

# Kyouiku Kanji Grade 1 Recognition Using MobileNet V2 Based on Android

Fathir Fathan <sup>a,1,\*</sup>, Alvi Syahrini Utami <sup>b,2</sup>, Danny Matthew Saputra <sup>b,3</sup>

<sup>a</sup> Undergraduated Informatics Student, Sriwijaya University, Indonesia

<sup>b</sup> Faculty of Computer Science, Sriwijaya University, Indonesia

<sup>1</sup> 09021282025088@student.unsri.ac.id\*; <sup>2</sup> alvisyahrini@ilkom.unsri.ac.id; <sup>3</sup> danny@ilkom.unsri.ac.id

\* corresponding author

## ARTICLE INFO

### Article history

Received

Revised

Accepted

### Keywords

MobileNet V2

ETL-9B

Handwritten Kanji Recognition

Kyouiku Kanji

## ABSTRACT

Character recognition has become a popular research topic in the field of pattern recognition and machine learning, including handwriting recognition, specifically *kanji* handwriting. This study performs handwriting recognition of *kyouiku kanji* grade 1, which is the *kanji* required to be learnt by grade 1 elementary school students in Japan. This research uses ETL-9B dataset from Electrotechnical Laboratory (now AIST), uses CNN MobileNet V2 deep learning method that has been customized for mobile devices, and uses Android application as the user interface implementation. Based on the study results, the highest accuracy model was obtained with an accuracy of 96,6875% and a size of 27.4MB for the alpha 1.0 hyperparameter. It can be concluded that the CNN MobileNet V2 deep learning method has performed quite well in the process of recognizing handwritten *kyouiku kanji* grade 1.

## 1. Introduction

Character recognition has become a popular research topic in the fields of pattern recognition and machine learning. With the development of digital technology, character recognition has become an important tool in many fields, including handwritten character recognition [1].

Character recognition involves complex phases such as preprocessing, segmentation, normalization, feature extraction, classification, and postprocessing. The task of character recognition also has its own difficulties, such as background complexity, uneven lighting, rotation, blur, degradation, aspect ratio, font, language, and others [1].

Recent advances in machine learning, especially deep learning, have shown promising results in the field of handwritten character recognition. Convolutional neural networks (CNN) are very effective in understanding the structure of handwritten characters by aiding in the automatic extraction of different features, making CNN a suitable approach for solving handwriting recognition problems [2].

Handwritten kanji recognition technology is widely used in pen-input interfaces, such as PDAs or mobile phone devices, and is expected to grow in popularity as its application scope expands in the future. However, its accuracy is still far from human capabilities [3].

Previous research on handwritten *kanji* usually focuses on the Kuzushiji-Kanji (ancient cursive kanji) dataset from Research Organization of Information and Systems - Center for Open Data in the Humanities (ROIS-CODH) due to the easy accessibility and format of the dataset, as well as its usefulness for recognizing classical documents. Regarding this dataset, there was a recognition study of 63 *kanji* for reading the Kiritsubo chapter of Genji Monogatari (a classical Japanese document) [4]. There is also research on recognizing 150 *kanji* that have the most samples in the dataset with the CNN method [5].

For the purposes of modern handwritten *kanji* recognition, the Electrotechnical Laboratory (ETL) dataset is used in this research. This dataset has a relatively large number of samples, but it is relatively more difficult to use because of its binary format. Regarding this dataset, there are several



recognition studies for 878 kanji using CNN method, including [6] and [7]. However, it is difficult to find research specific to *kyouiku kanji* grade 1, 80 *kanji* that all 1st grade elementary school children in Japan must learn, which is also a subset of *jouyou kanji* (2136 *kanji* designated by the Japanese government for daily use).

In this research, grade 1 *kyouiku kanji* recognition is carried out with an Android application as its user interface (UI) implementation, and using MobileNet V2 as CNN implementation that has been customized for mobile device usage [8]. MobileNetV2 is used in this research because, out of all the models available on Keras, this model has the smallest size and number of parameters, but still has a high accuracy value making it suitable for use on Android [9].

## 2. Literature Study / Hypotheses Development

### a. *Kanji*

*Kanji* are Japanese morphosyllabic characters derived from Chinese characters and are used in Japanese writing [10]. Japanese scholars attach Japanese meanings to Chinese characters, so that each Chinese character is considered to have, in addition to the Chinese sound, an additional Japanese sound that corresponds to its meaning in Japanese. These characters are now known as Japanese characters [11].

The Japanese government itself has published character lists periodically to help guide the education of its citizens through the myriad of characters that exist. Among them are 863 *kanji* that can be used in Japanese names [12], as well as 2136 *kanji* that can be used in general communication media [13].

### b. *Jouyou Kanji*

*Jouyou kanji* are *kanji* used to write modern Japanese in general daily social life, such as in laws and regulations, official documents, newspapers, magazines, and broadcasts. The currently used *jouyou kanji* was issued in 2010 and contains 2136 characters [14]. Students by the end of the sixth year in elementary school have learned 1006 of the 2136 *kanji*, and it is estimated that these 1006 *kanji* alone account for 95% of *kanji* usage in print media [15].

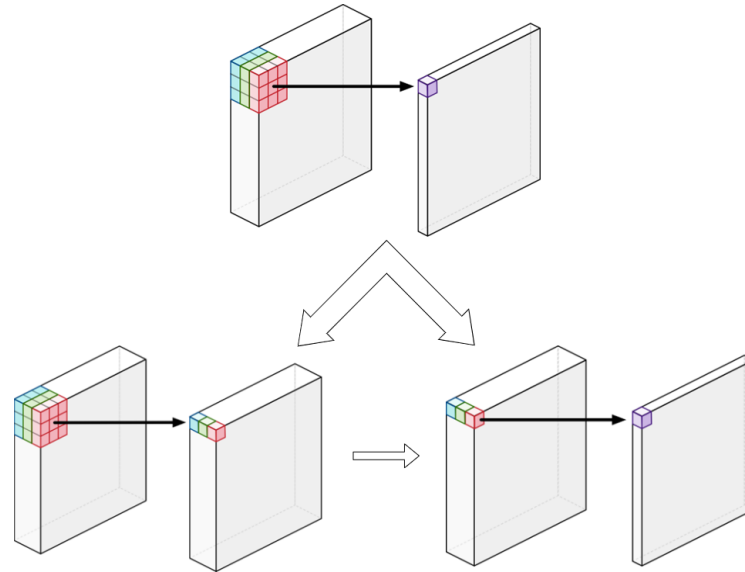
### c. *Kyouiku Kanji Grade 1*

*Kyouiku kanji* is the common name for Gakunen Betsu Kanji Haitouhyou in Gakushuu Shidou Youryou (general guide to teaching and learning) published by the Japanese Ministry of Education, Culture, Sports, Science, and Technology, which is a subset of *jouyou kanji* intended to be learned by students in grades 1 to 6 during compulsory education in Japan [16].

*Kyouiku kanji* itself is divided into six subsets, from grade 1 (Dai-1 Gakunen) to grade 6 (Dai-6 Gakunen). Here is a list of 80 *kanji* included in *kyouiku kanji* grade 1 [17]: 一 右 雨 円 王 音 下 火 花 貝 学 気 九 休 玉 金 空 月 犬 見 五 口 校 左 三 山 子 四 糸 字 耳 七 車 手 十 出 女 小 上 森 人 水 正 生 青 夕 石 赤 千 川 先 早 草 足 村 大 男 竹 中 虫 町 天 田 土 二 日 入 年 白 八 百 文 木 本 名 目 立 力 林 六

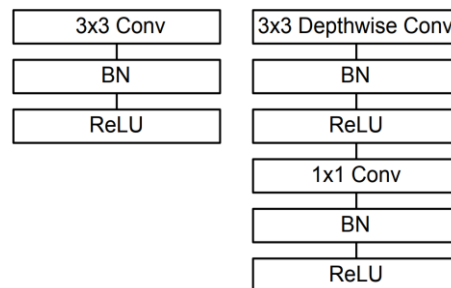
### d. MobileNet V1

The idea behind the development of MobileNet V1, an open-source mobile-first Convolutional Neural Network (CNN) architecture developed by Google, is that the convolution layer, which is essential for computer vision but quite expensive to compute, can be replaced with a depthwise separable convolution layer [18]. Depthwise separable convolutions factorize the standard convolution into depthwise convolution and pointwise convolution, as illustrated by Fig. 1.



**Fig. 1.** Standard and depthwise separable convolution

Standard convolution filters and combines inputs into one new output in one step. In MobileNet V1, the step is factorized into two steps. The depthwise convolution layer is used to apply one filter per input channel, and the pointwise convolution layer, a simple  $1 \times 1$  convolution, is used to perform a linear combination of the depthwise convolution layer outputs. MobileNet V1 uses batch normalization and Rectified Linear Unit (ReLU) for both layers, as visualized in Fig. 2. This factorization has the effect of drastically reducing computation and model size. MobileNet V1 requires between 8 and 9 times less computation than standard convolutions, with only a slight reduction in accuracy [18].

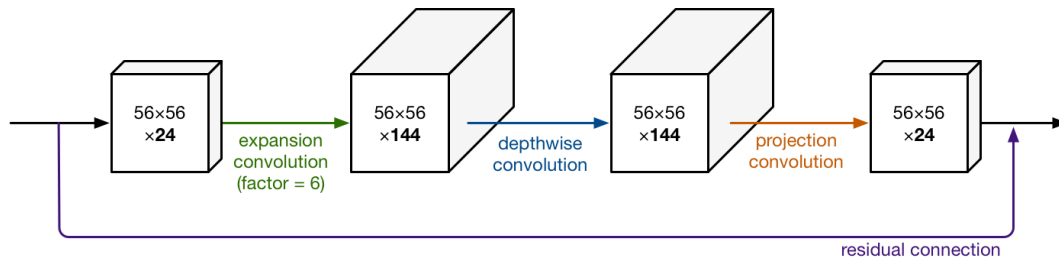


**Fig. 2.** Batch Normalization and ReLU after each Convolution Layer

In MobileNet V1 and V2, to build a smaller and computationally cheaper model, a parameter  $\alpha$  called the width multiplier can be used. The width multiplier  $\alpha$  is used to adjust the number of channels in each layer. Given the number of input channels  $M$  and output channels  $N$ , the width multiplier  $\alpha$  will set the number of channels to  $\alpha M$  and  $\alpha N$ . MobileNet V1 and V2 use  $\alpha = 1$  as the baseline and  $\alpha < 1$  as the reduced MobileNet. The width multiplier can reduce the computational cost and number of parameters quadratically around  $\alpha^2$  [18].

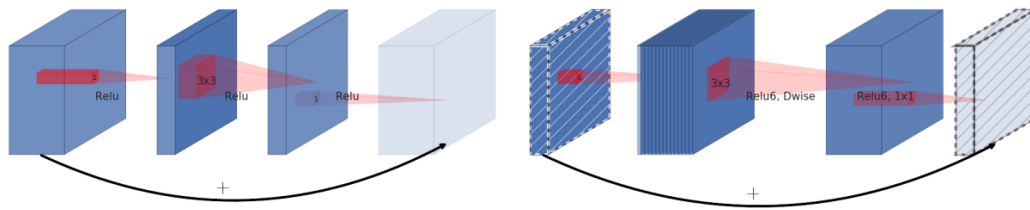
#### e. MobileNet V2

MobileNet V2 modifies the depthwise separable convolution into a bottleneck residual block by adding the utilization of linear bottleneck and inverted residual, as well as changing the usage of ReLU to ReLU6 [8]. In Fig. 3, the first block from the left, which is low-dimensional (channel), is expanded into a high-dimensional block by pointwise convolution. Then the block will be filtered by depthwise convolution and projected back to the low-dimensional block by pointwise convolution. If the first and last blocks are the same, they are added together.



**Fig. 3.** Bottleneck Residual Block

The last block in Fig. 3 is called the linear bottleneck because it uses linear activation function in place of ReLu6. The flow of data from the low dimension being expanded to the high dimension, filtered in the high dimension, and projected back to the low dimension is called inverted residual. Fig. 4 illustrates the comparison between residual block and inverted residual block.



**Fig. 4.** Residual (left) and Inverted Residual (right) Block

Table 1 lists the performance of MobileNet V1 and V2 on ImageNet by comparing the accuracy, number of parameters, and number of Multiply-Adds (MAdds) operations. It is evident that MobileNet V2 achieves higher results in those three metrics [8].

**Table 1.** MobileNet V1 and V2 Comparison

	Accuracy	Parameter	Multiply-Adds
MobileNet V1	70.6	4.2 Million	575 Million
MobileNet V2	72.0	3.4 Million	300 Million

### 3. Methodology

#### a. Data Collection

The data used in this research is secondary data, namely the ETL Character Database, which is a collection of images of approximately 1.2 million handwritten and machine-printed numbers, symbols, Latin letters, and Japanese characters, compiled into ETL-1 to ETL-9 datasets. In this research, the ETL-9B dataset was used. This database was collected by the Electrotechnical Laboratory for character recognition research from 1973 to 1984. Since January 2014, this database can be downloaded at [etlcdb.db.aist.go.jp](http://etlcdb.db.aist.go.jp).

#### b. Framework

Fig. 5 highlights the framework of this research. The research starts with the extraction of handwritten *kanji* data from the raw binary ETL-9B dataset into PNG format. The *kanji* used will also be limited to *kyouiku kanji* grade 1. Then the dataset will be divided into three different datasets, namely training, validation, and test datasets. The training and validation datasets will then be used to train the MobileNetV2 model implementation of the Keras API.

For each epoch that has been trained, the model data and metrics results will be stored as a checkpoint, so that when the training process is complete, a list of models with their metrics will be obtained for each epoch. This research uses 200 epochs. From the list, one epoch will be selected whose accuracy is the highest. The model at that epoch will then be used as the output model. This output model can then be used to recognize the test dataset, and the test metrics results will be analyzed further.

The model output will also be converted into the TensorFlow Lite model format so that it can be implemented into the Android application. After that, the Android application as a user interface will go through a testing and bug hunting process, and when it is considered that there are no more problems, then the final Android application can be deployed.

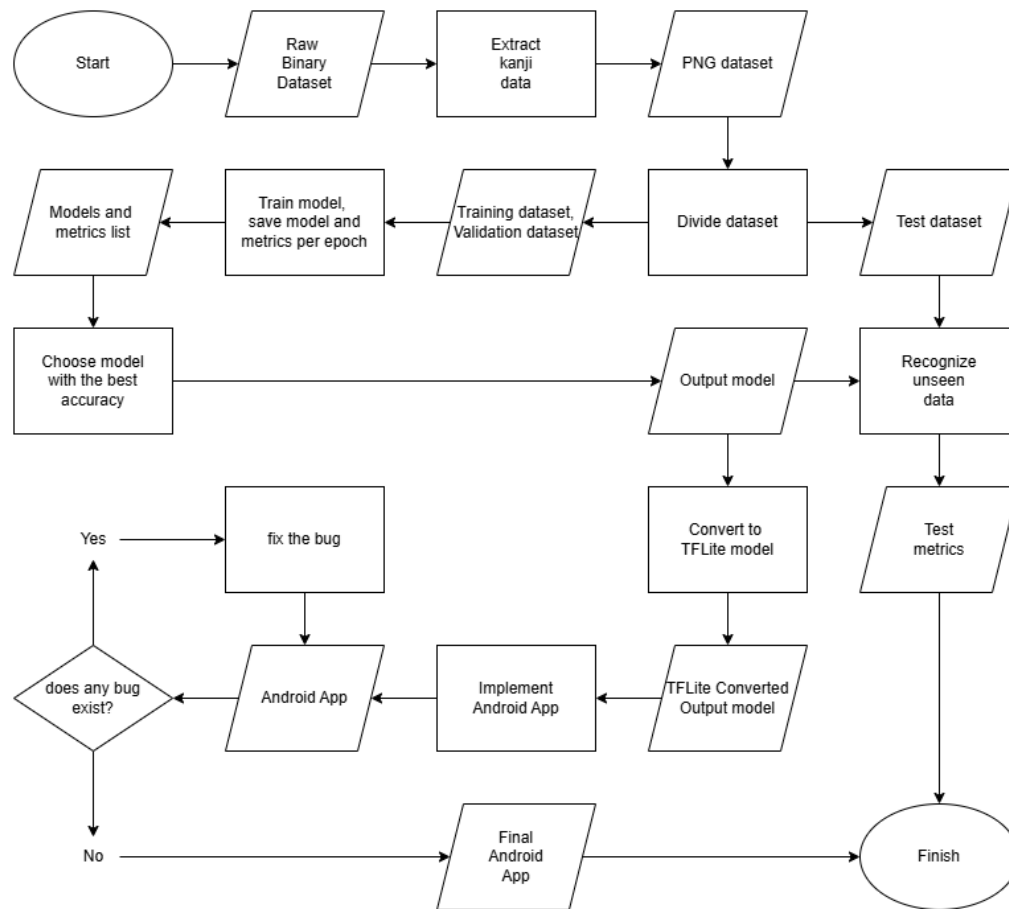


Fig. 5. Research Framework

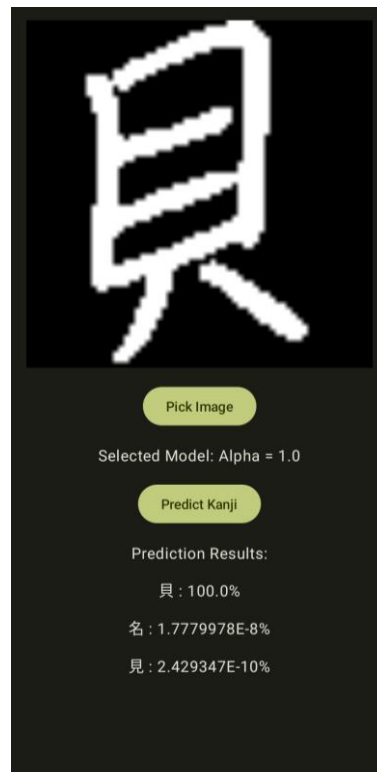
#### 4. Result and Discussion

Table 2 shows the results of validation accuracy, test accuracy, and model size for each different width multiplier alpha value. The Android application implementation result in Fig. 6 can be used as the UI for model inference. The research was conducted using a training dataset of 12800 images, a validation dataset of 1600 images, and a test dataset of 1600 images. The test dataset is used on the model at the epoch that has the highest validation accuracy value to get the test accuracy value. Each experiment uses the same settings and only differs in alpha value, with the aim of observing the relationship between accuracy and alpha value.

Table 2. Model Accuracy and Sizes

	Alpha					
	0.35	0.5	0.75	1.0	1.3	1.4
Size	6.4 MB	9.7 MB	17.4 MB	27.4 MB	44.9 MB	51.8 MB
Validation	92.0625%	95.3125%	93.1875%	97.1250%	70.6250%	47.5625%
Test	90.5000%	94.3750%	90.8125%	96.6875%	72.6875%	42.6250%
Difference	1.5625%	0.9375%	2.375%	0.4375%	-2.0625%	4.9375%

It can be observed in Table 2 that the higher the alpha value, the larger the model size. After  $\alpha = 1.0$ , the resulting accuracy value drops significantly. Furthermore, it can be observed that the accuracy relationship between alpha values for validation and test results is the same, with accuracy at  $\alpha 0.35 < 0.5 > 0.75 < 1.0 > 1.3 > 1.4$ . Finally, the model with the highest accuracy value is obtained at  $\alpha = 1.0$ , with a test value of 96.6875%.



**Fig. 6.** Android App UI for Model Inference

## 5. Conclusion

Based on the explanation in the previous chapters, it can be concluded that:

1. Handwritten *kyouiku kanji* grade 1 recognition using MobileNet V2 based on Android achieves the highest test accuracy value of 96.6875% in this research by using width multiplier alpha 1.0.
2. Handwritten *kyouiku kanji* grade 1 recognition using MobileNet V2 based on Android has been implemented.

## References

- [1] K. Hamad and M. Kaya, "A Detailed Analysis of Optical Character Recognition Technology", International Journal of Applied Mathematics Electronics and Computers, no. Special Issue-1, pp. 244–249, December 2016, doi: 10.18100/ijamec.270374.
- [2] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," Sensors, vol. 20, no. 12, p. 3344, 2020. doi: 10.3390/s20123344.
- [3] I. Ota, R. Yamamoto, S. Sako and S. Sagayama, "Online Handwritten Kanji Recognition Based on Inter-stroke Grammar," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 2007, pp. 1188-1192, doi: 10.1109/ICDAR.2007.4377103.
- [4] X. Hu, M. Inamoto, and A. Konagaya, "Recognition of Kuzushi-ji with Deep Learning Method," in Proc. Annu. Conf. JSAI, vol. JSAI2019, 33rd, Session ID 4H2-E-5-01, p. 4H2E501, Jun. 2019. doi: 10.11517/pjsai.JSAI2019.0\_4H2E501.
- [5] Solis, A. I., Zarkovacki, J., Ly, J., and Atyabi, A., "Recognition of Handwritten Japanese Characters Using Ensemble of Convolutional Neural Networks", arXiv e-prints, 2023. doi:10.48550/arXiv.2306.03954.
- [6] C. Tsai, "Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks," University Stanford, Stanford, CA, USA, 2016, pp. 405–410.

- 
- [7] J. V. T. Pérez, "Recognition of Japanese handwritten characters with machine learning techniques," B.S. thesis, Multimedia Engineering, Universidad de Alicante, 2020.
- [8] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.
- [9] "Keras Applications" Keras. <https://keras.io/api/applications/> (accessed Nov. 22, 2023).
- [10] P. M. Suski, *The Phonetics of Japanese Language*, Routledge Library Edition, vol. 59. London: Routledge, 1931.
- [11] *Jinmeiyō Kanjiyō*, Ministry of Justice, Japan, 2017. Accessed: Nov. 22, 2023. [Online]. Available: <https://www.moj.go.jp/content/001131003.pdf>
- [12] *Jōyō Kanjiyō*, Agency for Cultural Affairs, Japan, 2010. Accessed: Nov. 22, 2023. [Online]. Available: [https://www.bunka.go.jp/kokugo\\_nihongo/sisaku/joho/joho/kijun/naikaku/pdf/joyokanjiho\\_20101130.pdf](https://www.bunka.go.jp/kokugo_nihongo/sisaku/joho/joho/kijun/naikaku/pdf/joyokanjiho_20101130.pdf)
- [13] "Jōyō Kanjiyō" Japan Electronic Publishing Association. [https://www.jepa.or.jp/ebookpedia/201701\\_3385/](https://www.jepa.or.jp/ebookpedia/201701_3385/) (accessed Nov. 22, 2023).
- [14] S. Paxton and C. Svetanant, 'Tackling the Kanji hurdle: investigation of Kanji learning in non-Kanji background learners', *The International Journal of Research Studies in Language Learning*, vol. 3, no. 3, pp. 89–104, 2014.
- [15] "Kyouiku Kanji," NTT DOCOMO. <https://dictionary.goo.ne.jp/word/教育漢字/> (accessed Jun. 4, 2024).
- [16] *Shougakkō Gakushū Shidō Jōyō*, Ministry of Education Culture Sports Science and Technology, Japan, 2017. Accessed: Nov. 22, 2023. [Online]. Available: [https://www.bunka.go.jp/kokugo\\_nihongo/sisaku/joho/joho/kijun/naikaku/pdf/joyokanjiho\\_20101130.pdf](https://www.bunka.go.jp/kokugo_nihongo/sisaku/joho/joho/kijun/naikaku/pdf/joyokanjiho_20101130.pdf)
- [17] A. G. Howard et al., 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', arXiv [cs.CV]. 2017.