

Hiragana Handwriting Recognition Using Deep Neural Network Search

<https://doi.org/10.3991/ijim.v14i01.11593>

Rosalina ^(✉), Johanes P. Hutagalung, Genta Sahuri
President University, Jawa Barat, Indonesia
rosalina@president.ac.id

Abstract—These days there is a huge demand in “storing the information available in paper documents into a computer storage disk”. Digitizing manual filled forms lead to handwriting recognition, a process of translating handwriting into machine editable text. The main objective of this research is to create an Android application able to recognize and predict the output of handwritten characters by training a neural network model. This research will implement deep neural network in recognizing handwritten text recognition especially to recognize digits, Latin / Alphabet and Hiragana, capture an image or choose the image from gallery to scan the handwritten text from the image, use the live camera to detect the handwritten text real – time without capturing an image and could copy the results of the output from the off-line recognition and share it to other platforms such as notes, Email, and social media.

Keywords—Hiragana, Handwriting Recognition, Deep Neural Network Search, Android, Real-time

1 Introduction

There is a vast amount of history about handwriting because it has been around for a very long time, just like our writing today, it is used to store information and communicate with others. In the past humans used to only communicate using verbal or body language, then handwriting started off as pictographs drawn on rock and it progressed and developed into abstract symbols. Over the years handwriting keeps on evolving and depending on the region. Furthermore, different languages have very different characteristics of their alphabets which form the basis of this written text, and in some languages alphabets are written isolated from each other (e.g., Thai and Japanese) moreover, in some other cases they are cursive and sometimes the characters are connected with each other (e.g., Arabics). All of these challenges were recognized by many researchers [1 - 3].

In Japanese languages, for instance, the written script consists of three types: logographic kanji and syllabic hiragana and katakana. The scripts differ in both appearance and usage. For hiragana mostly used for grammatical morphemes while katakanas are used for transcribing foreign words. Examples of each scripts are shown in Fig. 1.

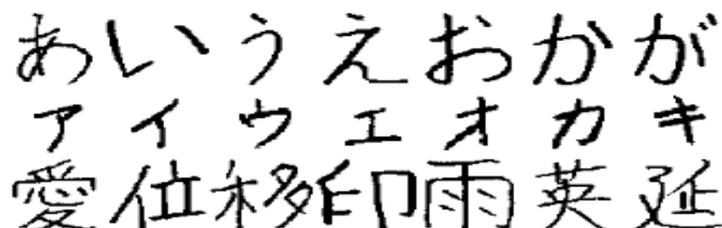


Fig. 1. Top row: hiragana scripts, second row: katakana scripts and third row: kanji scripts.

Japanese text does not have delimiters like spaces, separating different words. Also, several characters in the Japanese alphabet could be home-morphic, i.e. have similar shape definition which could add to the complexity of the recognition process. Thus, Japanese OCR is a very challenging task and many research efforts have been conducted to perform these task. A survey of some of the approaches to OCR for the Japanese language have been discussed in [4].

These days there is a huge demand in “storing the information available in paper documents into a computer storage disk” one of the example is in banking area where they need to processed many individual applications of their customers, the application fulfillment processes consists of manual information filling by the applicants in an application request form and then to be inputted by data processing staff which is actually redundant. To improve the performance, in addition to the solution, information filled in by the customers can be digitized, configured, validated and to be approved electronically in a better timing and a better accuracy.

Digitizing manual filled forms lead to handwriting recognition, a process of translating handwriting into machine editable text. The handwriting recognition itself in general classified into two types as offline and online handwriting recognition methods. Offline handwriting recognition involves automatic conversion of text into an image.

Handwriting Recognition is based on the technology known as OCR (Optical Character Recognition) with the difference being in recognizing handwritten characters and printed characters. OCR, is a technology that enables computers to convert different types of documents, such as scanned paper documents, PDF Files or images captured by a camera into editable and searchable text data. In Handwriting Recognition is much more difficult than OCR, because handwriting differ from one person to another. The handwriting style, shape of the alphabets and its sizes makes the difference and complexity to recognize the characters.

As described in [4], most of the recent Japanese character recognition approaches, both for handwritten and printed text, either use soft computing based approaches for classification, or, image shape/morphology characteristics for classification. The most popular of the soft computing approaches is the neural network [5] followed by Genetic Algorithms [6], Hidden Markov Model (HMM) [7] and Support Vector Machine [8].

This research will implement deep neural network in recognizing handwritten text recognition especially to recognize digits, Latin / Alphabet and Hiragana, capture an

image or choose the image from gallery to scan the handwritten text from the image, use the live camera to detect the handwritten text real – time without capturing an image and could copy the results of the output from the off-line recognition and share it to other platforms such as notes, Email, WA etc.

2 System Overview

This application implement deep neural networks in Python and trained using Keras and TensorFlow, deep neural network search is used in this application to train the model to recognize handwritten images. The deep neural network type used is CNN (Convolutional Neural Network) as the architecture of the trained model.

The handwriting recognition process happens in the android environment using JAVA. In On-line recognition the dataset is imported to the android application, then user draws the character on the bitmap and the function will compare the bitmap with all the data from the dataset and the ones with the highest percentage probability will be chosen and resulted as the predicted character. The methods in Off-line recognition for HWR Captured and Real-Time are quite similar, both scan the image or document and search for text, when the text are found it will create a text block and scan the characters inside the text block and compare it with the characters in the trained dataset for the correct output.

2.1 Data

The MNIST (Modified National Institute of Standards and Technology database) is a large database of handwritten digits. The MNIST database has a training set of 60000 examples and a test set of 10000 examples. The pixel values of images vary from 0 to 255 0 means background (white), 255 means foreground (black). While, for the handwritten Japanese characters were obtained from the Electrotechnical Laboratory (ETL) Character Database [9] as shown in Table 2.

Table 1. ETL dataset information

Dataset	Script	Writers	Classes	Examples
ETL-1	Katakana	1411	51	71,961
ETL-8	Hiragana	160	75	12000
ETL-8	Kanji	160	878	140,480

2.2 Model training

The first model is for the MNIST dataset and uses Keras to create the model. Below summarizes the neural network architecture:

- The first hidden layer is a convolutional layer called a Convolution2D. The layer has 32 feature maps, which with the size of 5×5 and a rectifier activation function.

This is the input layer, expecting images with the structure outline above [pixels][width][height].

- Next a pooling layer that takes the max called MaxPooling2D. It is configured with a pool size of 2×2.
- The next layer is a regularization layer using dropout called Dropout. It is configured to randomly exclude 20% of neurons in the layer in order to reduce overfitting(modeling error which occurs when a function is too closely fit to a limited set of data points).
- Next is a layer that converts the 2D matrix data to a vector called Flatten. It allows the output to be processed by standard fully connected layers.
- Next a fully connected layer with 128 neurons and rectifier activation function.
- Finally, the output layer has 10 neurons for the 10 classes and a softmax activation function to output probability-like predictions for each class.

The model is trained using logarithmic loss and the ADAM gradient descent algorithm. The model is fit over 10 epochs with updates every 200 images. Epochs are the number of training steps per n images the training progress will display. Epochs may take about 45 seconds to run on the GPU (e.g. on NVIDIA 950m GTX). The network achieves an error rate of 1.03.

2.3 On-line character recognition

On-line Character Recognition handles the input from the canvas where the user can draw the character desired and will predict the correct output. The most important piece for this feature is the Bitmap. After the Bitmap is created user must be able to draw on it.

The On-line Character Recognition Layout (Fig.2.b) can be accessed directly from the Main Screen (Fig.2. a) the user may choose between, digits, Latin/English and Hiragana dataset (Fig.2.c) for predicting, then the user can draw the character on the canvas and detect it. The device will predict it, provide the output of predicted character and the probability percentage of the character similarity with the dataset (Fig.2.d).

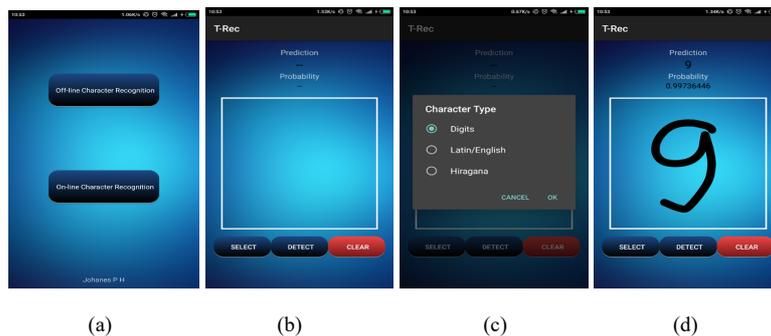


Fig. 2. T-Rec Main Screen (a), Online Recognition layout (b), Selecting Character Type (c), recognition result.

2.4 Off-line character recognition

The purpose of Off-line Character Recognition Layout (Fig.3.) is to direct the user to the choice of 2 features of Off-line Character Recognition which are HWR (Captured) and Real-time Recognition.



Fig. 3. Offline Character Recognition Layout

The HWR (Captured) Layout (Fig.4.a) can be accessed directly from the Off-line Character Recognition Layout (Fig.3.). The user may select the source of image (Fig.4.b) either capture from camera or select image from the gallery then the user is able to crop the image to fit the text desired to be scanned, and the result will appear on the bottom of the image (Fig.4.c.).

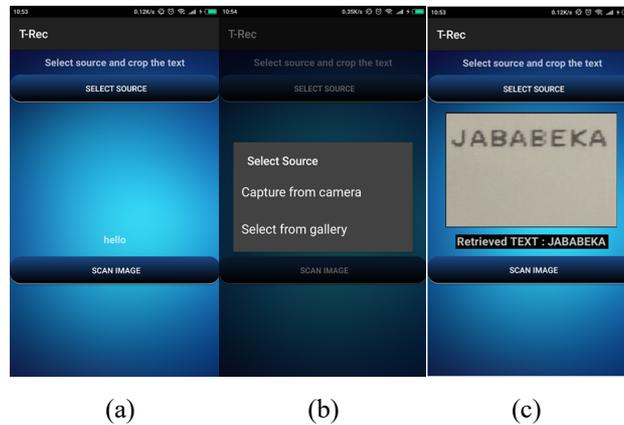


Fig. 4. The HWR (Captured) Layout (a), HWR (Captured) Select (b), HWR Result (c)

2.5 Real-time recognition

The Real-Time Recognition feature can detect handwritten text from live camera feed, it will show the text that is picked up using bounding boxes and then output that text.

Real-Time Recognition Layout (Fig.3.) can be accessed directly from the Off-line Character Recognition Layout (Fig.5.a.). The application will call the camera app (Fig.5.b) and user can tap on the desired text to paste it into the canvas, user may also turn on flash. The Result will display on the canvas and user can copy to clipboard and share it to other platforms (Fig.5.c)

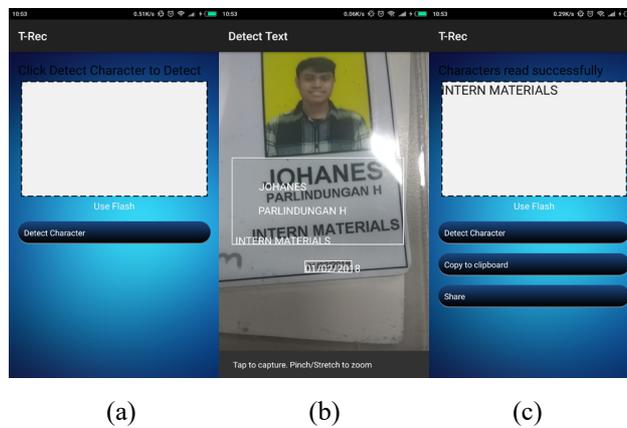


Fig. 5. The Real-Time Recognition Layout (a), The Real-Time Detect Text (b), The Real-Time Result (c)

3 Accuracy Results

The training data consists of handwritten samples of 3410 alphabet characters (55 each character), 60000 example digits (varied quantity) and 10944 Japanese Hiragana (152 per Hiragana character). The accuracy results of each dataset obtained after training the model using Keras and TensorFlow with CNN (model 1) Single layer network (model 2), can be seen in (Table 2), Result of accuracy of testing the application based on On-line, Captured and Real-Time recognition can be seen in (Table 3).

Table 2. Accuracy of each dataset training results depending different network

dataset	Model1(CNN)	Model2(Single-Layer)
Digits	99.56%	93.21%
Latin/English	94.22%	91.03%
Hiragana	90.78%	89.90%

The training accuracy can be increased also by the number of data used or the number of epochs increased with the disadvantage of increasing training time also

needing a stronger processor and graphics card. The following is the result of training time based on increasing the number of epochs using a basic CNN model and changing the dataset trained on.

Table 3. Training Time based on number of epochs

	10 epochs	20 epochs	50 epochs
Digits	5 Minutes	11 Minutes	1 Hour 5 Minutes
Latin/English	30 Minutes	1 Hour	3 Hours
Hiragana	45 Minutes	2 Hours	+4 Hour

Increasing the epoch doesn't always guarantee a higher accuracy, user must test out the models until the most optimal one with the right time and accuracy is found would. In conclusion add more data to the dataset and keep the number of epochs at an optimal level, because it will be redundant and waste of processing power.

4 Conclusion

This application has a lot of potential with more datasets of training and a more optimal model after testing with different types of models, T-Rec can be a reliable HWR application and will be useful to store/read handwriting from documents in the future.

5 References

- [1] Ciresan, Dan Claudiu, Meier, Ueli, Gambardella, Luca Maria, and Schmidhuber, Jürgen. (2010). Deep big simple neural nets excel on handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010. https://doi.org/10.1162/neco_a_00052
- [2] Rosalina, R.B. Wahyu. (2017). "Offline Handwriting Recognition Using Feed forward Neural Network", *International Journal of Information Technology and Computer Science (IJITCS)*, Vol.9, No.9, pp.11-17. <https://doi.org/10.5815/ijitcs.2017.09.02>
- [3] Song, Wang, Uchida, Seiichi, and Liwicki, Marcus. (2011). Comparative study of part-based handwritten character recognition methods. In 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 814–818. <https://doi.org/10.1109/icdar.2011.167>
- [4] Das S and Banerjee S (2014), Survey of Pattern Recognition Approaches in Japanese Character Recognition, *International Journal of Computer Science and Information Technology*, Vol. 5(1) 93-99.
- [5] Neural Networks which consist of Simple Recurrent Networks for Character Recognition by Template Matching, (2008) Y. Shimozawa, *Journal of the Information Processing Society* 49(10), 3703-3714.
- [6] The post processing of Character Recognition by Genetic Algorithms, Y Shimozawa, S. Okoma, (1999), *Journal of the Information Processing Society*, 40(3) 1106-1116.
- [7] Off-line Hand Printed Character Recognition System using directional HMM based on features of connected pixels, H. Nishimura, M. Tsutsumi, M. Maruyama, H. Miyao and Y. Nakano (2002), *Journal of the Information Processing Society* 43(12) 4051-4058.

- [8] Maruyama, K.-I.; Maruyama, M.; Miyao, H.; Nakano, Y.; "Hand printed Hiragana recognition using support vector machines"; *Frontiers in Handwriting Recognition*, (2002). Proceedings. Eighth International Workshop on Digital Object Identifier: 10.1109/IWFHR.2002.1030884; 2002, Pages: 55 –60. <https://doi.org/10.1109/iwfh.2002.1030884>
- [9] Electrotechnical laboratory character database. <http://etlcbd.db.aist.go.jp>, 2016

6 Authors

Rosalina is working as a lecturer in the Faculty of Computing at President University. She received her bachelor and master degree in Information Technology from President University in 2005 and 2015 respectively.

Johanes P. Hutagalung received his Bachelor degree in Information technology at President University and currently he is doing his masters degree programme in Institute Technology Bandung.

Genta Sahuri is currently working as a lecturer in the Faculty of Computing at President University, Indonesia.

Article submitted 2019-08-30. Resubmitted 2019-09-23. Final acceptance 2019-09-29. Final version published as submitted by the authors.