Mobile Client- Server Approach for Handwriting Digit Recognition

Hasbi Ash Shiddieqy
School of Electrical Engineering and Informatics, Institut Teknologi Bandung
University Center of Excellence on Microelectronic, Institut Teknologi Bandung
Bandung, Indonesia
hasbiida@gmail.com

Trio Adiono
School of Electrical Engineering and Informatics, Institut Teknologi Bandung
University Center of Excellence on Microelectronic, Institut Teknologi Bandung
Bandung, Indonesia
tadiono@stei.itb.ac.id

Infall Syafalni
School of Electrical Engineering and Informatics, Institut Teknologi Bandung
University Center of Excellence on Microelectronic, Institut Teknologi Bandung
Bandung, Indonesia
infall@itb.ac.id

***Abstract*—In the era of an Internet of Things, pattern recognition technology is growing rapidly, especially by the massive implementations of artificial intelligence (AI). Selecting the right implementation for an AI algorithm for an application could be quite challenging and time-consuming. In this paper, we propose a client-server system implementation for handwriting digit recognition. A client-server is set based on TensorFlow with multiple models for classifications. The client is set based on a mobile application that the user inputs the digit by touch panel of the mobile. In this paper, the models at the server are trained and tested by using MNIST database of handwritten. In addition, we use convolutional neural network (CNN) to improve the performance of the neural network. The client-server allows many users to be accessed by AI model from the same time. The advantage of using client-server approach is reducing power and time for processing handwriting recognition in client device. It also speeds up the time of development and implementation of algorithm on the server.**

Keywords—Client-server system, mobile application, artificial intelligence, digit recognition

With the advanced of the pattern recognition, more method and algorithm has been proposed. Most researchers use the same MNIST dataset to test the applied method[1]. However, the challenging problem is how to implement customized, low cost, high-speed pattern recognition system. Digit based recognition is regularly used as a standard basis for pattern recognition.

The breakdown of the implementation of digit recognition handwriting into two types as offline and online recognition method. The digit recognition consists of some modules for the processing, starting from the extraction, segmentation, and recognition. As for the recognition algorithm itself, various algorithms are used from on-hand designed heuristics, classical machine learning such as KNN, SVM, PCA, RBF to those based on neural networks NN, and CNN[2], [3].

In this paper, the final application implementation is taken out on the mobile application and the server with recognition services. The main problem how to implement pattern detection with state-of-the-art client-server system. Handwriting digit recognition is simply for case studies, to show the client-server system method in processing user input to produce recognition results.

In summary, the main contribution of this work is a client- server system implementation allowing development becomes faster and improved CNN training performance can be archived. Existing CNN model can be retrain using user specific data in server side.

I. LITERATURE STUDY

A. *Client-server system*

The client-server system included a user interface in mobile client and executing the function from input in the server computer system. Most of the application will run in server and small quantities data will be passed between mobile user to the server.

Because the most process that requires a large resource is running on the server so that the mobile client barely requires high specifications device for running the application. Overall performance can be represented with the server performance. [4]

The advantage of a client-server-based system is proven, highly scalable, easy to integrate, low cost. Example of client-server system-based services is AI Services (AWS), Cloud AI (Google Cloud), AI Platform (Microsoft Azure), AICS (IBM). AI service makes it effortless to combine machine learning / artificial intelligence features in the application. All the AI service is available throughout the API.[4]

Disadvantages of the client-server based system eliminate the ability to run offline, concern in data privacy and protection, the cost for service managing servers [5].

B. *Pattern Recognition of handwritten digits*

There are many methods have been developed to implement CNN for different AI applications. Among all, the LeNet5 is one of the basic and significant CNN architecture that designed for handwritten recognition. Gradient-Based Learning Applied to Document Recognition

Recent pattern recognition using deep learning with higher complexity compared with classical machine learning which additional preprocessing to decrease pattern complexity [1].

C. *Implementation State of the Art*

Implementation state of the art using a client-server system allowing development becomes faster because experiment and training are done in server-side. Moreover, flexibility and scalable in server enable changing the pattern recognition module more efficiently. Implementation

978-1-7281-2156-7/19/$31.00 ©2019 IEEE

recognition module can be applied in software CPU, GPU, or fully customized hardware FPGA[6], [7].

D. Mobile Application

Most mobile application developer using native tools from Apple or Google to build application with best user experience. The downside is that tools are hard to learn and time-consuming. There is another way using some sort of cross-platform solution.

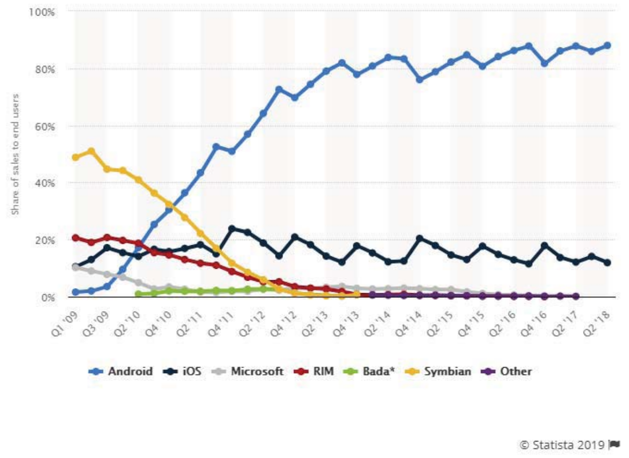

Fig. 1. Global mobile OS market share 2009-2018

With global share mobile from android 88%, ios 11.9% in Fig.1. Developer must cover both Android and iOS platforms. React native aiming to work on both platforms and required faster development if the developer already get used to *javascript* code.

II. Design And Implementation

In this paper, the image like a representation of handwriting becomes an input pattern recognition application. Furthermore, pattern recognition applied is based on a neural network with backpropagation training. Development is divided into two parts. First is training model in server side and the second is user interface with android client.

A. Design

The digit recognition consists of some modules for the processing, starting from the extraction, segmentation, and recognition. Many methods have been tested with training set and data set. In this paper, segmentation is not performed because recognition its only recognize single digit. Therefore, the module applied is simplified into extraction (preprocessing) and recognition (classifier).

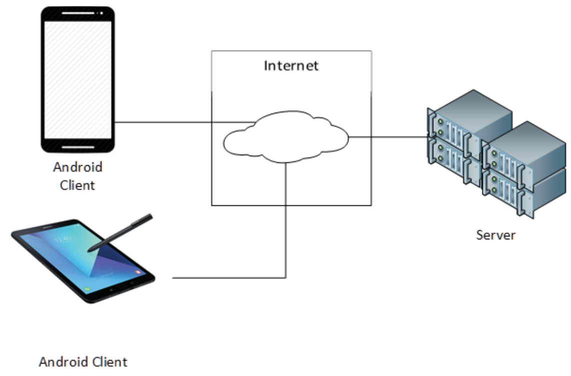

Fig. 2. Design system

Overall system consists of android client and server in Fig 1. Android client uses as sketch input from user, input from user then processed into images. The image represents of handwriting digit from user. After that the images sent to server via TCP connection. The server processes the image and replies to the user result of handwriting recognition.

B. Training handwritten digit recognition

Rather than development an algorithm, we proposed use of an existing algorithm for image recognition algorithm. This algorithm based on backpropagation learning. Artificial Neural network and convolutional neural network algorithm will be used for recognition module. The algorithm generates a model for training continued with optimizations to increase its performance.

C. Implementation Server digit recognition

Server service running in windows pc with network interface. This service develops using python programming with socket server service listen in port 9999. This service waiting for image request from client encode in base64. The moment server receives the data, then data will decode to transform into an image. After that, model loaded then applied to predict this image. Finally, server respond best result classified.

Server service utilized socket programming for receive data. *Tensorflow* as platform to predict digit recognition with scipy for preprocessing image and numpy array for fast array processing. For debugging matplotlib library used, but for deployment this option is not selected by default.

D. Development of Mobile Android Client

The proposed mobile application work as a client which get data sketch from user data and send data to server as image. We implemented android application using react native.

III. Experimental Result

Trained model implemented in server. System tested using real handwriting which takes input from android touch sensor as client. For the experiment server collected the user data handwriting as an image.

A. Digit recognition engine

The recognition engine consists of 2 architecture ANN and CNN. The proposed CNN model based on Lecun2

978-1-7281-2156-7/19/$31.00 ©2019 IEEE

architecture with decreased parameter for faster respond without losing accuracy.

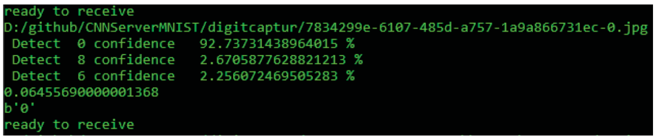

Fig. 3. Result of server service

In fig 3 Result of each received image displayed best 3 classifiy with best confidence.

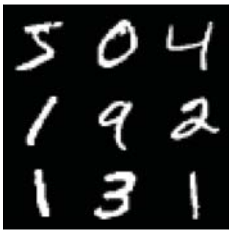

Fig. 4. Handwritten Digit from MNIST database

In fig 4 show the dataset handwritten digit which used to training our model.

B. Android mobile client

Fig. 5 shows the user interface design with three main parts; Canvas, Output, dan Send Image. The canvas is used to write the intended digit. Also, it also able to erase the canvas. The output will be shown in the bottom of the screen. And the Send Image is used to send the data from the user side to the server.

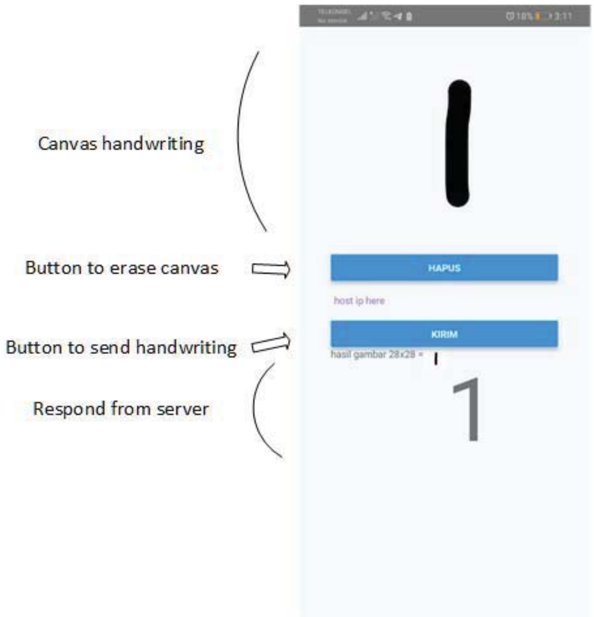

Fig. 5. User interface design

User interface design shown in fig 5. Make use of react native state to change between operation. Writing sketch, erase, drawn image, send image as base64 encode to server.

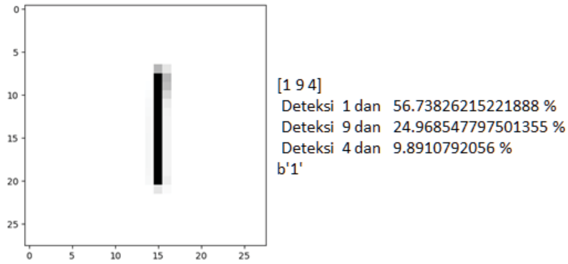

Fig. 6. Result server

In fig 6 shown result with confidence number from server, only b'1' or final detection is respond.

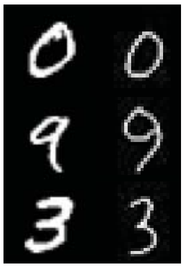

Fig. 7. Handwritten Digit MNIST (left) compared to real user input (right)

In fig 7 shown comparizon between dataset training with real user input. In the right side is the real user input, because android doesn't support thickness for its touch sensor the image has same default thickness.

C. Performance test

TABLE I. PERFORMANCE TEST

Architecture	Performance			
	Accuracy		*Total Parameter*	*Average Response time (ms)*
	Data Train	*Data Validation*		
ANN	93.75%	93.07%	53,568	47.643214
CNN	99.8%	99.2%	1,199,882	48.956417

In the performance test, we evaluate the accuracy and parameter for ANN and CNN for the proposed method. Table 1 presents performance of the architecture. We tested CNN model with real user data from android and resulted 56% accuracy.

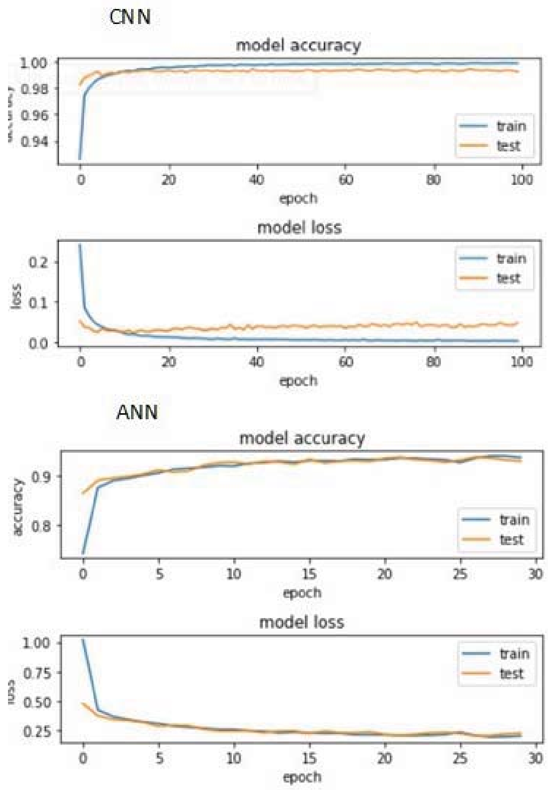

Fig. 8. Performance comparisons between CNN and ANN

The comparisons between CNN and ANN in epoch is shown in Fig. 8. As shown, the accuracy of CNN is nearly 1 while ANN is around 0.93.

IV. Conclusion and Future Work

The system operates and capable to classify handwritten digits with an accuracy of 99% in the test data MNIST and 56% in real user data from android.

Our future work add feature to retrains the model or network with small number user specific to allow an existing network to be customized without losing generality. Customized server with GPU or FPGA accelerated engine.

Acknowledgment

This work is supported by P3MI ITB.

References

[1] Y. Lecun, C. Cortes, and C. J.C. Burges, "THE MNIST DATABASE of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed: 01-Sep-2019].

[2] M. Y. W. Teow, "A minimal convolutional neural network for handwritten digit recognition," *in 2017 7th IEEE International Conference on System Engineering and Technology, ICSET 2017 - Proceedings*, pp. 171–176, October 2017.

[3] L. Y, B. L, B. Y, and H. P, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Lombardi, *WebSocket Lightweight Client-Server Comuunications*, 1st Editio. O'Reilly Media, 2015.

[5] M. Goodyear et al., *Enterprise System Architectures*, 2010th ed. CRC Press, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. deeplearningbook.org, 2016.

[7] Z. Chen, "Handwritten digits recognition," *in Proceedings of the 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2009*, vol. 2, pp. 690–694, November 2009.

[8] H. A. Shiddieqy, F. I. Hariadi, and T. Adiono, "Implementation of Deep-Learning based Image Classification on Single Board Computer," *ISESD 2017*, pp. 133–137, 2017.

[9] H. A. Shiddieqy, S. Ramadhan, and T. Adiono, "Implementations Fault Detection Phasor Measurement Unit using Zybo SoC," *ISESD 2017*, pp. 1–5, 2017.