

Development of an on-Premise Indonesian Handwriting Recognition Backend System Using Open Source Deep Learning Solution for Mobile User

Gianino Masasi, James Purnama, Maulahikmah Galinium

Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia

Article Information

Received: 29 January 2021
Accepted: 17 March 2021
Published: 18 March 2021
DOI: 10.33555/jaict.v7i2.109

Corresponding Author:

Name: Gianino Masasi
Email: gianino.masasi@student.sgu.ac.id

ISSN 2355-1771
eISSN 2723-4827

ABSTRACT

Existing handwriting recognition solution on mobile app provides off premise service which means the handwriting is processed in overseas servers. Data sent to abroad servers are not under our control and could be possibly mishandled or misused. As recognizing handwriting is a complex problem, deep learning is needed. This research has the objective of developing an on-premise Indonesian handwriting recognition using open source deep learning solution. Comparison of various deep learning solution to be used in the development are done. The deep learning solution will be used to build architectures. Various database format are also compared to decide which format is suitable to gather Indonesian handwriting database. The gathered Indonesian handwriting database and built architectures are used for experiments which consists of number of Convolutional Neural Network (CNN) layers, rotation and noise data augmentation, and Gated Recurrent Unit (GRU) vs Long Short-Term Memory (LSTM). Experiment results shows that rotation data augmentation is the parameter to be change to improve word accuracy and Character Error Rate (CER). The improvement is 64.8% and 23.2% to 69.6% and 20.6% respectively.

Keywords: CRNN, Deep Learning, Indonesian Handwriting Recognition, On premise, Tensorflow

1. Introduction

A salesperson who usually roams around finding clients usually gives out forms after a client purchases the goods they are selling. Later on, they will store the values in the forms (names, addresses, etc.) into computers in their office by means of scanning the document or manually typing each value. This way of storing data is time-consuming and prone to human error (mistyped values).

The solution to the previously mentioned problem is to use a handwriting recognition software. Existing handwriting recognition softwares are categorized into two types, online and offline (Plamondon & Srihari, 2000). Both of these types of software run on a desktop/laptop with an OS which means if a salesperson wanted to store the

values in the forms while out on the field then they need to bring a laptop or desktop everywhere they go, not to mention they need to boot up the laptop/desktop every time they need to scan a form. Valuable time will be wasted because of this low mobility.

By developing the Handwriting Recognition system for mobile user, then the user of said system will have high mobility. It is worth mentioning that the mobile app version of handwriting recognition system exists. However, the services they provide are off premise cloud based, meaning the handwriting we feed into the app is processed in the respective company's server which brings us to a new problem especially if the data is personal data (Name, Address, etc.)

Off premise servers are located faraway physically and we have no idea what the company will do with the data sent to their servers as we have no control towards our data being sent to these off-premise servers. One example of this are Facebook giving authorization to Cambridge Analytica to access its users' personal data which amounts to more than 87 million unknowing users' personal data (Isaak & Hanna, 2018). The solution is to use on premise server which means the data is under our control and furthermore there is a law prohibiting such case like Facebook.

According to the Rancangan Undang-Undang tentang Perlindungan Data Pribadi pasal 32 (Draft of Protection of Personal Data Law article 32), which is roughly translated as Companies are not allowed to transfer users' personal data abroad except the destination country has the same protection level as this regulation, except:

1. There has been a contract between the company and the data receiver located outside of Indonesia.
2. International Agreement between countries.

The transferring of user's personal data by companies is prohibited with some exception. To solve this, an on-premise system implementation which ensures that users' data are stored only in the respective company's server is needed. The on-premise requirement can be fulfilled by using an open-source library such as TensorFlow, Keras, PyTorch, etc.

Handwriting Recognition is a complex problem in which difficulties arise when reading different styles of handwriting since everybody have their own personal style of writing. Humans can recognize handwriting with near 100 percent when presented with fairly simple and clear pictures (no noise) of handwriting. However, when presented with increasing distortion, noise, and slants humans' performances are outperformed by computers by 5 percent (Chellapilla et al., 2005).

2. Related works

From literature review, comparisons between existing deep learning libraries had been made [4]. The libraries being compared consists of Theano, Torch, Caffe, Tensorflow, and DeepLearning4J. All libraries were tested using the MNIST database (Deng, 2012) with the task of classifying the digits present in the database. The architecture used was a Fully Connected Neural Network. Several categories are used for comparison purposes. The categories are Training Time, Prediction Time, Accuracy, and Lines of Code. The result shows that Tensorflow and Theano was two of them that performed better than the rest. Tensorflow performed best in prediction time and lines of code while Theano performed best in training time, accuracy, and lines of code. Tensorflow was eventually chosen since further research shows that Theano has been discontinued.

Papers show that IAM Database is generally used to train and test the architecture for detecting handwriting (Chowdhury & Vig, 2018) (Shi, Bai, & Yao, 2015). IAM Database format is word in an image with close margins between the word and the image border. The type of the images are also PNG.

The architecture used in the papers for detecting and translating handwritten words is CRNN (Chowdhury & Vig, 2018) (Shi, Bai, & Yao, 2015). CRNN comprises of CNN layers combined with RNN layers right after it. The architecture in the papers discussed the usage of 7 layers of CNN and 2 layers of bidirectional LSTM for the RNN. CNN layers helped with picking up features of an image that are used for classification purposes while RNN helps with the pattern recognition which in this case is the sequence of alphabets in a word.

Papers also shows that metrics that is used in evaluating and measuring the performance of the architecture consists of two metrics, Word Error Rate (WER) and Character Error Rate (CER) (Chowdhury & Vig, 2018) (Shi, Bai, & Yao, 2015). Word Error Rate measures the number of wrong words classified in a line of words or sentence while Character Error Rate measures the number of wrong characters classified in a word.

Development of a python backend are done by using Flask (Aslam, 2015). Flask is a micro framework for python that is useful in creating web applications.

3. Research methods

The research goes through several steps which is shown in figure 1. It is started by finding the suitable deep learning on premise solution. The solution is used to build the handwriting recognition. After the solution is found, the next step is finding a suitable handwriting database format. The format is used to gather handwriting. The collected handwriting then goes through pre-processing with the suitable handwriting database format as a guide. After pre-processing, the database is split into testing set and training set. An initial deep learning architecture are built to act as a baseline for later experiments. Literature Review is the reference to build the architecture. The aforementioned database is used to train and test the architecture in that particular order.

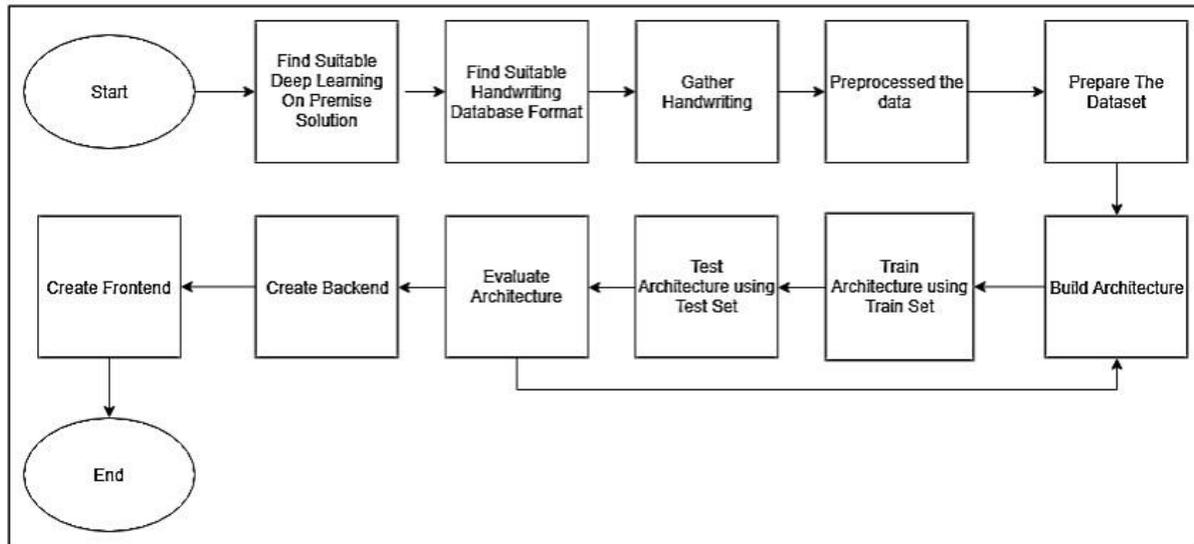


Figure 1. Research Overview

The results of testing the database are used for evaluation. After finding the optimal architecture, a backend is constructed with a frontend built later on to proof the backend system works.

3.1 Finding on premise deep learning solution

Table 1 is the summary of all the charts present in the paper in (Kovalev, Kalinovsky, & Kovalev, 2016). For training time, prediction time, and accuracy, the average is used.

Table 1. Comparison of deep learning libraries

	Training Time	Prediction Time	Accuracy	Lines of Code
Keras with Theano Backend	75	0.4	97.67	62
Torch	105	0.3	93	165
Caffe	75	0.48	95.5	110
Tensorflow	85	0.27	90.67	80
DeepLearning4J	1070	3.1	96.33	130

The first and second ranking of each category is in bold. Theano performed best in Training Time, Accuracy, and Lines of Code. Tensorflow followed in Prediction Time and Lines of Code. However, further research found that Theano has been discontinued.

3.2 Gather Handwriting

In this step, handwritings are collected with an A4 paper. 54 words are written which were “Banten”, “Jakarta”, “Bandar”, “Lampung”, “Bandung”, “Surabaya”, “Medan”, “Makassar”, “Semarang”, “Palembang”, “Batam”, “Padang”, “Jalan”, “JL”, “Satu”, “Dua”, “Tiga”, “Empat”, “Lima”, “Enam”, “Tujuh”, “Delapan”, “Sembilan”, “Sepuluh”, “Barat”, “Selatan”, “Timur”, “Utara” with each of them in uppercase and lowercase letters.

The collected handwriting database amount to 2617 pictures from 50 volunteers. Each picture is stored in PNG format and scanned using HP Deskjet 2050 J510 Series printer with 300 dpi resolution. Each scanned paper is put into an algorithm which crops the words in the papers into a format which is similar to the IAM Database data which are margins close to the image border and PNG type. Figure 2 is the example of the handwriting sample.

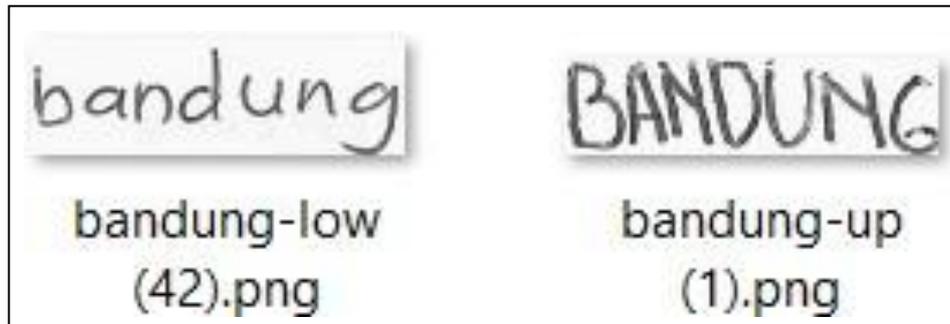


Figure 2. Cropped pictures examples

3.3 Evaluate Architecture

The initial architecture consists of 7 layers of CNN and 2 bidirectional LSTM. The result from using the said architecture will be used as a baseline for improvement. The evaluation is carried out by doing 3 experiments which are number of CNN layers, Data Augmentation, and GRU vs LSTM.

Number of CNN layers are carried out by adding or subtracting the CNN layers. 5 and 6 layers are used to find out whether lower number of layers yield better result while 8 and 9 layers are used for higher number of layers.

As illustrated in figure 3, data augmentation is carried out by adding rotation, noise, and combination of rotation and noise. Rotation value are randomized between 1 and 4. Noise value are randomized between 0.05 and 0.8 rounded to 2 numbers behind zero.

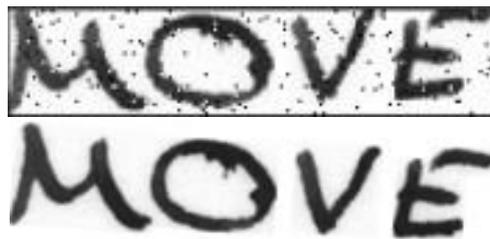


Figure 3. Top: added noise. Bottom: added rotation

The experiments are evaluated using Word Accuracy and Character Error Rate (CER).

$$\text{Word Accuracy} = \frac{\text{Number of words spelled correctly}}{\text{Number of words}} \quad (1)$$

$$\text{CER} = \frac{\text{Number of characters in a word wrongly classified}}{\text{Total characters in a word}} \quad (2)$$

4. Results and discussion

Table 2 shows the result of all the experiments. Both Gathered Handwriting and IAM Database are used as inputs to the experiments.

Table 2. Result of all experiments

Type of Experiments	Gathered Handwriting		IAM Database	
	Word Accuracy	Character Error Rate	Word Accuracy	Character Error Rate
CNN 5 Layers	59.7%	24.3%	59%	21.8%
CNN 6 Layers	62.5%	27%	59.6%	21%
Initial Experiment	64.8%	23.2%	60.9%	20.5%
CNN 8 Layers	66.4%	24.3%	62%	20.3%
CNN 9 Layers	61.7%	26%	63.1%	20.1%
Rotation	69.6%	20.6%	61%	20.3%
Added Noise	61.3%	20.4%	61.3%	20.4%
Rotation + Added Noise	67.8%	21.4%	63%	20.2%
Bidirectional GRU	61.7%	22.9%	61.2%	21.4%

Applying rotation augmentation when using Gathered Handwriting improves the initial experiments the most when compared with CNN layers and changing LSTM to GRU. This is due to more variation of an image learnt by the system. However, when using IAM Database it shows that more layers of CNN in this case 9 layers improves the initial experiment the most when compared with the Data Augmentation and changing LSTM to GRU.

Observation shows that using 8 layers of CNN and adding rotation augmentation improve the Word Accuracy and Character Error Rate (CER) of the initial experiment. Therefore, the combination of them would be done for research purpose.

Table 3. Result of combining 8 layers of CNN and rotation augmentation

	Word Accuracy	Character Error Rate
Initial Experiment	64.8%	23.2%
8 Layers	66.4%	24.3%
Rotation	69.6%	20.6%
8 Layers + Rotate Augmentation	70.1%	21.9%

Table 3 shows result of using 8 layers with the input image being rotated. By combining the result of the number of CNN layers experiment and data augmentation experiment, the initial experiment's results are improved by 5.3% and 1.3% for Word Accuracy and CER respectively.

5. Conclusion

In conclusion, the development of an on premise Indonesian Handwriting Recognition Backend System using Open Source Deep Learning Solution for Mobile User are built by using Tensorflow to provide the on premise capabilities, IAM Database format to gather Indonesian Handwriting Data, and adding rotation augmentation helps achieve improved result from the result of the initial experiment. However, in terms of parameter to be changed it is different in terms of data set. Adding data augmentation in particular rotation will help improve your Word Accuracy and CER if the database used amounts to 2617 data, 54 variety of words, and 50 writers while if the database is 115000 in amount, more than 54 variety of words, and 657 writers then more layers of CNN help achieve improvement. It is also important to note that combination of rotation augmentation and 8 layers CNN, since both shows best result in their respective experiments, improves the result of the 8-layer experiment results with 0.5% for Word Accuracy.

References

- Aslam, e. (2015). Efficient way of web development using Python and Flask. *International Journal of Advanced Research in Computer Science*, 6(2), pp.54-57.
- Chellapilla, K., et al. (2005). *Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs)*. Retrieved from 5th Conference on Email and Anti-Spam: <https://ceas.cc/papers-2005/160.pdf>
- Chowdhury, A., & Vig, L. (2018). An efficient end-to-end neural model for handwritten text recognition. *The proceedings of BMVC*, pp.1-11. Retrieved from <http://bmvc2018.org/contents/papers/0606.pdf>
- Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), pp.141-142.
- Isaak, J., & Hanna, M. J. (2018). User data privacy: Facebook. *Cambridge Analytica, and privacy protection. Institute of Electrical and Electronics Engineers*, 51, pp.56-59.
- Kovalev, V., Kalinovsky, A., & Kovalev, S. (2016). *Deep Learning with Theano, Torch, Caffe, Tensorflow, and Deeplearning4J: Which One is the Best in Speed and Accuracy?* Retrieved from http://elib.bsu.by/bitstream/123456789/158561/1/Kovalev_Kalinovsky_Kovalev.pdf
- Plamondon, R., & Srihari, S. N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(1), 63-84.
- Shi, B., Bai, X., & Yao, C. (2015, July). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p.99.