Scientific Journal of Informatics
Vol. 11, No. 2, May 2024

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Hiragana Character Classification Using Convolutional Neural Networks Methods based on Adam, SGD, and RMSProps Optimizer

Ibnu Utomo Wahyu Mulyono[1*], Yupie Kusumawati[2], Ajib Susanto[1], Christy Atika Sari[1], Hussain Md Mehedul Islam[3], Mohamed Doheir[4]

[1]Department of Informatics Engineering, Faculty of Computer Sciences, Universitas Dian Nuswantoro, Indonesia
[2]Department of Information Systems, Faculty of Computer Sciences, Universitas Dian Nuswantoro, Indonesia
[3]Department of Software Engineer, The Mathworks, Inc., United States
[4]Universiti Teknikal Malaysia Melaka, Malacca, Malaysia

Abstract.
Purpose: Hiragana image classification poses a significant challenge within the realms of image processing and machine learning. Despite advances, achieving high accuracy in Hiragana character recognition remains elusive. In response, this research attempts to enhance recognition precision through the utilization of a Convolutional Neural Network (CNN). Specifically, the study explores the efficacy of three distinct optimizers like Adam, Stochastic Gradient Descent with Momentum (SGDM), and RMSProp in improving Hiragana character recognition accuracy.
Methods: This research adopts a systematic approach to evaluate the performance of a Convolutional Neural Network (CNN) in the context of Hiragana character recognition. A meticulously prepared dataset is utilized for in-depth testing, ensuring robustness and reliability in the analysis. The study focuses on assessing the effectiveness of three prominent optimization methods: Stochastic Gradient Descent (SGD), RMSProp, and Adam.
Result: The results of the model performance evaluation show that the highest accuracy was obtained from the RMSP optimizer with an F1-Score reaching 99.70%, while the highest overall accuracy was 99.87% with the Adam optimizer. The analysis is carried out by considering important metrics such as precision, recall, and F1-Score for each optimizer.
Novelty: The performance results of the developed model are compared with previous studies, confirming the effectiveness of the proposed approach. Overall, this research makes an important contribution to Hiragana character recognition, by emphasizing the importance of choosing the right optimizer in improving the performance of image classification models.

Keywords: Hiragana, Image classification, Convolutional neural network, Confusion matrix
Received March 2024 / **Revised** April 2024 / **Accepted** May 2024

INTRODUCTION

Hiragana characters are one of the important components of Japanese characters used in everyday writing and communication [1]. Hiragana consists of simple characters that represent syllables and sounds in Japanese. A strong understanding of Hiragana characters is very important in learning Japanese, especially for new learners who want to master the basics of the language. However, learning Hiragana characters often requires a long time and demands high concentration. Analysis of the problem reveals that the lack of efficient and interactive learning aids can be a major obstacle in learning Hiragana characters [2], [3], [4]. Traditional learning methods often fail to engage students and make the learning process more tedious. The specific problem addressed in this research is the inefficiency and lack of interactivity in current Hiragana learning aids, which hampers effective language acquisition. Learners often struggle with memorization and recognition of characters due to repetitive and monotonous practice methods. Consequently, there is a need for innovative tools that can facilitate a more engaging and efficient learning experience. To overcome these challenges, researcher propose the development of a Convolutional Neural Network (CNN)-based application tailored for classifying and learning Hiragana characters. CNNs are powerful in image recognition tasks and can significantly enhance the learning process by providing immediate feedback and interactive learning experiences. This application aims to transform the traditional learning methods by integrating advanced machine learning techniques, thus making the learning process

[*]Corresponding author.
Email addresses: ibnu.utomo.wm@dsn.dinus.ac.id (Mulyono)
DOI: [10.15294/sji.v11i2.2313](https://doi.org/10.15294/sji.v11i2.2313)

more effective and enjoyable for Japanese language learners. The significance of this research lies in its potential to bridge the gap in existing learning tools by introducing a novel, technology-driven approach to language education.

Convolutional Neural Networks (CNN) have become the main foundation in various image processing and pattern recognition applications [5], [6], [7]. In the context of this research, CNN will play a key role in classifying Hiragana characters [7], [8]. The basic concept of CNN is similar to the function of the human brain, where connected artificial neurons will process information in Hiragana character image data. The ability of CNN to understand and recognize complex patterns in data makes it a very potential tool for solving Hiragana character classification problems [9], [10], [11]. Here, the aim of experiment is to develop a CNN-based application that is able to classify Hiragana characters with high accuracy. The application development process will include key steps such as collecting a labeled Hiragana character dataset, data preprocessing to prepare suitable data for model training, selecting and configuring an optimal CNN architecture, and training and evaluating the model using relevant metrics. The app will not only serve as a learning aid for Hiragana characters, but will also include additional features such as pronunciation training and interactive exercises to enrich the Japanese language learning experience.

In Literature Reviews related to image analysis, the use of Convolutional Neural Network (CNN) methods has been the subject of in-depth discussion [6], [7], [12], [13], [14], [15]]. Many researchers have adopted the CNN approach to perform image classification with high precision and accuracy, especially in the context of character classification such as Hiragana. Research [8] provide an interesting view regarding Hiragana character recognition using the Convolutional Neural Network (CNN) method. In his study, researcher focused on using preprocessing techniques with a thresholding approach to prepare data before the identification process. Next, a normalization and filtering stage is implemented to remove noise that may be present in the character image. At the training stage, researcher utilizes maxpooling and dense techniques as key approaches in the model training process. Interestingly, in the testing stage, the Optimizer Adam method is adopted to ensure the developed model has optimal performance. In his experiments, researcher succeeded in collecting a dataset of 1000 images representing 50 Hiragana characters, with a ratio of 950 data for training and 50 data for testing. As a result, with this approach, researcher managed to achieve an accuracy rate of 95%, showing the effectiveness of his proposed approach in classifying Hiragana characters.

Research [16] have proposed a Hiragana character classification method using a machine learning-based approach with minimal training data, prioritizing efficiency in computing power and time consumption. His work shows that by utilizing image recognition techniques, the proposed model is able to process samples from the dataset effectively. Furthermore, researchers have implemented six different models based on convolutional neural networks (CNN), using preprocessed image templates. In his research, researcher divided the dataset into training and validation data with a ratio of 5:95, which then resulted in the highest accuracy of 96.95% among the various models tested. These results demonstrate significant progress in the field of Hiragana character classification, especially when compared with the previously existing 80:20 training ratio-based classification.

Research [17] shows the importance of using the Convolutional Neural Network (CNN) method in recognizing handwriting of Hiragana characters in Japanese. According to this researcher, the Hiragana writing process has basic stroke rules that must be adhered to, namely curved lines and strokes with the writing direction from top to bottom or from left to right. In his experiment, researcher used 1000 image datasets with a variety of 50 characters, where each character had 20 image samples. For evaluation, he divided the dataset in a ratio of 70:30 and 60:40. The results show that using a 70:30 dataset split produces an accuracy of 86.5%, while with a 60:40 ratio, the accuracy drops to 83%. These findings confirm that the proportion of dataset divisions affects model performance, and the emphasis in this research is to achieve optimal accuracy in Hiragana character recognition via the CNN approach.

Based on an analysis of three relevant pieces of literature, the approach proposed in this research combines the power of a Convolutional Neural Network (CNN) with the application of various optimizers that have been proven effective in previous literature. Specifically, this research will utilize the Adam optimizer, which is known for its ability to adapt the learning rate dynamically and efficiently, to improve model convergence and accuracy. In addition, to get a comprehensive picture and compare performance, this research also considers the use of Stochastic Gradient Descent (SGD), which has become one of the classic optimization methods in machine learning. Not only that, this approach also includes testing with the RMSProp optimizer, which is known to be effective in dealing with learning rate scaling problems and non-stationary variables. By integrating these three optimizers, this research seeks to create a robust and adaptive framework to increase the efficiency and accuracy of Hiragana character classification using the CNN approach.

METHODS

The flow begins with the data collection stage, where the Hiragana character dataset is collected and prepared for the next process. After collecting data, the next step is to divide the data into two subsets, namely 80% of the data for the training stage and 20% for the testing stage. In the training stage, the CNN model is applied and trained using three different optimizers: Adam, SGDM, and RMSProp, to compare the effectiveness and performance of each optimizer in the training process. After the model is trained using the three optimizers, the evaluation stage is carried out using the Confusion Matrix to measure the performance and accuracy of the model. Confusion Matrix provides a clear picture of how well the model is at classifying Hiragana characters by comparing predictions with actual labels. After evaluation, the final stage is classification testing, where the trained model is re-evaluated using test data to measure accuracy, precision, recall, and other metrics to determine the model's effectiveness in classifying Hiragana characters. This flow can be seen in Figure 1.

Figure 1. Experiment stages

Data collection

The dataset obtained from the Kaggel Platform resource, consists of a total of 4,600 image samples divided evenly, namely 100 samples per class, with a total of 46 classes. Each class represents a specific Hiragana character, ensuring a balanced and comprehensive representation of all characters in the alphabet Hiragana. With this dataset structure, research can ensure that the models developed will be trained with sufficient and representative data for each character, enabling high accuracy and good generalization when the classification process is carried out. Sample datasets can be seen in Figure 2.

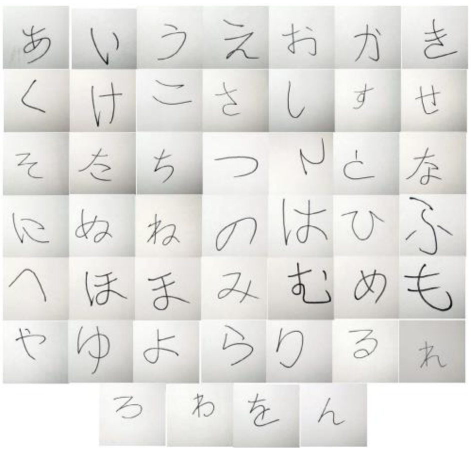
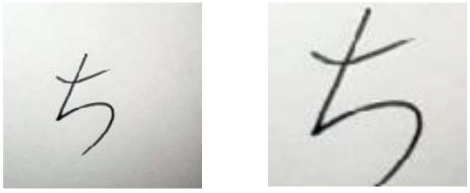


Figure 2. Framework of k-medoids clustering

Pre-processing

In this research, Hiragana characters serve as the focal point, yet details regarding the preprocessing of the images are pivotal for understanding the data preparation process thoroughly. The preprocessing stages, including cropping, resizing, and normalization, are integral to ensuring the suitability of the dataset for model training [12], [13]. Firstly, cropping is implemented to eliminate irrelevant parts from the image, such as excessive white background or any surrounding noise that may not contribute to character recognition. This step is crucial for enhancing the signal-to-noise ratio and focusing the model's attention on the relevant features of the characters. Secondly, resizing the images to a consistent size, such as 256 x

256 pixels, ensures uniformity and facilitates compatibility with the model architecture [14]. This consistency optimizes computational efficiency and enables effective feature extraction during the training process. Finally, normalization adjusts the pixel values of the images to a standardized scale, enhancing model convergence and stability during training. Regarding the statement about meeting the model's needs, the criteria for cropping and resizing are guided by the necessity to provide the model with clear, relevant input data while minimizing computational complexity. For instance, irrelevant parts may include excessive whitespace or artifacts that do not contribute to character recognition, while the model's needs refer to ensuring the images are of a size and format compatible with the model architecture. Therefore, by meticulously implementing these preprocessing stages, the dataset is carefully curated to optimize the performance of the model during training and ultimately improve Hiragana character recognition accuracy. Pre-processing samples can be seen in Figure 3.



(a) Original Sample (b) Pre-Processed

Figure 3. Image pre-processing

Figure 3 illustrates the process of image pre-processing applied to Hiragana character samples. In Figure 3 (a) are the original sample image, which may contain various imperfections such as noise, uneven lighting, and background distractions. Figure 3 (b) demonstrates the pre-processed image, where these imperfections have been effectively mitigated. The enhanced image in Figure 3 (b) shows a clearer and more defined representation of the Hiragana character, with improved contrast and sharpness.

Convolutional neural network

Convolutional Neural Network is a method included in the Feed Forward Neural Network class which is inspired by the visual cortex of the brain and is specifically for processing data that has a grid structure [15], [16]. CNN has several types of layers that can be used, namely subsampling layers, convolutional layers, loss layers and fully connected layers [2], [17], [18], [19]. CNN is an important foundation in Hiragana classification which is able to automatically understand, extract and classify unique features from Hiragana letters [20]. Layer-based CNN can be seen in Figure 4.

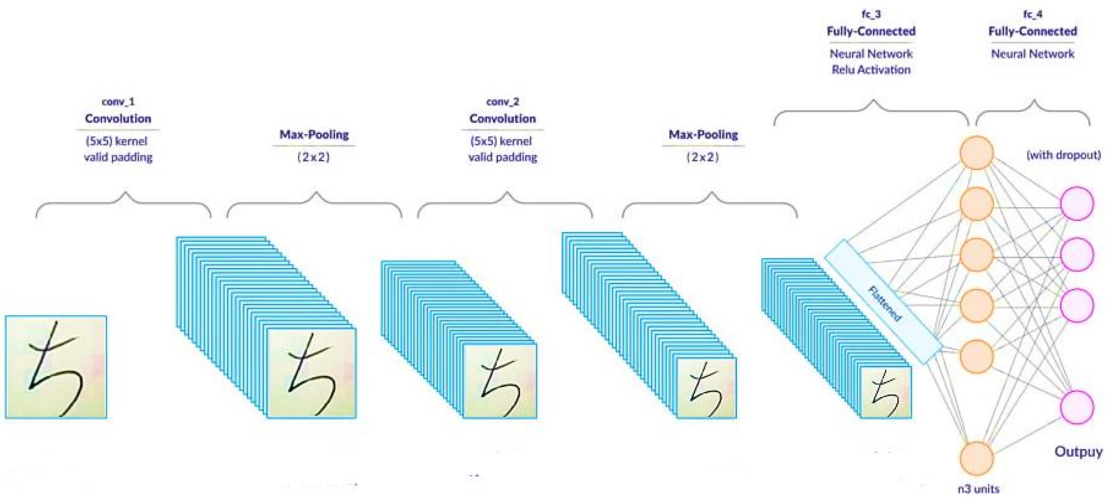


Figure 4. Proposed CNN

Hyperparameter and parameter initialization

The training strategy utilized in this study involves the implementation of three optimization algorithms: Adam, RMSProp, and Stochastic Gradient Descent with Momentum (SGDM). Adam, short for Adaptive Moment Estimation, combines elements of both RMSProp and Momentum optimization methods. It computes adaptive learning rates for each parameter by incorporating both the first and second moments of the gradients. RMSProp, on the other hand, adapts the learning rate for each parameter based on the moving average of squared gradients, thereby addressing the issue of vanishing or exploding gradients. SGDM Stochastic integrates momentum into the stochastic gradient descent algorithm, enabling faster convergence by dampening oscillations in the optimization process [21], [22], [23], [24].

Based on Learning CNN Architecture in Figure 4, Parameters are determined to ensure optimal and efficient model training. Specifically, this study set the maximum number of epochs at 8 times, indicating a focus on careful training iteration to prevent overfitting or excess training. Next, the learning rate of the model was set at 0.0001, a value chosen carefully to ensure stable and effective convergence during the training process. Additionally, the validation frequency is also set at 30 intervals, allowing periodic evaluation of the model to ensure optimal performance. To maximize training efficiency and speed, this research was carried out utilizing GPU resources, speeding up the computing process and enabling the handling of large data volumes more quickly and effectively.

Evaluation method

Confusion matrix is an evaluation tool commonly used in classification analysis to measure model performance [25]. This matrix describes the extent to which the classification model is able to correctly predict instances of each class. Consisting of four cells: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), the confusion matrix presents the number of predictions that match and do not match reality. From this matrix, a number of evaluation metrics can be calculated, such as accuracy, precision, recall, and F1 value, providing a comprehensive view of the model's performance in dealing with diverse datasets. Confusion matrices help researchers and practitioners to understand more deeply the strengths and weaknesses of classification models [26], [27], facilitating more informed decision making in the development and improvement of such models [28], as seen in equation (1) until equation (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision \times Recall} \tag{4}$$

RESULTS AND DISCUSSIONS

The first stage in this research, before going through training, pre-processing steps are carried out which can be seen in Algorithm 1. After the pre-processing process is complete, the next step in this research is to initialize the Convolutional Neural Network (CNN) based layer. Initializing this layer is a crucial stage that allows the model to understand and extract important features from the previously processed dataset. The initialization of the CNN layer can be seen in Algorithm 2. After adding the layers mentioned in algorithm 2, the CNN model then undergoes a training process. The training process was conducted on a machine equipped with a Ryzen 7 5600 processor, an RTX 2060 Super graphics card, 16GB of RAM, and a 2TB SSD. At this stage, the model is given training data to learn important features from the Hiragana character dataset that has been previously processed. During the training process, the model parameters will be adjusted based on the calculated loss function, with the aim of optimizing the model's performance in classifying the images. Details of the training progress graph can be found and analyzed further in Figure 5.

Algorithm 1: Pre-Processing

```
Function Resize (image, target_size):
   image_resized = Resize (image, target_size)
   Return image_resized
End Function
Function Crop (image, crop_size):
   start_x = (width(image) − crop_size) / 2
   start_y = (height(image) − crop_size) / 2
   cropped_image = image[start_x: end_x, start_y: end_y]
   Return cropped_image
End Function
Function Normalize (image):
   mean_value = hitung_rata_rata(image)
   std_deviation = hitung_standar_deviasi(image)
   norm_image = (image − mean_value) / sdv
   Return normalized_image
End Function
```

Algorithm 2: CNN Layers

```
Function Inisialisasi_Model_CNN ():
   model = Sequential ()
      model.add (Conv2D(filters = 32, kernel_size = (3, 3), activation = 'relu', input_shape = (input_dim)))
   model.add(MaxPooling2D(pool_size = (2, 2)))
   model.add(Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
   model.add(MaxPooling2D(pool_size = (2, 2)))
   model.add(Flatten())
         model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
   return model
End Function
```

Based on the algorithms provided above, the image pre-processing stage (Algorithm 1) consists of three functions: Resize, Crop, and Normalize. The Resize function resizes the input image to the target size, while the Crop function extracts a central region from the image based on the specified crop size. Subsequently, the Normalize function calculates the mean value and standard deviation of the image and normalizes it accordingly. Moving on to the CNN layers (Algorithm 2), the Inisialisasi_Model_CNN function initializes a Convolutional Neural Network (CNN) model using the Sequential API. The model comprises two sets of convolutional and max-pooling layers, followed by a flattening layer to prepare the data for the fully connected layers. Finally, the model is compiled with the Adam optimizer, categorical cross-entropy loss function, and accuracy metric.

Based on the analysis carried out on Figure 5, it can be concluded the model performance has been evaluated and analyzed through the use of the Confusion Matrix. The Confusion Matrix which describes the model's classification performance for each specific Hiragana character class can be seen further in Table 1. It can be clearly seen the number of correct and incorrect predictions for each class, allowing researchers to evaluate the level of accuracy, precision, recall, and F1-Score. Based on Table 1, we can conclude that the CNN model shows excellent performance with the three optimizers used.

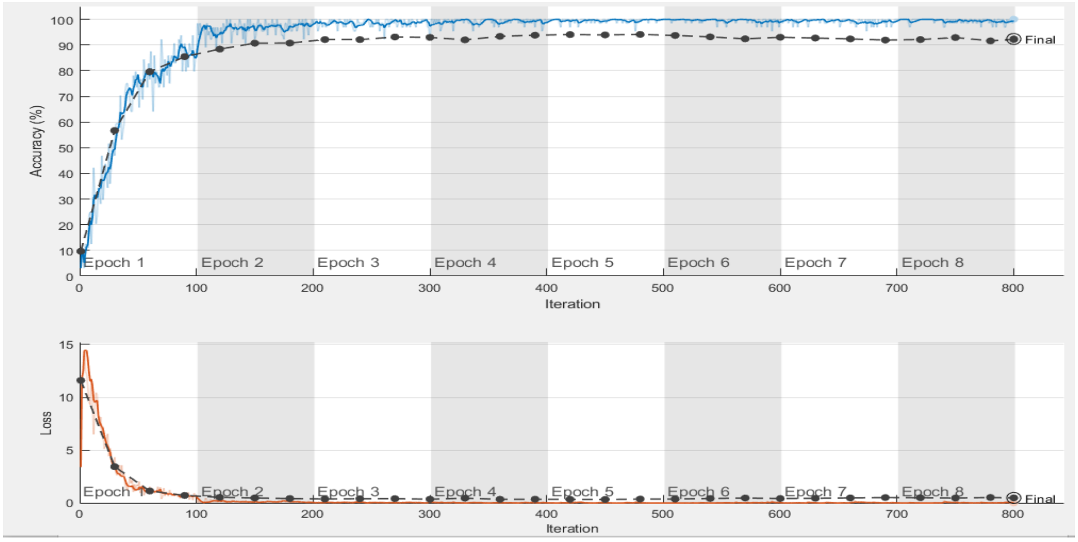


Figure 5. Training graph and graph loss

Based on table 1, the evaluation results indicate that Adam achieves the highest accuracy at 99.87%, with precision, recall, and F1-Score values of 98%, 100%, and 99%, respectively. Meanwhile, SGDM achieves an accuracy of 99.25% with identical precision and recall to Adam, albeit with a slightly lower F1-Score of 99%. The RMSP optimizer demonstrates an accuracy of 99.31%, with a precision of 99.25%, perfect recall, and the highest F1-Score of 99.70% among the three evaluated optimizers. After the internal evaluation process has been completed, the next step in this research is to compare the performance of the model developed with relevant previous research. The results can be seen in Table 2.

Table 1. The confusion matrix result

Optimizer	Accuracy	Precision	Recall	F1-Score
Adam	99.87%	98%	100%	99%
SGDM	99.25%	98%	100%	99%
RMSP	99.31%	99.25%	100%	99.70%

Table 2. The comparison result based on previous research

Previous Research	Method	Accuracy	Precision	Recall	F1-Score
[8]	Base CNN Adam Optimizer	95%	-	-	-
[29]	Base CNN Adam Optimizer	96.95%	-	-	-
[30]	Base CNN Adam Optimizer	86.5%	-	-	-
	CNN with Adam Optimizer	99.87%	98%	100%	99%
Proposed Method	CNN with SGDM Optimizer	99.25%	98%	100%	99%
	CNN with RMSP Optimizer	99.31%	99.25%	100%	99.70%

CONCLUSION

Model performance with various optimizers used, it can be seen that the CNN model achieves a very high level of accuracy with all three optimizers applied. With the Adam optimizer, the model managed to achieve an accuracy of 99.87%, with precision and recall values of 98% and 100% respectively, and an F1-Score reaching 99%. Meanwhile, by using the SGDM optimizer, the model recorded an accuracy of 99.25%, with equally optimal precision and recall values of 98% and 100%, and an F1-Score reaching 99%. No less impressive, the use of the RMSP optimizer also shows satisfactory results with 99.31% accuracy, 99.25% precision, 100% recall, and F1-Score reaching 99.70%. From the data shown, although all three optimizers show excellent performance, the RMSP optimizer stands out with the highest F1-Score, showing its effectiveness in classifying Hiragana characters with extraordinary precision. The overall results from Table 1 confirm that the developed models are highly competent in their classification tasks, with each optimizer offering its own advantages in handling the Hiragana character dataset. For future research and development in the field of Hiragana character recognition using Convolutional Neural Networks (CNNs), a promising area for exploration is the integration of transfer learning techniques. Transfer learning involves leveraging pre-trained CNN models on large-scale image datasets and fine-tuning them for specific tasks with smaller datasets, such as Hiragana character recognition. By adopting transfer learning, researchers can benefit from the knowledge and feature representations learned from vast image datasets, potentially enhancing the performance and efficiency of Hiragana classification models, especially in scenarios with limited training data. Furthermore, exploring transfer learning approaches tailored specifically to the nuances of Hiragana characters could lead to novel insights and advancements in the field. Additionally, the integration of natural language processing (NLP) technology holds great potential for enriching Hiragana character learning applications.

REFERENCES

[1] S. Y. Harunasari and N. Halim, "Gegar Bahasa pada Program Pertukaran Mahasiswa Indonesia di Jepang: Sebuah Studi Kasus," *Diglosia: Jurnal Kajian Bahasa, Sastra, dan Pengajarannya*, vol. 4, no. 4, pp. 401–412, Nov. 2021, doi: 10.30872/diglosia.v4i4.212.

[2] Charlie Tsai, "Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks," 2016.

[3] 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, doi: 10.1109/TPAMI.2007.1090.

[4] S. Das and S. Banerjee, "An Algorithm for Japanese Character Recognition," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 1, pp. 9–15, 2014, doi: 10.5815/ijigsp.2015.01.02.

[5] N. Amalia, E. Wahyu Hidayat, A. Putra Aldya, and U. Siliwangi Jl Siliwangi No, "PENGENALAN AKSARA SUNDA MENGGUNAKAN METODE JARINGAN SARAF TIRUAN BACKPROPAGATION DAN DETEKSI TEPI CANNY," 2020.

[6] R. Yohannes and M. E. Al Rivan, "Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM," *Jurnal Algoritme*, vol. 2, no. 2, pp. 133–144, Apr. 2022, doi: 10.35957/algoritme.v2i2.2363.

[7] N. E. W. Nugroho and A. Harjoko, "Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 221, Jul. 2021, doi: 10.22146/ijccs.66062.

[8] C. Umam, D. Andi Krismawan, and R. Raad Ali, "CNN for Image Identification of Hiragana Based on Pattern Recognition," 2021.

[9] N. R. D. Cahyo, C. A. Sari, E. H. Rachmawanto, C. Jatmoko, R. R. A. Al-Jawry, and M. A. Alkhafaji, "A Comparison of Multi Class Support Vector Machine vs Deep Convolutional Neural Network for Brain Tumor Classification," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2023, pp. 358–363. doi: 10.1109/iSemantic59612.2023.10295336.

[10] E. H. Rachmawanto and H. P. Hadi, "OPTIMASI EKSTRAKSI FITUR PADA KNN DALAM KLASIFIKASI PENYAKIT DAUN JAGUNG," vol. 22, no. 2, p. 2021.

[11] D. Fajri Riesaputri, C. Atika Sari, D. R. Ignatius Moses Setiadi, and E. Hari Rachmawanto, "Classification of Breast Cancer using PNN Classifier based on GLCM Feature Extraction and GMM Segmentation," in *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability, Scalability, and Security in the Age of Digital Disruption, iSemantic 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 83–87. doi: 10.1109/iSemantic50169.2020.9234207.

[12] M. M. Sardeshmukh, M. Chakkaravarthy, S. Shinde, and D. Chakkaravarthy, "Crop image classification using convolutional neural network," *Multidisciplinary Science Journal*, vol. 5, no. 4, 2023, doi: 10.31893/multiscience.2023039.

[13] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, Jan. 2022, doi: 10.1016/j.aiia.2021.12.002.

[14] Z. Azouz, B. Honarvar Shakibaei Asli, and M. Khan, "Evolution of Crack Analysis in Structures Using Image Processing Technique: A Review," *Electronics (Basel)*, vol. 12, no. 18, p. 3862, Sep. 2023, doi: 10.3390/electronics12183862.

[15] J. I. D. Prabandari, C. A. Sari, E. H. Rachmawanto, C. Irawan, B. R. Altamer, and M. A. Alkhafaji, "Early Detection and Classification of Melanoma Based on Android MobileNet V2 Convolutional Neural Network," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2023, pp. 400–405. doi: 10.1109/iSemantic59612.2023.10295362.

[16] A. Susanto, C. A. Sari, E. H. Rachmawanto, I. Utomo, W. Mulyono, and N. M. Yaacob, "A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19," *Scientific Journal of Informatics*, vol. 11, no. 1, 2024, doi: 10.15294/sji.v11i1.47305.

[17] M. A. Khayer, M. S. Hasan, and A. Sattar, "Arabian date classification using CNN algorithm with various pre-trained models," in *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, Institute of Electrical and Electronics Engineers Inc., Feb. 2021, pp. 1431–1436. doi: 10.1109/ICICV50876.2021.9388413.

[18] R. V. Kumar Reddy, B. Srinivasa Rao, and K. P. Raju, "Handwritten Hindi Digits Recognition Using Convolutional Neural Network with RMSprop Optimization," in *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018*, 2019, pp. 45–51. doi: 10.1109/ICCONS.2018.8662969.

[19] H. A. Morsy, "Developing a New CCN Technique for Arabic Handwritten Digits Recognition," *International Journal of Recent Technology and Engineering*, vol. 9, no. 3, pp. 520–524, 2020, doi: 10.35940/ijrte.c4588.099320.

[20] F. Ilham and N. Rochmawati, "Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN," *Journal of Informatics and Computer Science (JINACS)*, vol. 1, no. 04, pp. 200–208, Jul. 2020, doi: 10.26740/jinacs.v1n04.p200-208.

[21] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput Appl*, vol. 35, no. 23, pp. 17095–17112, Aug. 2023, doi: 10.1007/s00521-023-08568-z.

[22] H. K. Ravikiran, J. Jayanth, M. S. Sathisha, and K. Bindu, "Optimizing Sheep Breed Classification with Bat Algorithm-Tuned CNN Hyperparameters," *SN Comput Sci*, vol. 5, no. 2, Feb. 2024, doi: 10.1007/s42979-023-02544-z.

[23] D. Kilichev and W. Kim, "Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO," *Mathematics*, vol. 11, no. 17, Sep. 2023, doi: 10.3390/math11173724.

[24] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, "Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization," *Algorithms*, vol. 16, no. 6, Jun. 2023, doi: 10.3390/a16060308.

[25] S. C. Kim and Y. S. Cho, "Predictive System Implementation to Improve the Accuracy of Urine Self-Diagnosis with Smartphones: Application of a Confusion Matrix-Based Learning Model through RGB Semiquantitative Analysis," *Sensors*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/s22145445.

[26] C. R. Babu, D. S. Rao, M. .Seetha, and K. Prasad, "AccuracyAssessment of Supervised and Unsupervised Image Classification of Fused Satellite Images," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 1, pp. 169–174, 2017.

[27] M. S. Faridi, M. A. Zia, Z. Javed, I. Mumtaz, and S. Ali, "A Comparative Analysis Using Different Machine Learning: An Efficient Approach for Measuring Accuracy of Face Recognition," *Int J Mach Learn Comput*, vol. 11, no. 2, pp. 115–120, Mar. 2021, doi: 10.18178/ijmlc.2021.11.2.1023.

[28] A. Theissler, M. Thomas, M. Burch, and F. Gerschner, "ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices," *Knowl Based Syst*, vol. 247, Jul. 2022, doi: 10.1016/j.knosys.2022.108651.

[29] A. Chouza, R. Hernández, J. Jimenez, U. Orozco-Rosas, and K. Picos, "Handwritten hiragana classifier with minimal training data utilizing convolutional neural networks," in *Proc.SPIE*, Sep. 2019, p. 111360L. doi: 10.1117/12.2528046.

[30] E. H. Rachmawanto and P. N. Andono, “Deteksi Karakter Hiragana Menggunakan Metode Convolutional Neural Network,” *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 183–191, Dec. 2022, doi: 10.23887/janapati.v11i3.50144.