

Identification of Incung Characters (Kerinci) to Latin Characters Using Convolutional Neural Network

Tesa Ananda Putri¹, Tri Suratno^{*2}, Ulfa Khaira³

^{1,2,3}Department of Information Systems, FST Universitas Jambi, Jambi, Indonesia
e-mail: ¹tananda366@gmail.com, ^{*2}trie123@gmail.com, ³ulfa.ilkom@gmail.com

Abstrak

Aksara Incung adalah aksara warisan suku Kerinci yang terdapat di Kabupaten Kerinci, Provinsi Jambi. Pada 17 Oktober 2014 aksara Incung ditetapkan oleh Kemdikbud sebagai warisan tak benda yang dimiliki oleh Provinsi Jambi. Namun pada kenyataannya, aksara Incung tergolong hampir punah di masyarakat. Penelitian ini bertujuan untuk melakukan pengenalan karakter aksara Incung (Kerinci) dengan output berupa karakter latin dari aksara incung. Metode klasifikasi yang digunakan adalah metode Convolutional Neural Network (CNN). Dataset yang digunakan sebanyak 1400 citra karakter aksara Incung yang dibagi kedalam 28 kelas. Pada penelitian ini dilakukan percobaan untuk mendapatkan model yang paling optimal. Menunjukkan hasil dengan menggunakan metode CNN pada saat melakukan proses training mendapatkan akurasi data training mencapai 99% dan akurasi data testing mencapai 91% dengan menggunakan hyperparameter optimal dari pengujian yang telah dilakukan yaitu batch size 32, epoch 100, dan optimizer Adam. Pada saat evaluasi model CNN menggunakan 80 citra berupa kata (gabungan dari beberapa karakter) dengan 4 skenario pengujian, menunjukkan bahwa model mampu mengenali data citra hasil scanning buku cetak, data uji tulisan digital, data uji dengan citra berisi lebih dari 2 karakter, serta mampu mengenali citra dengan berbagai ukuran huruf yang berbeda.

Kata kunci—Aksara Incung, Convolutional Neural Network, Citra, Deep Learning, Pengenalan Pola.

Abstract

Incung script is a legacy of the Kerinci tribe located in Kerinci Regency, Jambi Province. On October 17, 2014, the Incung script was designated by the Ministry of Education and Culture as an intangible heritage property owned by Jambi Province. But in reality, the Incung script is almost extinct in society. This study aims to identify the characters of the Incung (Kerinci) script with the output in the form of Latin characters from the Incung script. The classification method used is the Convolutional Neural Network (CNN) method. The dataset used as many as 1400 incung character images divided into 28 classes. In this study, an experiment was conducted to obtain the most optimal model. Showing the results using the CNN method during the training process that the accuracy of the training data reaches 99% and the accuracy of the testing data reaches 91% by using the optimal hyperparameters from the tests that have been done, namely batch size 32, epoch 100, and Adam's optimizer. It evaluates the CNN model using 80 images in words (a combination of several characters) with 4 test scenarios. It shows that the model can recognize image data from scanning printed books, digital writing test data, test data with images containing more than two characters, and check images with different font sizes.

Keywords—Incung Script, Convolutional Neural Network, Deep Learning, Image, Pattern Recognition.

1. INTRODUCTION

The Incung script is a heritage script that is a form of diversity from the culture of the Kerinci civilization in Kerinci Regency, Jambi Province. In the past, used the Incung script to write literature, customary law, and spells written on containers in bark, cow horn, buffalo horn, bamboo, and palm leaves [1]. The Incung (Kerinci) script is the only script from central Sumatra whose existence needs to be preserved. On October 17, 2014, the Incung script was designated by the Ministry of Education and Culture (Ministry of Education and Culture) as an intangible heritage property owned by Jambi Province. However, in reality, the Incung script is almost extinct in society. It can seem that at least the people of Kerinci know and understand the Incung script as a cultural identity.

One of the steps to preserve the Incung script is through a technological approach. In this case, by introducing the incung character pattern by utilizing deep learning technology in image processing or image processing. Pattern recognition (Pattern Recognition) is one of the developments of artificial intelligence (Artificial Intelligence), which can recognize character patterns recognized by humans and then processed by computers [2].

The technology that currently has excellent capabilities in the case of object classification in images or pattern recognition today is deep learning technology [3], and it is at the heart of the current rise of artificial intelligence [4]. Deep Learning is machine learning that allows computers to learn from experience and understand commands based on given concepts [5]. The existence of deep learning has changed the paradigm of pattern recognition research, which previously separated feature extraction and classification methods that were carried out separately [6].

Character pattern recognition has become the focus of research with various purposes and different methods. Like the Javanese script research conducted by [7] using the Convolutional Neural Network (CNN) and Deep Neural Network (DNN) methods as a comparison with an accuracy rate of 70.22% for the CNN method and 64.65% for the DNN method. Then research by [8] used the CNN method and compared it with the MLP method, where CNN produced an accuracy of 89% while MLP only reached 62%. Research by [9] regarding handwritten characters of hiragana and katakana using the CNN method obtained an accuracy of 87.82%, while using a combination of CNN and SVM obtained an accuracy of 88.21%. Research by [10] by trying to recognize Javanese characters in the form of words using CNN resulted in 99.62% accuracy in the training process, and the segmentation process using a combination of projection profile and connected component labeling with 90% accuracy, and when testing using 20 new test data. The form of words (several various characters) obtained 80% accuracy.

From research on character pattern recognition that has been done before, the results of the average value of higher accuracy were obtained using the Convolutional Neural Network (CNN) method. The CNN method tries to imitate the image recognition system in the human visual cortex to process image information like humans [11][12]. In addition, the CNN method is a method found in Deep Learning which makes computer vision develop very rapidly [13]. This CNN capability is also stated as the best model to solve the problem of object recognition, pattern recognition [12], and face identification [14].

2. METHODS

2.1 Dataset

This study uses a dataset containing training and testing images. This dataset contains 28 classes of Incung (Kerinci) script, namely KA, GA, NGA, TA, DA, NA, PA, BA, MA, CA, GA, NYA, RA, LA, YA, WA, SA, HA, A, MBA, NGGA, NDA, NJA, MPA, NGKA, NTA, NCA, NGSA. The following are the Incung (Kerinci) characters shown in Figure 1.

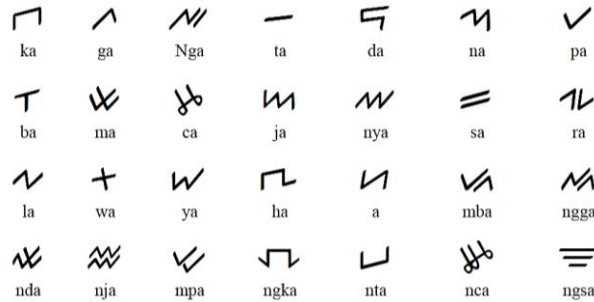


Figure 1 Incung script of the Kerinci tribe

The data collection process is carried out in two ways: Collecting Incung script data from books [15][16] and creating a digital image dataset of the Incung script with reference sources and validated by the Siginjei Jambi Museum and the Incung School Community.

The data obtained from the book will go through a scanning process or data scanning first to then be used as a digital image. The scan results from the data are manually cropped to separate the data into characters (one letter) to become data ready to be processed. Scanned 425 image data from books and 975 digital writing data from the data collection process, so the total dataset used was 1400 images.

2.2 Data Preprocessing

The data preprocessing stage is done by resetting the image size to 45 x 45 pixels and setting the color mode to grayscale, which reduces computation time because the image matrix is only processed in one channel [17]. After preprocessing the data, divide the 1400 data into training and testing data. The data sharing scenario used in this study was randomly divided into 80% or 1120 data for training data and 20% or 280 data for testing data [18]. The training data is larger than the testing data because the training data is the data used for model training. While the testing data is used to measure how well the model performs on the new data (data that does not exist in the training data).

2.3 Data Analysis Technique

The data analysis technique used in this study is the Convolutional Neural Network algorithm. Figure 2 below is a data analysis technique workflow.

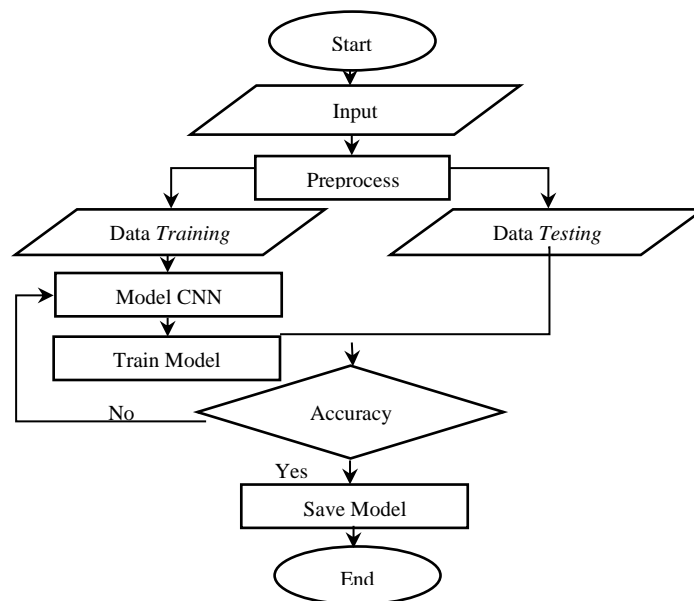


Figure 2 Flowchart of Data Analysis Techniques

Hyperparameters determine the overall structure of the network, but no standard approach to identify the optimal hyperparameters configuration [19]. For this reason, hyperparameter testing is carried out to compare the model by conducting a test scheme with different data scenarios to obtain the best accuracy results from the model. In this study, the hyperparameters that will use for the experiment are:

1. Batch size with 32, 64, and 128 data scenarios.
2. The epoch with data scenarios is 10 to 100.
3. The optimizers used are Adam, Adamax, and RMSProp.

2.4 Convolutional Neural Network

The structure of the CNN model is used in Table 1 below. The network consists of an input layer, three convolutional layers, three pooling layers, two fully connected layers, and an output layer.

Table 1 Convolutional Neural Network model design flowchart

Layer	Output	Parameter
Input	45x45x1	0
Conv2D_1	45x45x16	160
Max_pooling2D_1	22x22x16	0
Conv2D_2	22x22x96	13920
Max_pooling2D_2	11x11x96	0
Conv2D_3	11x11x128	110720
Max_pooling2D_3	5x5x128	0
Flatten	3200	0
Dense	480	1536480
Dropout	480	0
Dense	64	30784
Dropout	64	0
Dense output	28	1820
Total Params		1.693.884

The input layer uses an image that is 45x45 pixels in size. The input data is then processed in the first convolution layer using 2D convolution with 16 filters, the same padding or the same padding as the input data padding so that the input in the first convolution layer remains 45x45 [20], kernel with 3x3 matrix, activation using ReLu, and Max Pooling 2x2. The second layer uses 2D convolution with 96 filters and the third layer uses 2D convolution with 128 filters using the same settings as the first layer. The next layer is a flatten layer used to convert the dimensions of the feature map into a one-dimensional vector value [21] which is used to input a dense layer with a vector number of 480 and uses ReLu activation. The dense layer adds a fully connected layer at the end of the network [22], and the dropout value used is 0.18. It was followed by a second dense layer using vector 64, activation ReLu, and a dropout value of 0.06. The last layer is a dense layer with a vector value of 28 using softmax activation.

2.5 Design Of Image Testing

At the image testing stage, the model is tested with different data to determine whether the model has produced a good performance in classifying Incung script images. Figure 3 is the image testing process. This testing process is carried out in several stages. The first stage is to load the previously created model enter the image to test in the form of words (a combination of several characters) for further segmentation process on the input image or image cutting to separate the characters so that can detect the characters according to the CNN model. The last stage is the prediction process or the results of image classification, and then the prediction results will come out in the form of Latin characters from the incung script.

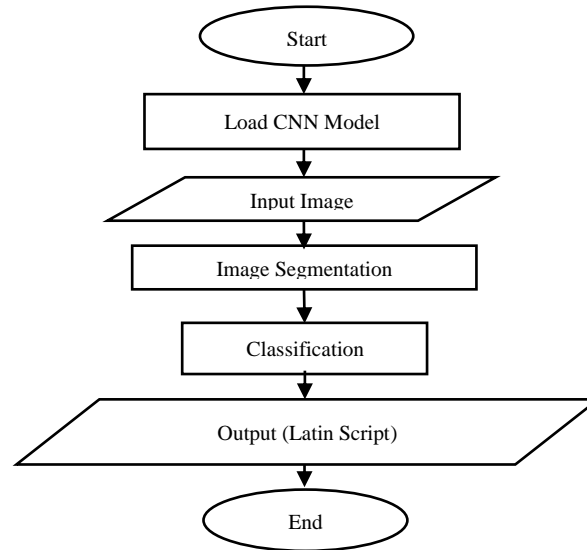


Figure 3 Flowchart of the image testing process.

3. RESULTS AND DISCUSSION

3.1 Test Results using Hyperparameter

Hyperparameter testing is done to compare which model produces the best level of accuracy. The hyperparameters that will be tested are the influence of Batch size, epoch, and Optimizer values. This test is crucial for using its magnitude in the model. The training process has different accuracy and loss results even though the resulting values are not significantly different. However, the training results for each hyperparameter have a reasonably significant difference when tested. The following is a table of hyperparameter test results:

Table 2 Test Result using Hyperparameter

Optimizer	Batch size	Epoch	Training accuracy	Testing accuracy
Adam	32	100	0,995	0,917
	64	100	0,984	0,892
	128	90	0,917	0,875
Adamax	32	100	0,747	0,774
	64	100	0,685	0,649
	128	100	0,575	0,589
RMSprop	32	100	0,994	0,871
	64	100	0,958	0,892
	128	100	0,871	0,835

Based on Table 2 shows a summary of the results of hyperparameter testing. The epoch test carried out with each epoch 10 to 100 showed the most optimal accuracy value at epoch 100. In experiment using adam's optimizer and batch size 128 obtained the highest accuracy at epoch 90. While, at epoch 100 the accuracy decreased.

Then the results from the accuracy value obtained showed that the most optimal optimizer was using the Adam optimizer with an accuracy value of batch size 32 was 0.995. batch size 64 is 0.984, and batch size 128 is 0.917. The batch size results show the most optimal value in batch size 32, which means that the smaller the batch size value, the higher the accuracy value obtained.

3.2 Training Result

After going through several processes in the Convolutional Neural Network (CNN) and testing hyperparameters to determine the best model, the results obtained are accurate in the

training process. The training process uses the results of the most optimal hyperparameter test, namely a learning rate of 0.0001 with an image input of 45x45 pixels and the number of Batch size 32, epoch 100, and using the Adam Optimizer. The following is the result of the training graph:

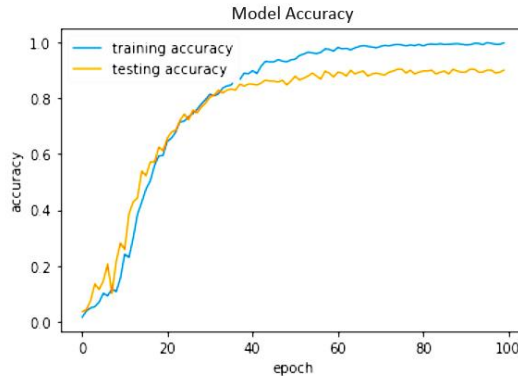


Figure 4 Graph of Training Results

Figure 4 shows the training results showing the model's accuracy. Model accuracy is the accuracy value obtained from the model training process, the higher the accuracy value got, the better the model that has been made. In the graph, there are two indicators, namely training accuracy, which is indicated by a blue graph line and testing accuracy, which is characterized by a yellow graph line.

The results of the accuracy of the train data shown by the blue graph show the accuracy value at epoch 10 which is 0.1848, at epoch 20 the graph increases significantly with an accuracy of 0.6598, then at epoch 30 the graph shows an accuracy value of 0.80, at epoch 40 with an accuracy of 0.9048, and accuracy at epoch 50 which is 0.9482. For the next epoch value, the accuracy value is not significantly different. Each accuracy value, namely 60 epochs is 0.9795, epoch 70 is 0.9839, epoch 80 is 0.9893, epoch 90 is 0.9929, and epoch 100 shows the highest graph, namely 0.9946. So, the results of accuracy training using 1120 data trains are 0.9955 or 99.55% accuracy data.

Meanwhile, the yellow graph shows the data testing results, namely the data used to measure how well the model performs on the new data. The graph shows the results of the data testing accuracy, reaching 0.9179 or 91% using 280 testing data.

3.3 Confusion Matrix

The testing data results will then be summarised in a confusion matrix shown in the following figure.

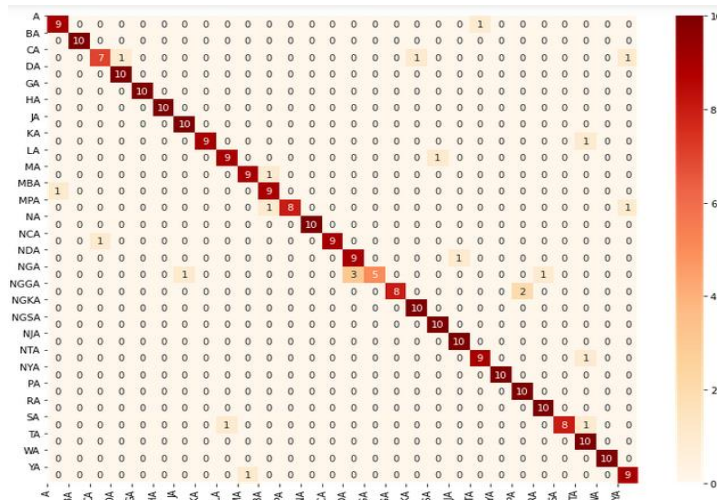


Figure 5 Confusion Matrix

Based on Figure 5, the classification results from the model to the testing data show good results. Of the 280 testing data used, the model can classify correctly as many as 257 data. The calculation of the accuracy of the matrix is:

$$Accuracy = \frac{TP}{Total\ test\ data} \times 100\%$$

$$Accuracy = \frac{257}{280} \times 100\% = 91.78\%$$

According to the accuracy graph during the training model, the accuracy value of the data test is 91.78%. The precision, recall, and f1-score values are calculated in Table 3 below based on the confusion matrix.

Table 3 The results of the classification report value accuracy, precision, recall, and f1-score.

No	Script	precision	Recall	F1-score
1	A	0.90	0.90	0.90
2	BA	1.00	1.00	1.00
3	CA	0.88	0.70	0.78
4	DA	0.91	1.00	0.95
5	GA	1.00	1.00	1.00
6	HA	1.00	1.00	1.00
7	JA	0.91	1.00	0.95
8	KA	1.00	0.90	0.95
9	LA	0.90	0.90	0.90
10	MA	0.90	0.90	0.90
11	MBA	0.82	0.90	0.86
12	MPA	1.00	0.80	0.89
13	NA	1.00	1.00	1.00
14	NCA	1.00	0.90	0.95
15	NDA	0.75	0.90	0.82
16	NGA	1.00	0.50	0.67
17	NGGA	1.00	0.80	0.89
18	NGKA	0.91	1.00	0.95
19	NGSA	0.91	1.00	0.95
20	NJA	0.91	1.00	0.95
21	NTA	0.90	0.90	0.90
22	NYA	1.00	1.00	1.00
23	PA	0.83	1.00	0.91
24	RA	0.91	1.00	0.95
25	SA	1.00	0.80	0.89
26	TA	0.77	1.00	0.87
27	WA	1.00	1.00	1.00
28	YA	0.82	0.90	0.86
Accuracy		0.92		
Macro avg		0.93	0.92	0.92
Weighted avg		0.93	0.92	0.92

Based on Table 3, the accuracy of the precision value or the level of accuracy between the requested data and the results provided by the model is 0.93. The recall value, the model's success rate in retrieving information, is 0.92, and the f1-score value, a single parameter measure of success retrieval that combines recall and precision, is 0.92.

3.4 Image Test

In model testing, the first step is to load the previously saved training model. Then input the image data to be tested in the form of an Incung script image read "NAMA" as shown in Figure 6.



Figure 6 New test data

Figure 6 is the test data input which will then enter into the segmentation process or character separation using a vertical projection profile. The most necessary thing in the projection profile method is to get a histogram of the image shown in Figure 7 below.

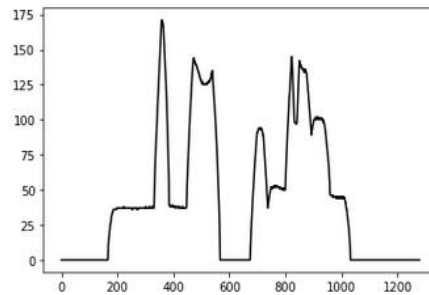


Figure 7 Histogram of Figure 6

This segmentation process will detect the white color of each column. If white is seen, it will be determined as the initial limit for cutting characters, if black is seen, it will be defined as the end limit for cutting characters. The results of character segmentation using a vertical projection profile are shown in Figure 8 which separates the characters in order from left to right:

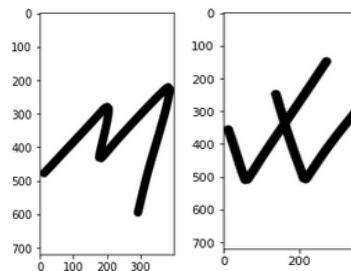


Figure 8 Segmentation Results using Projection Profile

After successfully separating the characters, as shown in Figure 8, the next step is to classify the labels using CNN. The test data produces an array ['NA', 'MA']. The array classification process does not go through the cropping or character separation process when the tested image has only one character. The display of the test results can be seen in Figure 9.

['NA', 'MA']

Figure 9 Prediction results with CNN

3.5 Graphic User Interface

Graphic User Interface (GUI) facilitates testing images in incung script recognition. The GUI was created in a web form using the Django framework in this research. The system that has been built has three pages, namely the home page that contains a description of the system, the Incung Script page that includes the appearance of images of Incung scripts and examples of the use of Incung script in word form, and the Incung script recognition page which is the core page that is to test previously created models using new Incung script image data.

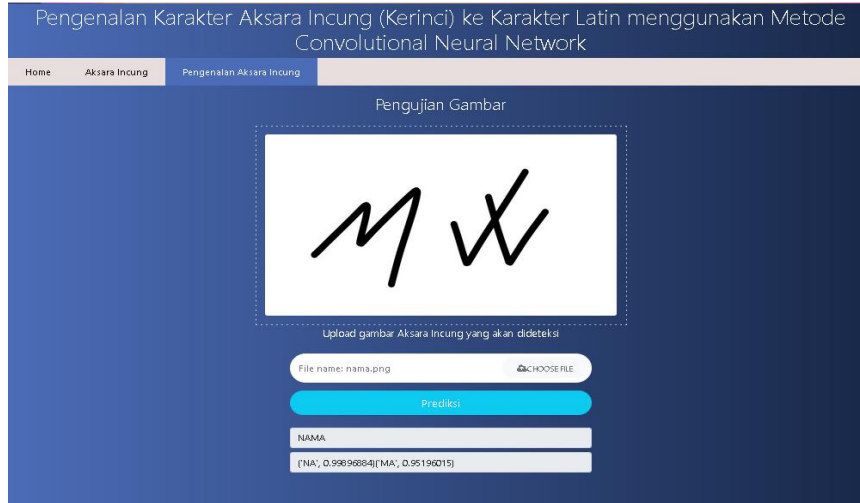


Figure 10 Test results with GUI

Figure 10 shows the results of the Incung script prediction with the results in the form of Latin characters from the Incung script and the confidence score of the predicted script. The prediction results show the results of the Latin character 'NAMA' with a confidence score of NA of 0.9989 and MA of 0.9519.

3.6 Evaluation Model

After the CNN modeling process and obtaining the best accuracy value, the model is tested to validate various scenarios. It will try the evaluation of this model with four different scenarios. The test is carried out using test data of 80 different images, where each test scenario uses 20 data. The purpose of testing is to determine the level of effectiveness and accuracy of this study using new data.

Table 4 Model Evaluation Results

No	Testing Scenario	Total Data	True	False	Accuracy
1	Image Data from Scanning Book	20	20	0	100%
2	Image Data from digital writing	20	20	0	100%
3	Image data more than two characters	20	20	0	100%
4	Image Data Different Font Size	20	20	0	100%

3.7 Test Sample

Table 1 Image Data from Scanning Book

No	Test Data	Word	Predict	Character	Score	Description
1		Hamba	Hamba	Ha	1	True
				Mba	0.99	
2		Nanda	Nanda	Na	1	True
				Nda	0.99	
3		Tangga	Tangga	Ta	1	True
				Ngga	0.85	
4		Panda	Panda	Pa	1	True
				Nda	0.99	
5		Ganda	Ganda	Ga	0.98	True
				Nda	0.78	

Table 5 is a sample test on the image data of the book's scanning results. Did this test to see how accurate incung character pattern recognition is in the image of the scanning results.

Table 6 Image Data from digital writing

No	Test Data	Word	Predict	Character	Score	Description
1		Baba	Baba	Ba	0.99	True
				Ba	1	
2		Kata	Kata	Ka	1	True
				Ta	0.99	
3		Mara	Mara	Ma	0.96	True
				Ra	0.99	
4		Nada	Nada	Na	1	True
				Da	1	
5		Lama	Lama	La	0.99	True
				Ma	1	

Table 6 is a sample test on digital writing image data. Did this test to see how accurate incung character pattern recognition is in digital writing images.

Table 7 Image data more than two characters

No	Test Data	Word	Predict	Character	Score	Description
1		Antara	Antara	A	1	True
				Nta	1	
				Ra	0.99	
2		Dahaga	Dahaga	Da	1	True
				Ha	1	
				Ga	0.96	
3		Hambasaha ya	Hambasaha ya	Ha	1	True
				Mba	0.99	
				Sa	0.99	
				Ha	1	
				Ya	1	
4		Samarata	Samarata	Sa	1	True
				Ma	1	
				Ra	0.99	
				Ta	1	
5		Ramayana	Ramayana	Ra	1	True
				Ma	1	
				Ya	1	
				Na	1	

Table 7 is a sample test on image data over two characters. Did this test to see if the system can break down characters and classify more than two incung characters.

Table 8 Image Data Different Font Size



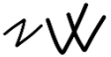


No	Test Data	Word	Predict	Character	Score	Description
1		Damba	Damba	Da	0.99	True
				Mba	0.99	
2		Rajajawa	Rajajawa	Ra	1	True
				Ja	0.78	
				Ja	1	
				Wa	1	
3		Lama	Lama	La	0.99	True
				Ma	1	
4		Sahaja	Sahaja	Sa	1	True
				Ha	1	
				Ja	0.99	
5		Panda	Panda	Pa	1	True
				Nda	0.81	

Table 8 is a sample test on the image data of the book's scanning results. Did this test to see if character size affects the segmentation and classification of Incung script characters.

4. CONCLUSIONS

Based on research that has been done has succeeded in making a CNN model that will be used for incung script image testing. Hyperparameter testing mechanisms are performed to obtain optimal CNN models by testing on batch size, epoch, and optimizer using multiple data scenarios with trial and error methods. This research got the best accuracy rate results in batch size is 32, epoch 100, and Adam optimizer obtained an accuracy rate of 99% in data training and 91% in data testing.

Evaluation of the model made using test data in the form of words (combined characters) incung script obtained good results in the introduction of the Incung script. This model can recognize image data from scanning printed books, digital writing test data, test data with imagery containing more than two characters, and recognize images with various letter sizes. It shows that the model that has been created by applying the Convolutional Neural Network method can perform incung script recognition well.

REFERENCES

- [1] H. Mubarat, "Aksara Incung Kerinci Sebagai Sumber Ide Penciptaan Seni Kriya," *Ekspresi Seni*, vol. 17, no. 2, 2015, doi: 10.26887/ekse.v17i2.101.
- [2] P. Pudil, P. Somol, and M. Haindl, "Introduction To Statistical Pattern Recognition.," *Proc Natl Electron Conf*, vol. 29, pp. 349–352, 1974, doi: 10.1016/0098-3004(96)00017-9.
- [3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [4] N. Akhtar and A. Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018, doi: 10.1109/ACCESS.2018.2807385.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, USA: The MIT Press, 2016.

- [6] J. Ruiz-del-Solar, P. Loncomilla, and N. Soto, "A Survey on Deep Learning Methods for Robot Vision," Mar. 2018, Accessed: Nov. 22, 2021. [Online]. Available: <http://arxiv.org/abs/1803.10862>.
- [7] R. Khadijah and A. Nurhadiyatna, "Deep Learning for Handwritten Javanese Character Recognition," *Int. Conf. Informatics Comput. Sci.*, pp. 59–64, 2017.
- [8] C. K. Dewa, A. L. Fadhilah, and Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 1, pp. 83–94, 2018, doi: 10.22146/ijccs.31144.
- [9] N. Euclides, W. Nugroho, and A. Harjoko, "Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 3, 2021.
- [10] F. Ilham and N. Rochmawati, "Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN," *JINACS (Journal Informatics Comput. Sci.*, vol. 01, no. 4, pp. 200–208, 2020.
- [11] P. Grace W. Lindsay, "Convolutional neural networks as a model of the visual system: Past, present, and future," *J. Cogn. Neurosci.*, vol. 33, no. 10, pp. 2017–2031, 2021, doi: 10.1162/jocn_a_01544.
- [12] X. Mao, S. Hijazi, R. Casas, P. Kaul, R. Kumar, and C. Rowen, "Hierarchical CNN for traffic sign recognition," *IEEE Intell. Veh. Symp.*, no. 4, pp. 130–135, 2016, doi: 10.1109/IVS.2016.7535376.
- [13] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.
- [14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1701–1708, 2014, doi: 10.1109/CVPR.2014.220.
- [15] B. V. Jauhari, J. Mardizal, Zulwachdi, and Yozerizal, *Mengenal Aksara Incung Suku Kerinci Daerah Jambi*. Sungai Penuh, Provinsi Jambi: Lembaga Bina Potensia Aditya Mahatva Yodha, 2013.
- [16] B. V. Jauhari and Martono, *Belajar Aksara Incung Suku Kerinci Daerah Jambi*. Sungai Penuh, Provinsi Jambi: Lembaga Bina Potensia, 2013.
- [17] F. Utaminingrum, A. W. Satria Bahari Johan, I. K. Somawirata, Risnandar, and A. Septiarini, "Descending stairs and floors classification as control reference in autonomous smart wheelchair," *J. King Saud Univ. - Comput. Inf. Sci.*, 2021, doi: 10.1016/j.jksuci.2021.07.025.
- [18] Z. Ahmad and N. Khan, "CNN-Based Multistage Gated Average Fusion (MGAF) for Human Action Recognition Using Depth and Inertial Sensors," *IEEE Sens. J.*, vol. 21, no. 3, pp. 3623–3634, 2021, doi: 10.1109/JSEN.2020.3028561.
- [19] P. L. Neary, "Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning," *IEEE Int. Conf. Cogn. Comput.*, pp. 73–77, 2018, doi: 10.1109/ICCC.2018.00017.
- [20] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," pp. 1–31, 2016, [Online]. Available: <http://arxiv.org/abs/1603.07285>.
- [21] W. Ng *et al.*, "Convolutional neural network for simultaneous prediction of several soil properties using visible/near-infrared, mid-infrared, and their combined spectra," *Geoderma*, vol. 352, pp. 251–267, Oct. 2019, doi: 10.1016/J.GEODERMA.2019.06.016.
- [22] A. Ramdan, V. Zilvan, E. Suryawati, H. F. Pardede, and V. P. Rahadi, "Tea clone classification using deep CNN with residual and densely connections," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 4, pp. 289–296, 2020, doi: 10.14710/jtsiskom.2020.13768.