where the data gets introduce to the CNN, these neuron sets map to small regions of input image. Deeper layers of the network can be composed of local or global pooling (fully-connected) layers which combine outputs of the neuron sets from previous layer. Pooling is achieved through convolution-like operations. Deep neural networks (DNN), especially CNNs, are trained to automatically learn features instead of manually extracting features and likewise its multi layers learning can get more effective expression. When it comes to CNN design, the inclination in the past few years has pointed in one direction: deeper networks [3]. This move towards deeper networks has been beneficial for many applications. Amongst most outstanding application has been object classification, where the deeper the neural network, the better the performance. However, existing CNNs are designed for photos, and they are trained on a massive amount of data to avoid overfitting. Traditional CNNs are limited in depth, as empirical results showed that training error increased with depth, suggesting that deeper networks become increasingly hard to train. This problem was addressed by He et al. with the introduction a deep residual network architecture, which uses shortcut connections to allow convolutional layers to approximate residuals rather than actual mappings [2]. Their model was able to set new records for both the ImageNet and COCO datasets, and through the application of residual networks, CNNs with over a thousand layers have been trained.

III. METHODOLOGY

A. Convolutional Neural Network

An input image after processing is classified by CNN under certain categories. For CPU, an input image is an array of pixels and it depends on the image resolution.Thereafter, each input image will pass through a series of convolution filter layers(kernels), followed with feature extraction, also known as pooling all connected layers and implement softmax function to classify an object with probabilistic values between 0 and 1.

Fig 3.1 Neural network with many convolutional layers

Stride is the number of pixel shifts over the input matrix. If the stride is 1 then move the filters to 1 pixel at a time. If the stride is 2 then move the filters to 2 pixels at a time and so on. Occasionally when filter does not perfectly fit the input image, it allows to have two options:

- Pad the picture with zeros (zero-padding) so that it fits.
- Drop the part of the image where the filter did not fit and keep only valid part of the image called as valid padding.

Rectified Linear Unit for a non-linear operation is used to introduce non-linearity in our ConvNet. The output is

$f(x) = \max(0,x)$.

For larger images, pooling layers section would reduce the number of parameters. Max pooling and average take the largest element from the rectified feature map.

B. Proposed System Architecture

For a $28 \times 28 \times 1$ doodle, first run the image through two convolutional filters. Zero padding border around the image were added so that resultant outcome have same width and height. The output then goes through a max pooling layer with a kernel size of 2×2. Subsequently, it will flatten the tensor and feed the result through two fully-connected or dense layers. Every layer utilizes the ReLu activation function as well as dropout. The outcome is then passed through one more affine transformation before applying softmax to generate probabilities for each class.

IV. SIMULATION AND RESULTS

Quick, Draw! dataset containing over 50 million images across 345 categories was openly released by Google with multiple different representations for the images. One dataset represents each drawing as a series of line vectors, and another contains each image in a 28x28 grayscale matrix. Since focus is on classification of the entire doodle, the latter version of the dataset is used. We treat each 28x28 pixel image as a 784 dimensional vector

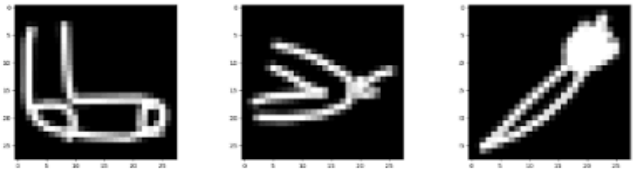

Fig 4.1 Sample doodles of a sock, elbow, and carrot (left to right) from the training dataset

For testing our models, the data got split into two different folds: 80% for training and 20% for testing10% randomly sampling of the drawings categorically were created to reduce computation time and storage of the data. As a result, obtained approximately 4,000 examples for the training set and 1,000 examples for the testing set Evaluation Parameters

Although accuracy penalizes harshly for an incorrect prediction (wrong predictions receive 0 points and right predictions receive 1 point), it is a good measure to detect performance. As it has so many categories, including some that are highly analogous, it evaluates methods not only with accuracy but also with a scoring metric that is more tolerant of errorneous predictions. Top_k_categorical_accucary metric provided by keras calculates the top-k categorical accuracy rate

A. Performance Evaluation

Best performance for the CNN was accomplished by tuning various hyper parameters including the number of units in each dense layer, dropout rate, and learning rate. Mostly, found that the model yielding the best prediction had two dense layers with 512 and 256 units with each layer having a dropout rate of 0.2. Furthermore, it has trained model with batch size of 256 across 10 epochs.

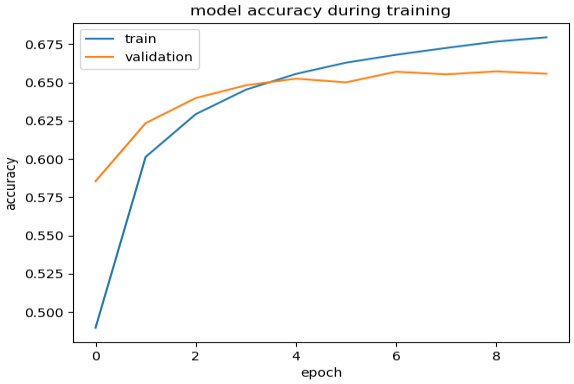

Fig 4.2 Training Accuracy

As seen from figure 4.2, the end architecture fits the data well as the validation accuracy has more or less converged after the 6th epoch. Furthermore, following were the accuracies achieved on the testing dataset:

1. Final accuracy: 65.57%
2. top-3 accuracy: 82.71%

Fig 4.3 Sample of Input/Output

We would like to experiment with advanced CNN architectures such as VGG-Net and ResNet, which have already reached state-of-the-art levels of image classification performance, although not for sketches in particular. Additionally, we have only used approximately 10% of the total Quick, Draw! dataset, and we believe training our models on the complete dataset would improve accuracy.

V. REFERENCES

[1] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects ACM Trans. Graph. (Proc. SIGGRAPH), 31(4):44:1– 44:10, 2012

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[4] R. G. Schneider and T. Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. ACM Transactions on Graphics (TOG), 33(6):174, 2014