CNN for Image Identification of Hiragana Based on Pattern Recognition

Chaerul Umam*[1], Danang Andi Krismawan[2]

[1,2]*Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia 50131*
E-mail : chaerul@dsn.dinus.ac.id*[1], andidanang@dsn.dinus.ac.id[2]
*Corresponding author

Rabei Raad Ali[3]

[3]*Faculty of Computer Science and Information Technology, Tun Hussein Onn Malaysia University, Malaysia, 86400*
E-mail : rabei.aljawary@gmail.com

Abstract - Hiragana is one of the letters in Japanese. In this study, CNN (Convolutional Neural Network) method used as identication method, while he preprocessing used thresholding. Then carry out the normalization stage and the filtering stage to remove noise in the image. At the training stage use maxpooling and danse methods as a liaison in the training process, wherea in testing stage using the Adam Optimizer method. Here, we use 1000 images from 50 hiragana characters with a ratio of 950: 50, 950 as training data and 50 data as testing data. Our experiment yield accuracy in 95%.

Keywords – Hiragana image, CNN, Adam Optimizer, Thresholding

1. INTRODUCTION

Image recognition is very important in the evolution of artificial intelligence. In particular, finding an efficient model for reading handwritten characters is an ongoing study for very different letters [1][2], whose symbols vary drastically [3]. One of them is Japan, Japan is a developed country and many people like this country, because it is a clean country and many works of art are in demand by many people [4]. It is also an interesting language for someone who likes Japanese and characters. For example, comics, anime and films produced by the Japanese state attract people to learn Japanese [5], one of them is the Hiragana letter.

Japanese itself has 3 characters, namely Kanji letters, Hiragana letters, and Katakana letters. where Hiragana and Katakana letters are used as alphabets from Japanese, while kanji are used as root words in Japanese. And also writing in Japanese is different from writing in Indonesia, where in Indonesia the way of writing is from left to right while in Japanese writing is from top to bottom. And the writing for the next sentence is written on the left after the previous writing so that it reads from right to left. Hiragana characters were used as a smooth writing which is widely used by Japanese women. Hiragana letters exist on several syllables and have their own rules and the base letter as 46 letters. Which are used as verbs and adjectives, adverbs and are used for formal writing.

Hiragana letters are the basic letters used in writing the original Japanese vocabulary. Japanese has a sentence structure consisting of Hiragana letters, kanji and katakana [6][7]. The syllabic combination uses hiragana letters and katakana letters, while those for pictures use

kanji characters [8]. In addition, the addition of other symbols can change different meanings, therefore Japanese may have more than one combination.

In terms of character optimization, one of the methods of the neural network can be used. The process in a neural network [9] is likened to the workings of the human brain which must be trained first to get the right results with a high degree of accuracy. Each pixel that is analyzed and matched with the training data that has been carried out by the neural network method is very suitable for solving the specification case or for detecting damaged writing because the method is efficient than other methods.

Convolutional Neural Network or CNN for short is a continuation of a neural network that is good at handwriting recognition [10]. For this occasion, it will imply the convolutional neural network algorithm (CNN) neural network method [11]. In the last few years the Convolutional Neural Network has made very big breakthroughs because the Convolutional Neural Network has significantly improved in terms of image classification and processing. This is supported by a strong computational factor and a variety of training techniques. The Convolutional Neural Network algorithm is called the best model in solving problems such as object detection and others. The evidence is based on the results of previous studies. The data set used was taken containing 50 different characters and 20 different samples in each letter character. The aim of this study was to show that although using a small sample as training data, it is possible to create a learning model using a convokutional neural network to classify 50 different hiragana characters. In addition, CNN has several other models, namely CNN with 1 convosional layer, CNN with 2 layers, CNN with 3 layers and CNN with 4 layers. However, CNN also has a weakness in the long training process [9]. However, with the support of today's hardware, it is very easy to solve it. Another problem is the lack of suitable training data due to the difficulty of obtaining a dataset.
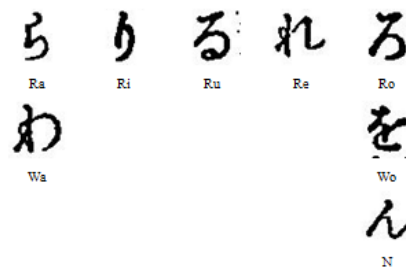
Figure 1. Hiragana Character [11]

In order to achieve the desired achievement to achieve the goal, review journals published on the internet. The learning used is the hiragana letter pattern recognition with CNN. After learning enough about the methods and concepts, a GUI and CNN (Convolutional Neural Network) architectural design were made by implementing the previously studied theories and concepts.

2. RESEARCH METHOD

2.1. Convolutional Neural Network (CNN)

Artificial intelligence is a technique used to imitate intelligence or the workings of the human brain or inanimate objects in solving problems. For this purpose, at least 3 methods have been developed namely fuzzy logic, evolutionary computation and machine learning. In ANN, there is Deep Learning where machines are trained first from the inside [12]–[15]. Deep learning is a branch of machine learning that uses deep neural networks to solve problems in the machine learning field [5], [16], [17]. Machine learning can be defined as a method based on experience with increased performance to get accurate predictions. Experience here is likened to pre-testing knowledge or knowledge that has been known before being taught.

Convolutional Neural Network is an Artificial Neural Network (ANN) method that develops and is inspired by human nerves that can distinguish and recognize an object [18]. Convolutional Neural Networks can also be used to detect and recognize an image. Convolutional Neural Network can be said that it is included in the Deep Neural Network because of its high depth, therefore it is used as an implication for an image [16], [19]. Convolutional Neural Networks are also the same as neural networks that choose rules from input to produce output (feature map). The rule consists of 3 layers: convolution, layer activation and pooling [10], see Figure 2.
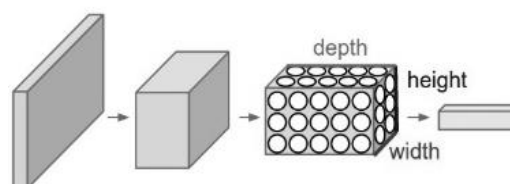

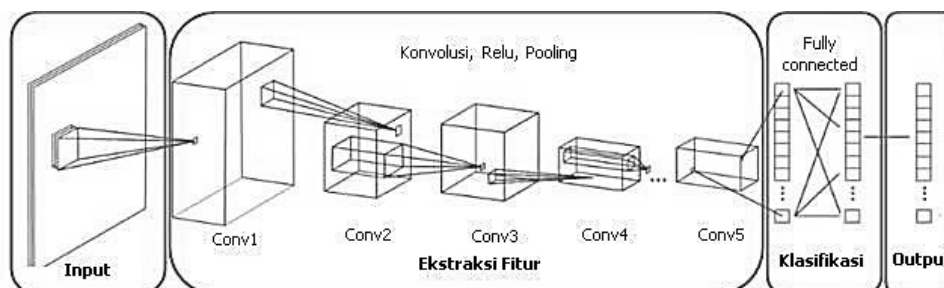Figure 2. Convolutional Neural Network (CNN) Architecture

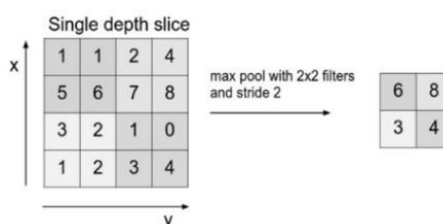Figure 3. Convolutional Neural Network (CNN) Layer

Figure 4. Maxpooling Process

CNN carries out the convolution process by moving a filter with a certain size in the form of an image, then the computer will analyze and get new information with the result of multiplying parts of the image using the filter used. The process carried out in CNN is to split the image into a few pixels. Each pixel resulting from the convolution is then used as input to get the feature representation result. This process is used as an object recognition step when the object appears. From the results, it will be made into a new array, but the array is still large, so the step to unravel the array is even smaller, or is called max pooling. The next process is fully connect using the smallest array to be inputted into other neural networks and the last neural network will be used as the key whether the image matches or not.

2.1. Proposed Method

The initial stage carried out in conducting this research is to collect data on Hiragana letters, where the data will later be used as data in training and test data. This data collection was carried out and searched from internet sources as well as some editing. In terms of data collection, of course there are limitations that are carried out, one of which is that data collection is carried out only in the form of basic Hiragana letters, and then it will be made into several classes. Where the Hiragana letters are divided into 50 classes. In one class, it contains approximately 20 character images with a pixel size of less than 83x84 and uses the .JPG format which is opened using the Windows Photos software, which is an application available on Windows. The data that has been collected in this study are 1000 handwritten image data of Hiragana letters which are divided into two, namely as test data and training data where 950 is used as training data and 50 is used as test data.

After dividing or splitting the image into several classes for testing and testing data, then the image that has been collected will be carried out in the preprocessing stage. This stage is an initial process carried out in carrying out handwriting recognition. The process carried out in the preprocessing stage is image normalization, thresholding, filtering and labeling.
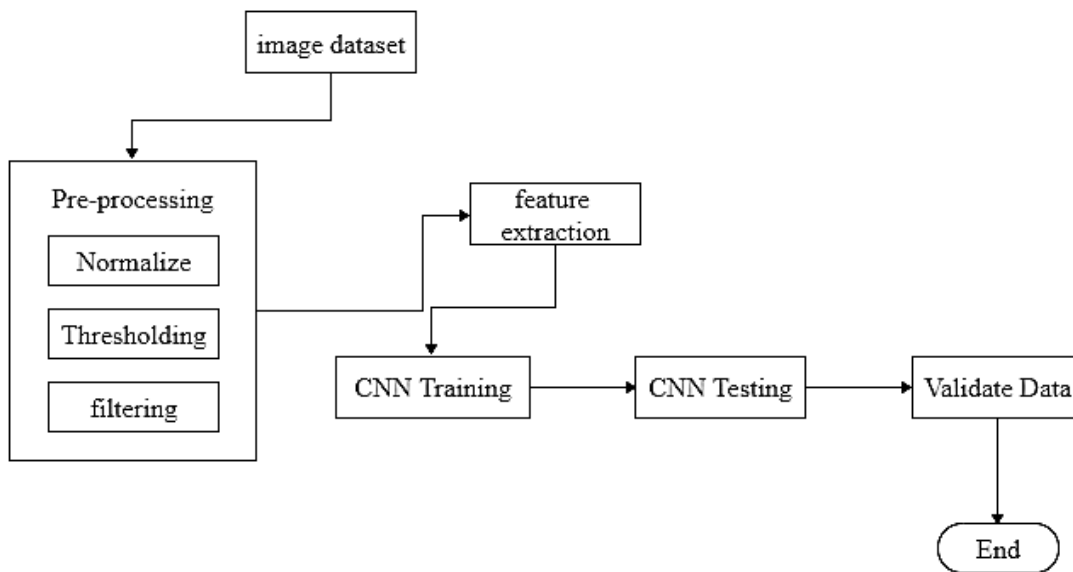
Figure 5. Proposed Method

3. RESULTS AND DISCUSSION

The data collected is a dataset in the form of a JPG file format. Each character will be tested by collecting data, namely trainer data and testing data. In the current study, 45 Hiragana vowel consonants and 5 Tenten Maru letters were used. Training data and test data are each taken from 45 Hiragana Vowels Consonants and 5 Hiragana Tenten Maru letters, where 20 samples of each letter will be tested to get a role, then you will get 1000 sample data. In this test, the test was carried out several times using different files. The author can write down the resulting level of accuracy. Testing is done by taking several samples from the dataset and taking the accuracy value according to equation (1). The results of the process at this stage produce an image like Figure 7.

$$Accuracy = \frac{number\ of\ correct\ images}{total\ image} \times 100\% \tag{1}$$

Figure 6. Example Dataset

Figure 7. Example of dataset after normalization

Then at the labeling stage, each hiragana letter is labeled according to the class to be determined. With the labeling stage, the data is in accordance with the predetermined classes. Can be seen in Figure 8.

Figure 8. Class Hiragana

Based on Figure 8, then proceed with the modeling stage using the CNN (Convolutional Neural Network) method. After labeling and separating the training and test data, it is continued by training the dataset with CNN by forming a network architecture to get the accuracy of the results. Data input was carried out by taking data from a previously collected dataset in the form of handwritten hiragana letters which were used as a database for training using CNN. At this stage there are 950 training data images, where each character has 20 training data samples. The epoch data used is 200 where the process at times of season passes 200 times with the aim of getting the smallest error results to produce more optimal accuracy. While the loss used is 'categorical_crossentropy' with added Adam's optimizer using code below.

```
def baseline_model():
    # create model
    model = Sequential()
    model.add(Conv2D(50, (5, 5), input_shape=(1, 84, 83),
data_format='channels_first', activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

```
return model
```

Convolution is the combination of two series of a number in an image to produce a new series. The numbers contained in the image are in the form of an array matrix. Based on the input image, the image has a pixel size of 83x84, which can explain the size of the pixel height and width of the image. And the input image has 3 colors red, green and blue or called RGB. Each pixel already has a different pixel, the input image will be convoluted with the filter that has been determined. A filter is another block with a smaller height and width but the same depth as the original image. The purpose of using a filter is to determine the pattern which will be recognized by multiplying the value in the input matrix, therefore the values in the column and row will depend heavily on the recognized pattern.

In the last stage, validation was carried out by testing using 50 hiragana letter test data, where an accuracy value of 95% was obtained, but previously a comparison of 800: 200 was also carried out where obtaining an accuracy of 82% in detail can be seen in Table 1.

Table 1. Testing Using CNN with data percentage 800 : 200

Epoch	Dataset Hiragana				Epoch	Dataset Hiragana			
	Loss	Accept	Val_loss	Val_acc		Loss	Accept	Val_loss	Val_acc
1/200	2.1354	0.4925	1.1724	0.7200	37/200	1.0747e-05	1.0000	1.0882	0.8150
2/200	0.1776	0.9450	0.9189	0.7800	38/200	1.0102e-05	1.0000	1.0898	0.8150
3/200	0.0350	0.9900	0.8481	0.7850	39/200	9.5251e-06	1.0000	1.0918	0.8150
4/200	0.0077	0.9975	1.1077	0.7950	40/200	8.9631e-06	1.0000	1.0932	0.8150
5/200	0.0020	1.0000	1.0217	0.8150	41/200	8.4545e-06	1.0000	1.0963	0.8150
6/200	4.4331e-04	1.0000	1.0194	0.8100	42/200	7.9924e-06	1.0000	1.0979	0.8150
7/200	2.8917e-04	1.0000	1.0201	0.8150	43/200	7.5461e-06	1.0000	1.1012	0.8150
8/200	2.1999e-04	1.0000	1.0243	0.8150	44/200	7.1305e-06	1.0000	1.1016	0.8150
9/200	1.7509e-04	1.0000	1.0248	0.8200	45/200	6.7417e-06	1.0000	1.1043	0.8150
10/200	1.4316e-04	1.0000	1.0261	0.8200	46/200	7.3243e-06	1.0000	1.1563	0.8150
11/200	1.2026e-04	1.0000	1.0303	0.8200	47/200	6.9358e-06	1.0000	1.1594	0.8150
12/200	1.0213e-04	1.0000	1.0318	0.8200	48/200	6.5723e-06	1.0000	1.1610	0.8150
13/200	8.8440e-05	1.0000	1.0349	0.8200	49/200	6.2373e-06	1.0000	1.1640	0.8150
14/200	7.6692e-05	1.0000	1.0380	0.8200	50/200	5.9139e-06	1.0000	1.1668	0.8150
15/200	6.7537e-05	1.0000	1.0417	0.8200	51/200	5.6109e-06	1.0000	1.1700	0.8150
16/200	5.9838e-05	1.0000	1.0454	0.8200	52/200	5.3343e-06	1.0000	1.1719	0.8150
17/200	5.3281e-05	1.0000	1.0479	0.8200	53/200	5.0625e-06	1.0000	1.1748	0.8150
18/200	4.7699e-05	1.0000	1.0480	0.8200	54/200	4.8186e-06	1.0000	1.1767	0.8150
19/200	4.2927e-05	1.0000	1.0514	0.8200	55/200	4.5801e-06	1.0000	1.1786	0.8150
20/200	3.8931e-05	1.0000	1.0537	0.8150	56/200	4.3576e-06	1.0000	1.1812	0.8150
21/200	3.5324e-0	1.0000	1.0564	0.8150	57/200	4.1499e-06	1.0000	1.1846	0.8150
22/200	3.2263e-05	1.0000	1.0581	0.8150	58/200	3.9482e-06	1.0000	1.1863	0.8150
23/200	2.9550e-05	1.0000	1.0597	0.8150	59/200	3.7599e-06	1.0000	1.1884	0.8150
24/200	2.7114e-05	1.0000	1.0631	0.8150	60/200	3.5840e-06	1.0000	1.1914	0.8150
25/200	2.4934e-05	1.0000	1.0631	0.8150	61/200	3.4148e-06	1.0000	1.1932	0.8150
26/200	2.3045e-05	1.0000	1.0661	0.8150	62/200	3.2583e-06	1.0000	1.1960	0.8150
27/200	2.1352e-05	1.0000	1.0685	0.8150	63/200	3.1060e-06	1.0000	1.1973	0.8150
28/200	1.9823e-05	1.0000	1.0703	0.8150	64/200	2.9623e-06	1.0000	1.2005	0.8150
29/200	1.8382e-05	1.0000	1.0725	0.8150	65/200	2.8279e-06	1.0000	1.2027	0.8150
30/200	1.7086e-05	1.0000	1.0736	0.8150	66/200	2.7014e-06	1.0000	1.2054	0.8150
31/200	1.5942e-05	1.0000	1.0771	0.8150	67/200	2.3945e-06	1.0000	1.1360	0.8150
32/200	1.4872e-05	1.0000	1.0783	0.8150	68/200	2.2842e-06	1.0000	1.1379	0.8150
33/200	1.3904e-05	1.0000	1.0794	0.8150	69/200	2.1840e-06	1.0000	1.1408	0.8150
34/200	1.3016e-05	1.0000	1.0813	0.8150	70/200	2.0831e-06	1.0000	1.1425	0.8150
35/200	1.2192e-05	1.0000	1.0843	0.8150	....	....	....	....	....
36/200	1.1431e-05	1.0000	1.0850	0.8200	200/200	1.1921e-07	1.0000	1.3130	0.8200

Following are the results of the second experiment with training and test data with a ratio of 950: 50, 950 as training data and 50 data used as testing data, the results can be seen in Table 2.

Table 2. Testing Using CNN with data percentage 950 : 50

Epoch	Dataset Hiragana				Epoch	Dataset Hiragana			
	Loss	Accept	Val_loss	Val_acc		Loss	Accept	Val_loss	Val_acc
1/50	1.9036	0.5418	0.6964	0.8500	26/50	1.8610e-05	1.0000	0.2034	0.9500
2/50	0.2062	0.9418	0.4053	0.9500	27/50	1.6991e-05	1.0000	0.2023	0.9500
3/50	0.0507	0.9786	0.2869	0.9000	28/50	1.5555e-05	1.0000	0.2013	0.9500
4/50	0.0376	0.9898	0.5277	0.9500	29/50	1.4263e-05	1.0000	0.2001	0.9500
5/50	0.0067	0.9990	0.3658	0.9500	30/50	1.3135e-05	1.0000	0.2010	0.9500
6/50	0.0096	0.9969	0.3099	0.9500	31/50	1.2092e-05	1.0000	0.2010	0.9500
7/50	0.0016	1.0000	0.2648	0.9500	32/50	1.1183e-05	1.0000	0.2016	0.9500
8/50	4.1177e-04	1.0000	0.2494	0.9500	33/50	1.0349e-05	1.0000	0.2016	0.9500
9/50	2.5729e-04	1.0000	0.2438	0.9500	34/50	9.5877e-06	1.0000	0.2017	0.9500
10/50	1.8467e-04	1.0000	0.2392	0.9500	35/50	8.9025e-06	1.0000	0.1989	0.9500
11/50	1.4114e-04	1.0000	0.2343	0.9500	36/50	8.2697e-06	1.0000	0.2003	0.9500
12/50	1.1231e-04	1.0000	0.2314	0.9500	37/50	7.7131e-06	1.0000	0.1998	0.9500
13/50	9.2176e-05	1.0000	0.2280	0.9500	38/50	7.1740e-06	1.0000	0.2003	0.9500
14/50	7.7010e-05	1.0000	0.2252	0.9500	39/50	6.6946e-06	1.0000	0.1996	0.9500
15/50	6.5540e-05	1.0000	0.2230	0.9500	40/50	6.2612e-06	1.0000	0.1996	0.9500
16/50	5.6431e-05	1.0000	0.2213	0.9500	41/50	5.8497e-06	1.0000	0.1979	0.9500
17/50	4.9139e-05	1.0000	0.2188	0.9500	42/50	5.4744e-06	1.0000	0.1979	0.9500
18/50	4.3190e-05	1.0000	0.2171	0.9500	43/50	5.1240e-06	1.0000	0.1991	0.9500
19/50	3.8227e-05	1.0000	0.2154	0.9500	44/50	4.8044e-06	1.0000	0.1973	0.9500
20/50	3.4033e-05	1.0000	0.2135	0.9500	45/50	4.5041e-06	1.0000	0.1968	0.9500
21/50	3.0466e-05	1.0000	0.2107	0.9500	46/50	4.2294e-06	1.0000	0.1962	0.9500
22/50	2.7386e-05	1.0000	0.2098	0.9500	47/50	3.9680e-06	1.0000	0.1950	0.9500
23/50	2.4729e-05	1.0000	0.2081	0.9500	48/50	3.7257e-06	1.0000	0.1933	0.9500
24/50	2.2414e-05	1.0000	0.2067	0.9500	49/50	3.4997e-06	1.0000	0.1952	0.9500
25/50	2.0376e-05	1.0000	0.2051	0.9500	50/50	3.2962e-06	1.0000	0.1916	0.9500

After going through several stages, it's time to test the accuracy of the data and its loss using the Adam optimizer.

```
# Compile model
     model.compile(loss='categorical_crossentropy',          optimizer='adam',
metrics=['accuracy'])
     return model
#NUMBER OF CLASSES
num_epochs = 200 #jumlah iterasi
    metrics = model.fit(x_train, y_train, validation_data = (x_test, y_test),
epochs=num_epochs, batch_size=10) #returns val_loss, val_acc, loss, acc
    modelJSON = model.to_json()
```

Where using epoch iterations of 50 with validation accuracy results from different experiments using 950 training samples and 50 testing tests as below.

```
loss: 3.2962e-06- acc: 1.0000-val_loss: 0.1916-val_acc: 0.9500
```
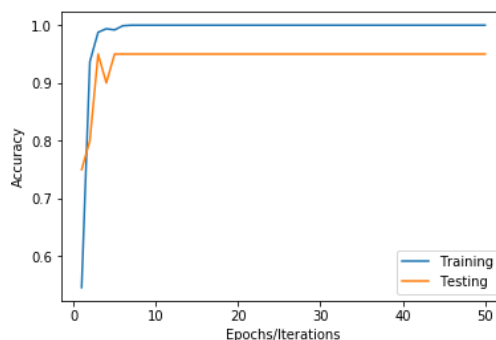

Figure 8. Graph of training and testing results

69

It can be seen in the image above, that the accuracy results show a large loss of 33% with the highest accuracy value of 100%. For validation data, it is known that the loss value is 19% and the accuracy is 95%. it can be stated that the accuracy rate obtained is 95%. For a classification with many classes, namely 50 classes, with this accuracy it can be stated that it is good to run. It can be seen in Figure 8.

4. CONCLUSION

The Convolutional Neural Network (CNN) model uses input measuring 83x84 and uses 200 Epochs. The dataset used is 1000 images. The training process uses 950 images while the testing or validation process uses 50 images. With the results of the level of accuracy training or training and testing in the recognition of the hiragana character handwriting pattern of 100% for its accuracy. Meanwhile, 95% is validated for accuracy. When testing with a ratio of 800: 200, the accuracy value obtained is only 82%. Thus it can be stated that the more training data used as training data, the higher accuracy and more accurate it will be.

REFERENCES
[1]   C. L. Liu, "Normalization-cooperated gradient feature extraction for handwritten character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1465–1469, 2007.
[2]   Y. Wang, Z. Lian, Y. Tang, and J. Xiao, "Boosting scene character recognition by learning canonical forms of glyphs," *Int. J. Doc. Anal. Recognit.*, vol. 22, no. 3, pp. 209–219, 2019.
[3]   Q. Miao, R. Liu, P. Zhao, Y. Li, and E. Sun, "A Semi-supervised Image Classification Model based on Improved Ensemble Projection Algorithm," *IEEE Access*, vol. 4, no. c, pp. 1–1, 2017.
[4]   S. D. Budiwati, J. Haryatno, and E. M. Dharma, "Japanese character (Kana) pattern recognition application using neural network," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, 2011, no. July.
[5]   M. Karki, Q. Liu, R. DiBiano, S. Basu, and S. Mukhopadhyay, "Pixel-level reconstruction and classification for noisy handwritten bangla characters," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2018-Augus, pp. 511–516, 2018.
[6]   F. Chang, C. C. Lin, and C. J. Chen, "Applying a hybrid method to handwritten character recognition," in *Proceedings - International Conference on Pattern Recognition*, 2004, vol. 2, no. 1, pp. 529–532.
[7]   M. Nguyen and N. A. V. B, "SVMs with Deep Learning and Random," in *Springer*, vol. 2, Springer International Publishing, 2019, pp. 157–172.
[8]   S. Das and S. Banerjee, "An Algorithm for Japanese Character Recognition," *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 1, pp. 9–15, 2014.
[9]   H. A. Morsy, "Developing a New CCN Technique for Arabic Handwritten Digits Recognition," *Int. J. Recent Technol. Eng.*, vol. 9, no. 3, pp. 520–524, 2020.
[10]   Charlie Tsai, "Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks," 2016.
[11]   H. Shah, R. Rajasekaran, and J. Masih, "A Comparative Study of Classification Algorithms with Varying Training Dataset Sizes on Cursive Hiragana Characters," *J. Xi'an Univ. Archit. Technol.*, vol. XII, no. Viii, pp. 1388–1396.
[12]   T. Purwaningsih, T. Nurhikmat, and P. B. Utami, "Image classification of Golek puppet images using convolutional neural networks algorithm," *Int. J. Adv. Soft Comput. its Appl.*, vol. 11, no. 1, pp. 34–45, 2019.
[13]   L. Fu *et al.*, "Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-

tiny model," *Precis. Agric.*, no. 0123456789, 2020.

[14]   S. Mishra, R. Sachan, and D. Rajpal, "Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition," *Procedia Comput. Sci.*, vol. 167, pp. 2003–2010, 2020.

[15]   M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. Rajendra Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cogn. Syst. Res.*, vol. 54, pp. 176–188, May 2019.

[16]   L. Mou, P. Ghamisi, and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[17]   M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic Fruit Classification Using Deep Learning for Industrial Applications," *IEEE Trans. Ind. Informatics*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019.

[18]   K. Kaplan, Y. Kaya, M. Kuncan, and H. M. Ertunç, "Brain tumor classification using modified local binary patterns (LBP) feature extraction methods," *Med. Hypotheses*, vol. 139, no. March, p. 109696, Jun. 2020.

[19]   I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," *2017 4th Int. Conf. Signal Process. Commun. Networking, ICSCN 2017*, pp. 1–6, 2017.