# Testing strength of the state-of-art image classification methods for hand drawn sketches

Ochilbek Rakhmanov
*Computer Science Department,*
*Nile University of Nigeria,*
*Abuja, Nigeria*
ORCID: 0000-0001-6768-234X
ochilbek.rakhmanov@nileuniversity.edu.ng

***Abstract*— Classification of hand drawn sketches (images) reached a classification accuracy of %77 with the latest state-of-the-art method, called Sketch-a-Net, in 2017. Most of the developed methods use image feature extractor techniques like HOG, BOVW, or CNN. In this paper, we tested the classification accuracy of hand drawn sketches with SVM and ANN, without using image feature extraction algorithms and compared the results with the findings of a number of important state-of-art researches. Our findings show that existing methods are reasonable to accept, even though the results of our experiments also produced some valuable results. We propose that our findings can serve as kind of 'minimal milestone' on future prediction experiments.**

***Keywords—hand drawn image, classification, machine learning, deep learning.***

## I. INTRODUCTION

The new era in machine learning and image classification has brought some new ways of classifying not only real life images, but also images produced by humankind. The rapid technological advancement over the past few years has brought about numerous new opportunities. For instance, it has become possible to sketch not only on paper, but also touchscreens, tablets, phones and other devices. Thus, research on sketches has flourished in recent years, attracting many researches to conduct experiments on the classification of sketches.

Recognizing free-hand sketches is a very difficult task (see Fig. 1). This is due to a number of reasons:

- Sketches can be very abstract and deformed, but still represent the same object. The different sketches representing the face of girl and bicycle in the Fig.1 are good examples of this.
- Style and lines of the figure will definitely change with respect the person drawing it. However, while some people may draw all features of the object, others may miss some features.
- The sketches lack colors; they are usually in black and white representation.

There are a number of existing methods and algorithms that are helpful in the image classification task. Examples include threshold value, dilation/erosion, and edge detection, among other image processing methods. K means, SVM, ANN, and CNN are some popular classification methods being used in the machine learning field. Lastly there are some useful feature extractor algorithms in the computer vision field like HOG or BOVW.

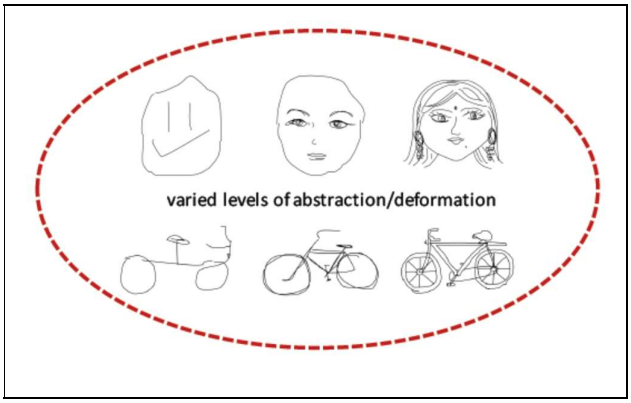


Fig.1. Different representation of face and bicycle [5].

Most of the aforementioned concepts were used by researchers in their experiments to classify hand drawn sketches.

However, none of the previous researchers presented straight forward results on the classification of sketches, using the pixel value of the picture as input, without using computer vision algorithms. It was probably thought that this method will give very low accuracy; however, if implemented effectively, it would serve as a good milestone to compare with new methods.

## II. PURPOSE OF THE STUDY

- To test the classification accuracy of hand drawn images by using only pixel value information, with algorithms like SVM and ANN.
- Compare the results of our study with the findings of current state-of-art methods to determine how much they differ in terms of prediction accuracy.
- To propose our findings as kind of 'minimal milestone' for future prediction experiments.

## III. BACKGROUND AND RELATED LITERATURE

There are many researches done in this field in recent 5-6 years. But we have chosen only milestone papers during literature review and tried to develop our methodology based on their findings. This challenging task gained popularity in 2012, when Eitz *et al* released a dataset of hand drawn images with 20,000 samples (250 different objects containing 80 samples each) [1]. As machine learning was

on the verge of exploration of neural networks in 2012, they used the best possible method to classify the images, Support Vector Machines (SVM), after the calculation of Histogram of Oriented Gradients (HOG) for each picture. This resulted in a %56 of classification accuracy. What makes this research important is that they made the dataset available (open source) to everyone on the internet. This triggered many other researches in this field.

In the following three years, after release of up mentioned dataset, a number of significant studies were conducted by the group of researchers led by Li Y. [2,3]. These researchers used ensemble learning and multi-kernel image processing in their studies, which enabled them to increase classification accuracy to %68.

When the CNN became one of most powerful image classifiers, Yu *et al* proposed a CNN structure for hand drawn image classifications in 2015 [4]. This proposed CNN structure reached a new height of %74 of classification accuracy. Two years later, the same group proposed an updated version for their CNN structure [5], with some additions, and managed to increase accuracy by %3. This figure remains the best result to this day.

As we stated above, the main goal of our experiment was to compare some existing state-of-art classification methods results with our own results. So we accepted three papers ([1], [2] and [5]) as leading research papers in this field and compared the results from our experiment with them.

## IV. Instruments for Experiment

Python programming language was used during the experiment. We used open source Scikit-Learn library during the machine learning training, and all statistical analysis results were produced with help of tools from this library [6,7]. All image processing jobs were done using OpenCV library [8]. During the neural network training, we used Keras library, with Tensorflow backend support [9].

## V. Experiment and Results

The methodology section consists of three parts:

1. Image selection and image processing.
2. Testing accuracy with SVM.
3. Testing accuracy with ANN.

### *A. Image selction and image processing*

Our first step was to prepare our dataset. As stated above, [1] dataset consists of 250 different sets, with 80 samples in each set. We formed 2 datasets. The 1st dataset contains all 250 classes, while the 2nd dataset contains one-fourth of dataset (62 selected sets) to minimize computational expense. Minimization of dataset was also used by Eitz *et al* during their research [1]. Fig.2 presents some samples of the 2nd dataset. It is easy to observe how an object can vary when it is drawn by a human.

Next step involved passing images through some process. Every picture was converted to binary format (only black and white). Subsequently, colors were inverted to make calculation easier as a pixel value of zero (black) will save significantly more time than a pixel value of 255 (white).

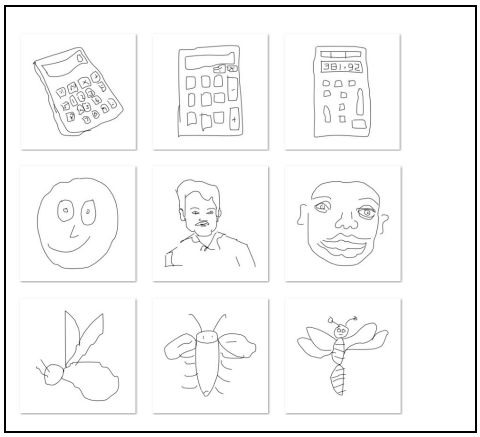

Fig.2. Variation of hand drawn calculators, faces and bees.

Dilation was applied to the picture to make lines thick, so they do not lose features during the resizing operation, where the pictures were resized to 100x100 pixels from the original 1111x1111 pixels. Fig.3 presents the sample image which passed through these steps.

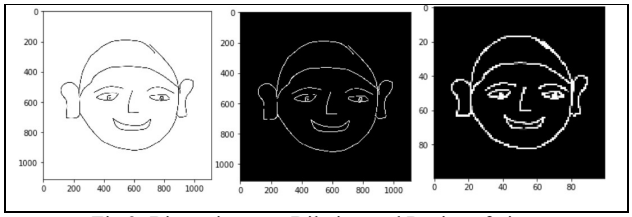

Fig.3. Binary inverse, Dilation and Resize of picture.

The last step of image processing and the 1st step of classification operation entailed flattening the final image from 100x100 matrix to [10000, 1] array to feed as input for classification algorithm.

### *B. Testing dataset with SVM*

As is evident in Section 5.A, we did not apply any feature extraction algorithm. The only thing we possess is the value of each pixel ranging from 0 to 255.Our first job was to search for the best parameters to train SVM. Several options were selected to train SVM with 2nd dataset. Table I contains the list of all parameters selected during grid search.

TABLE I.

| Parameter | Selected values |
|---|---|
| Penalty parameter C | 1, 10, 100, 1000 |
| Kernel | Linear, RBF |
| Kernel coefficient for 'rbf' | 0.001, 0.0001 |

The resultant set of parameters with highest the prediction accuracy was {C=10, Kernel=RBF, Gamma=0.001}. Next, we trained the 1st and 2nd datasets using SVM with these given parameters. As a rule of thumb, we reserved one-fourth of the dataset for testing and trained model with remaining %75 of the dataset.

The results of training model using SVM and pixel value of the pictures are presented in Table II. The 2nd dataset produced promising results (%59) compared to the 1st dataset (%29). As it was expected, the 1st dataset training resulted

with less prediction accuracy comparing to 2nd dataset, since it has more classes to predict (62 vs. 250).

TABLE II.

| Dataset | Precision | Recall | F1 score |
|---|---|---|---|
| 1st dataset | 0.29 | 0.27 | 0.26 |
| 2nd dataset | 0.61 | 0.59 | 0.59 |

### C. *Testing accuracy with ANN*

Secondly, we used our datasets as input to artificial neural network model. We tested several network structures to observe the difference between classification accuracy and level of closeness to the state-of-art prediction accuracy of Yu *et al* [5], which is %77.

Table III is the structure of artificial neural networks we used during our experiment. We tried both deep and shallow network structures during our study.

TABLE III.

| ANN-1(Shallow) | ANN-2(Deep) |
|---|---|
| Input layer – 10,000 | Input layer – 10,000 |
| Hidden layer 1- 5,000 | Hidden layer 1- 10,000 |
| Dropout (0.25) | Dropout (0.25) |
| Hidden layer 2 – 512 | Hidden layer 2 – 2048 |
| Output layer (50/250) | Dropout (0.25) |
| | Hidden layer 3 – 512 |
| | Output layer (50/250) |

All hidden layers are configured with Rectifier Linear Units (ReLUs), which have proven to be faster than their equivalent tanh or sigmoid units [10]. Dropout was performed after some layers to avoid co-dependences between different nodes [11]. Cross entropy was used as loss function [12], and weights were optimized using Adam optimizer [13]. Unlike other layers, the final layer used the Softmax function for final probability prediction [14].

Network trained with criteria to reach at least %95 of training accuracy and reduce loss value to less than 0.1. While the 2nd dataset required less epochs for this, the 1st dataset needed at least twice more epochs to reach criteria.

The accuracy of predictions on test data using trained models is presented in Table IV. Just like in Table II, we have better results for the 2nd dataset.

TABLE IV.

| Dataset | ANN-1 | ANN-2 |
|---|---|---|
| 1st dataset | 0.25 | 0.21 |
| 2nd dataset | 0.52 | 0.51 |

## VI. DISCUSSIONS

The prediction accuracy of Eitz *et al* was %56. We should note that they reduced the data size to lower computation cost and speed. We also reduced the dataset to one-fourth its original size, it was the 2nd dataset. However, we also trained the complete dataset, 1st dataset, using SVM. While the 2nd dataset results are better than the results of Eitz *et al*, the results of the 1st dataset are much lower, comparing to Eitz *et al*, at %29. We can conclude that minimal milestone accuracy should be pinned to %30, and all future experiments can reference their prediction accuracy to it. The result of the 2nd dataset is really high, which provides us very valuable information. Thus, when researchers work with a smaller size of sets, they should ensure that, first of all, the data is trained without any feature extraction, and continue further to see how well their new method is performing.

By comparing the results of Yu *et al* [5] with our results, we found that convolutional neural network showed very high performance by reaching a prediction accuracy of %77. If our 2nd dataset outperformed reference Eitz *et al* [1] results on SVM; this is not the same case for neural network. In neural network, both of the datasets fell far behind the reference Yu *et al* [5] results. Thus, we have no point to propose results of Table IV as minimal milestone for research comparison, as the results are not significantly important.

## VII. REFERENCES


[1] Eitz, Mathias, James Hays, and Marc Alexa. "How do humans sketch objects?." ACM Trans. Graph. 31.4 (2012): 44-1.

[2] Li, Yi, Yi-Zhe Song, and Shaogang Gong. "Sketch Recognition by Ensemble Matching of Structured Features." BMVC. Vol. 1. 2013.

[3] Li, Yi, et al. "Free-hand sketch recognition by multi-kernel feature learning." Computer Vision and Image Understanding 137 (2015): 1-11.

[4] Yu, Qian, et al. "Sketch-a-net that beats humans." arXiv preprint arXiv:1501.07873 (2015).

[5] Yu, Qian, et al. "Sketch-a-net: A deep neural network that beats humans." International journal of computer vision 122.3 (2017): 411-425.

[6] VanderPlas, Jake. Python data science handbook: essential tools for working with data. " O'Reilly Media, Inc.", 2016.

[7] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[8] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[9] Gulli, Antonio, and Sujit Pal. "Deep Learning with Keras". Packt Publishing Ltd, 2017.

[10] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[11] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.

[12] De Boer, Pieter-Tjerk, et al. "A tutorial on the cross-entropy method." Annals of operations research 134.1 (2005): 19-67.

[13] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." International Conference on Learning Representations 2015. ( arXiv preprint arXiv:1412.6980).

[14] Jang, Eric, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax." International Conference on Learning Representations 2017 . (arXiv:1611.01144 (2016)).