Lampung Script Recognition Using Convolutional Neural Network

Panji Bintoro*[1], Agus Harjoko[2]
[1]Master Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2] Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: *[1]**panjibintoro09@mail.ugm.ac.id**, [2]aharjoko@ugm.ac.id

Abstrak
Aksara Lampung sering digunakan dalam menuliskan kata dalam bahasa Lampung. Bahasa Lampung sendiri digunakan oleh orang Lampung asli maupun orang yang belajar bahasa Lampung. Aksara Lampung sulit dipelajari karena banyak kombinasi induk aksara dan anak huruf. CNN merupakan metode di bidang pengenalan objek yang memiliki lapisan khusus, yaitu lapisan konvolusi dan lapisan pooling yang memungkinkan proses pembelajaran fitur dengan baik. Pengenalan tulisan tangan seperti pada pengenalan karakter pada MNIST, CNN menghasikan performa yang lebih baik dibandingkan dengan metode lain. Dari keunggulan CNN tersebut, maka dipilih metode CNN dengan arsitektur DenseNet sebagai arsitektur terbaik untuk mengenali setiap aksara Lampung. Pada penelitian ini terdapat 2 proses utama yaitu preprocessing, dan pengenalan. Penelitian ini berhasil menerapkan metode CNN yang dapat mengenali aksara Lampung. Dataset dibagi menjadi 4 jenis aksara yang memiliki bunyi yang berbeda. Pertama, data induk aksara mendapatkan akurasi 98%. Kedua, data induk aksara dengan anak huruf di atas mendapatkan akurasi 98%. Ketiga, data induk aksara dengan anak huruf di samping mendapatkan akurasi 98%. Keempat, data induk aksara dengan anak huruf di bawah mendapatkan akurasi 97%.

Kata kunci*— Aksara Lampung, Pengenalan Pola, Deep Learning, Convolutional Neural Network, Densely Connected Convolutional Network*

Abstract
The Lampung script is often used in writing words in Lampung language. The Lampung language itself is used by native Lampung people and people who learn Lampung language. The Lampung script is difficult to learn because there are many combinations of parent characters and subletters. CNN is a method in the field of object recognition that has a specific layer, namely a convolution layer and a pooling layer that allows the feature learning process well. Handwriting recognition as in character recognition in MNIST, CNN produces better performance compared to other methods. From the advantages of CNN, the CNN method with DenseNet architecture was chosen as the best architecture to recognize each Lampung script. In this study, there are 2 main processes, namely preprocessing, and recognition. This study succeeded in applying the CNN method which can recognize Lampung script. The dataset is divided into 4 groups of characters that have different sounds. First, the parent character data get 98% accuracy. Second, the parent letter data with the above letters get 98% accuracy. Third, the parent character data with the sub-letters on the side get 98% accuracy. Fourth, the parent letter data with the lower letters get 97% accuracy.

Keywords*— Lampung Script, Pattern Recognition, Deep Learning, Convolutional Neural Network, Densely Connected Convolutional Network*

1. INTRODUCTION

Each region has a culture that is characteristic of each region. The script is one of the cultural values that is maintained and preserved. The Lampung tribe has a traditional script called Lampung script or it can also be called Kagangapa. Aksara serves to write documents in Lampung language. The Lampung script consists of 20 main letters and 12 sub-letters [1]. Figure 1 shows the parent letter, and Figure 2 shows the subletter.
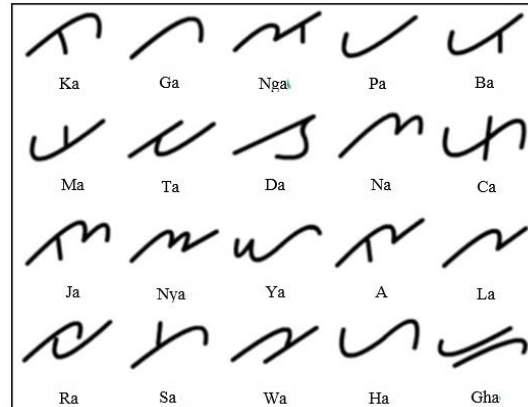

Figure 1. Parent character

Figure 2. Sub-letter

The Lampung script currently needs attention because it is rarely used and the scope of use is less. As a result, someone has difficulty recognizing when looking at Lampung script [1]. Each region has a culture that is characteristic of each region. The script is one of the cultural values that is maintained and preserved. The script is a system of visual symbols printed on paper or other media to express expressive elements in a language. The visual symbol in question is how to write sounds in a language into letter symbols.

There have been several studies on character recognition such as Lampung script, Javanese script, and Balinese script. Most of the research uses machine learning techniques such as convolutional neural networks to perform classification tasks with some feature extraction techniques. Fitriawan and Setiawan [1] used backpropagation to recognize handwritten Lampung scripts. The research resulted in an accuracy rate of 80% for all characters. The application of the backpropagation algorithm to the Lontara Bugis-Makassar letter pattern recognition has been carried out by Alwi and Wardoyo [2]. Aims to apply backpropagation neural network to recognize Bugis-Makassar script. The research resulted in a system that can create several network architectures that are stored in a profile. The network architecture obtained from the experimental results to recognize the Bugis-Makassar script achieved an accuracy of up to 95.2%. The recognize of the Lontara script has also been carried out by Sudarman and Hartati [3]. Aims to apply the CNN method to recognize syllables in the image of the Lontara script with handwritten data from the Lontara script. This research applies the dropout technique to the CNN architectural model at the fully connected layer. The study succeeded in increasing the average accuracy to 99.25% with a dropout value of 0.9 in the 10th fold. Dewa et. al [4] compared CNN and MLP with the results of the CNN model getting better

accuracy than MLP accuracy for handwriting recognition of Javanese characters for all folds. However, the CNN model takes longer to train than the MLP model. From the experiment, it can be seen that the accuracy of the CNN model for handwritten Javanese script cannot reach 90% for all folds. The recognize of Balinese handwriting by applying CNN was also carried out by Arya and Harjoko [5]. The test was carried out using the hyperparameter configuration selection, namely learning speed 0.001. The number of filters in the convolution layers 1 and 2 are 15 and 24. The number of neurons in the FC layer as many as 190 shows an accuracy of only 91.85%. After applying the dropout technique, the accuracy increased to 95.74% with a dropout value of 0.5. The transliteration of Javanese characters by Widiarti et. al [6] by looking for an accurate recognition value. In Javanese handwriting by applying several models in the preprocessing process and feature extraction. The transliteration system on the image of the manuscript document catalog numbered SB.141 with a 95% confidence level. The success rate of manuscript segmentation is between 85.9% to 94.82%. The success rate of transliteration of Javanese script images is between 73.51% to 85.69%, and the success rate of syllable merging is between 69.20% to 87.29%.

In this study, we propose the CNN method using the DenseNet architecture. DenseNet architecture can improve the flow of information and gradients across the network. This results in much better accuracy. The CNN method with DenseNet architecture is feasible to use for image recognition of Lampung script on printed and handwritten documents.

2. METHODS

In this section, the data and methods used in this study will be explained.

2. 1 Pengumpulan Data

Lampung script image data collection is very much needed. Lampung script image data was obtained from printed documents [7]. Lampung script image data that has been collected is then divided into 4 types of data. The first data is an image of the main character. The second data is in the form of a combination of parent letters with the above children. The third data is in the form of a combination of parent letters with the letters on the side. The fourth data is in the form of a combination of the parent letters with the lower letters. Figure 3 shows the process of collecting Lampung script data.
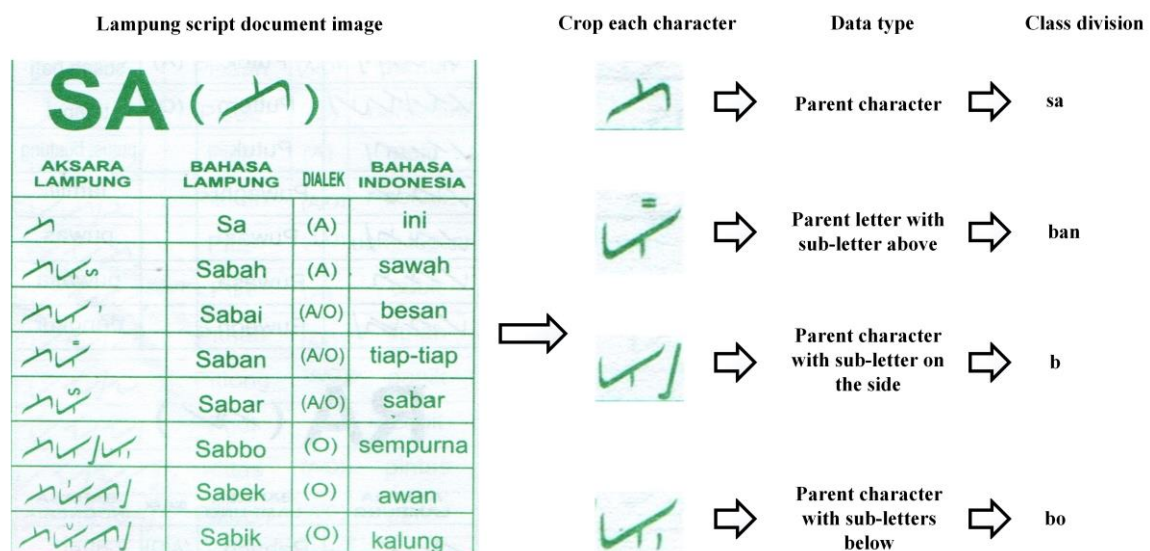

Figure 3. Data collection process

The results of the data collection process in the form of images of Lampung script that have not been divided according to the type of data. Furthermore, the image of Lampung script

is divided according to the type of data. The parent character data obtained 520 parent character images which were divided into 20 classes. The parent character data with the above letters get 2496 images which are divided into 96 classes. The parent character data with the sub-letters on the side get 1222 which is divided into 47 classes. The parental data with the letters below get 1482 which is divided into 57 classes.

2. 2 Preprocessing

Preprocessing is a process carried out on data with the aim that the data is ready to be used as input. At this stage the preprocessing is carried out in 4 stages, namely cropping, resizing, noise removal, and data augmentation. In the introduction, preprocessing only performs the cropping stage. In this study, the preprocessing carried out is as follows.

2. 2. 1 Cropping

Data obtained from documents and handwriting must go through a cropping process. Figure 3 on the left shows the results of cropping done manually. The results of cropping manually still have outer space. So cropping should be done automatically to eliminate outer space. Figure 4 on the right shows the results of cropping that is done automatically.
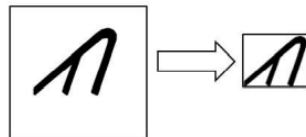

Figure 4. The result of cropping characters ka

2. 2. 2 Resize

Data that has gone through the cropping process still has various sizes. Resize is necessary so that the data has the same size. In this study, the data was resized to 256 x 256 pixels for the document image and 200 x 200 pixels for Lampung script handwritten image. Figure 5 shows the resizing results.
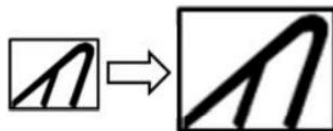

Figure 5. The result of resizing character ka

2. 2. 3 Noise Removal

The resized data is then carried out by the noise removal process. This process aims to remove small spots that are considered noise in the image. Next, we invert the image. Inverting is the process of changing colors in the image. In this study, the white background was replaced with black and black characters were replaced with white. Figure 6 shows the results of inverting.
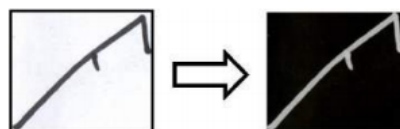

Figure 6. The result of inverting the character ka

2. 2. 4 Data Augmentation

The next preprocessing is data augmentation, this is done because the data obtained during the data collection process is very limited. First, we rescale by dividing the RGB values from 0-255 by 255, so we get RGB values in the range 0-1. Second, we do a shear range by tilting the image by 0.2 degrees. Third, we zoom the range by enlarging the image by 0.2. Fourth, we do a rotation range by rotating the image by 15 degrees. Fifth, we perform a horizontal flip by flipping all rows and columns of image pixels horizontally. Figure 7, part (a) shows the original image of the character ka, part (b) shows the image of the character ka that has been rotated.
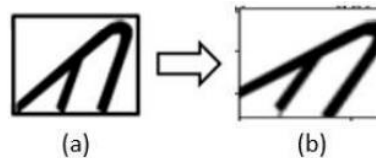

Figure 7. Example of the result of the rotation image of the character ka

2. 3 Convolutional Neural Network

In this study, we chose 2 CNN architectures, namely CNN Normal and CNN DenseNet. The architecture was carried out in several variations during training and testing. From these two architectures, we are looking for the optimal architecture for Lampung script recognition.

2. 3. 1 CNN Normal

In this study, the details of the best normal CNN model are shown in Table 1. Normal CNN architecture uses the Alexnet architecture extraction layer [8]. This architecture consists of 5 stages of convolution operations, 2 batch normalizations, 4 stages of pooling operations, 1 flatten layer, 2 hidden layers with ReLU and dropout activation functions, and a softmax output layer.

Table 1. CNN Normal best model

Layers	CNN Normal	Output Size	Param#
Input layer	256 x 256 x 3	256 x 256 x 3	0
Convolutional layer	11 x 11 x 128	123 x 123 x 128	46592
Batch normalization	-	123 x 123 x 128	512
Pooling layer	3 x 3 max pooling	61 x 61 x 128	0
Convolutional layer	5 x 5 x 128	61 x 61 x 128	409728
Batch normalization	-	61 x 61 x 128	512
Pooling layer	3 x 3 max pooling	30 x 30 x 128	0
Convolutional layer	3 x 3 x 256	30 x 30 x 256	295168
Convolutional layer	3 x 3 x 384	30 x 30 x 384	885120
Pooling layer	3 x 3 max pooling	14 x 14 x 384	0
Convolutional layer	3 x 3 x 256	14 x 14 x 256	884992
Pooling layer	3 x 3 max pooling	6 x 6 x 256	0
Flatten layer	-	9216	0
Hidden layer	1024, ReLU	1024	9438208
	batch normalization	1024	4096
	dropout 0,2	1024	0
Hidden layer	512, ReLU	512	524800
	batch normalization	512	2048
	dropout 0,2	512	0
Output layer	num class, softmax	20	10260

The Normal CNN architecture above consists of 2 parts, namely the extraction layer and the classification layer. The input layer uses a 256 x 256 x 3 matrix. The input size is used based on the size of the preprocessed training image data. The number of channels in the input layer is 3. The first extraction layer is a convolution layer with a size of 11 x 11 x 128, 5 x 5 x 128, 3 x 3 x 256, 3 x 3 x 284, and 3 x 3 x 256. The input matrix will go through a multiplication process with the kernel matrix. Processes at this layer will be activated with the ReLU function. This function changes all negative values in the matrix to 0 [9]. The second extraction layer is the pooling layer in this study using the max pooling layer. Max pooling aims to calculate the maximum value for each path from the feature map [10]. The third extraction layer is the flatten layer which aims to reshape the features into a vector. The vector is used as input to the fully connected layer. After processing the convolution layer and pooling layer, it produces a vector measuring 1 x 4096. The vector results are used as input for the classification layer.

The classification layer uses a fully connected layer based on backpropagation. The vector from the extraction layer becomes the input to train the weights in the classification layer. In this study, the classification layer consists of 2 hidden layers. The number of nodes in the hidden layer is 1024 and 512. In the activation function using ReLU. We also use dropout with size 0.2. The next classification layer is the output layer with the softmax activation function.

2. 3. 2 CNN DenseNet

In this study, the details of the best CNN DenseNet model are shown in Table 2 and Table 3. CNN DenseNet architecture uses the extraction layer DenseNet architecture [11]. This architecture consists of 1 convolution operation stage, 1 pooling operation stage, 4 dense block operation stages, 3 transition layer operation stages, 1 hidden layer with ReLU and dropout activation functions, and a softmax output layer.

Table 2. CNN DenseNet's best models

Layers	CNN DenseNet		Output Size	Param#
Input layer	256 x 256 x 3		256 x 256 x 3	0
Convolution layer	7 x 7 conv, stride 2		128 x 128 x 64	9408
Pooling layer	3 x 3 max pool, stride 2		64 x 64 x 64	0
Dense Block 1	1 x 1 conv	x 6	64 x 64 x 128	8192
	3 x 3 conv		64 x 64 x 32	36864
Transition layer	1 x 1 conv		64 x 64 x 128	32768
	2 x 2 average pool, stride 2		32 x 32 x 128	0
Dense Block 2	1 x 1 conv	x 12	32 x 32 x 128	16384
	3 x 3 conv		32 x 32 x 32	36864
Transition layer 2	1 x 1 conv		64 x 64 x 128	32768
	2 x 2 average pool, stride 2		32 x 32 x 128	0
Dense Block 3	1 x 1 conv	x 24	16 x 16 x 128	32768
	3 x 3 conv		16 x 16 x 32	36864
Transition layer 3	1 x 1 conv		16 x 16 x 512	524288
	2 x 2 average pool, stride 2		8 x 8 x 512	0
Dense Block 4	1 x 1 conv	x 16	8 x 8 x 128	65536
	3 x 3 conv		8 x 8 x 32	36864

Table 3. Best model of CNN DenseNet continued

Layers	CNN DenseNet	Output Size	Param#
Klasifikasi layer	7 x 7 global average pool	1024	0
	Batch normalization	1024	4096
	Dropout 0.2	1024	0
	Hidden layer 1024 node, ReLU	1024	1049600
	Batch normalization	1024	4096
	Dropout 0.5	1024	0
	num class, softmax	20	20500

The CNN DenseNet architecture above consists of 2 parts, namely the extraction layer and the classification layer. The input layer uses a 256 x 256 x 3 matrix. The input size is used based on the size of the preprocessed training image data. The number of channels in the input layer is 3. The first extraction layer is the convolution layer with a size of 7 x 7 with stride 2. The second extraction layer used in this study is the max pooling layer. The max pooling operation is used to calculate the maximum value for each path from the feature map [10]. In this study, there is 1 layer max pooling used and has a size of 3 x 3 with stride 2.

The third extraction layer used in this study is a dense block layer. A process on dense blocks using bottleneck layers. The first process uses convolution with a size of 1 x 1 which has 4*k where k is the growth rate. The second process is to use a convolution of size 3 x 3 which is owned by k. In this study, the number of dense blocks used is 4. In the first dense block, the process was carried out 6 times. The second Dense block process was carried out 12 times. The third Dense block process was carried out 24 times. The fourth Dense block process was carried out 16 times.

The fourth extraction layer used in this study is the transition layer. There are two processes in the transition layer. The first process uses convolution with a size of 1 x 1. Before the convolution process must pass the batch normalization function, and ReLU. The next process is average pooling with a size of 2 x 2 with stride 2. This average pooling aims to calculate the average value of each path from the feature map [10]. In this study, all transition layers use the same size and there are 3 transition layers.

This classification layer uses a fully connected layer based on backpropagation. The vector from the extraction layer becomes the input to train the weights in the classification layer. In this study, the classification layer consists of several processes. The first process is global average pooling which aims to unify the global average of the feature map. The classification layer consists of 1 hidden layer. The number of nodes in the hidden layer is 1024. In the activation function using ReLU. We also use batch normalization and dropout sizes 0.2 and 0.5. Kernel weight initialization is done using he uniform. The weights will go through the multiplication process with the kernel matrix that has been initialized. The next classification layer is the output layer with the softmax activation function.

2. 4 Model Training and Testing

At the training and testing stage of the CNN model, Lampung script data was divided into 80% for training data, 10% for validation data, and 10% for test data. This stage uses the stochastic gradient descent optimizer with learning speed 0.1, decay 0.000001, momentum 0.9, and loss using categorial cross entropy. After training and testing, the next stage is the evaluation stage. CNN model performance is measured by using the values of accuracy, precision, recall, and f1-score.

Formula (1).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * \mathbf{100}\%$$

(1)

Formula (2).

$$Presisi = \frac{TP}{FP+TP} * 100\%$$

(2)

Formula (3).

$$Recall = \frac{TP}{FN+TP} * 100\%$$

(3)

Formula (4).

$$F1\text{-}score = 2 * \frac{Recall * Presisi}{Recall + Presisi}$$

(4)

where TP is True Positive, namely the number of positive data that are classified correctly by the system. TN is True Negative, namely the number of negative data that are classified correctly by the system. FP is False Positive, namely the number of positive data but is classified as wrong by the system. FN is a False Negative, namely the number of negative data but is classified incorrectly by the system [12].

3. RESULTS AND DISCUSSION

3. 1 Benchmark CNN Normal And CNN DenseNet Lampung Script

In this study, the CNN Normal and CNN DenseNet models were benchmarked from the results of training and testing. The benchmark aims to determine whether the CNN DenseNet model used in Lampung script recognition has a high accuracy than the CNN Normal model. Pada Table 4 shows the training and testing benchmarks for the CNN Normal and CNN DenseNet models.

Table 4. Benchmark CNN Normal and CNN DenseNet

No.	Data	Training		Testing	
		CNN Normal	CNN Densenet	CNN Normal	CNN DenseNet
1.	Parent character	0.87	0.98	0.83	0.98
2.	Parent letter with sub-letter above	0.90	0.98	0.90	0.98
3.	Parent character with sub-letter on the side	0.92	0.99	0.87	0.98
4.	Parent character with sub-letters below	0.73	0.94	0.77	0.97

Table 4 shows the benchmark results between the CNN Normal model and the CNN DenseNet model. The learning speed results of the CNN Normal model are better than the CNN DenseNet. The reason is the difference in the number of layers where CNN Normal has 16 layers while CNN DenseNet has 128 layers. The level of comparison of accuracy in training and testing is also different. Benchmark results show that CNN DenseNet is superior to CNN Normal.

3. 2 Evaluation of CNN Normal and CNN DenseNet Lampung Script

In this study, the CNN Normal and CNN DenseNet models were evaluated from the results of the confusion matrix. Table 5 shows the evaluation of the Normal CNN model and Table 6 shows the evaluation of the DenseNet CNN model.

Table 5. CNN Normal evaluation results

No	Data	Accuracy	Precision	Recall	F1-score
1.	Parent character	0.77	0.80	0.77	0.76
2.	Parent letter with sub-letter above	0.90	0.92	0.90	0.90
3.	Parent character with sub-letter on the side	0.91	0.92	0.91	0.91
4.	Parent character with sub-letters below	0.78	0.82	0.78	0.77

Table 6. CNN DenseNet evaluation results

No	Data	Accuracy	Precision	Recall	F1-score
1.	Parent character	0.98	0.99	0.98	0.98
2.	Parent letter with sub-letter above	0.99	0.99	0.99	0.99
3.	Parent character with sub-letter on the side	0.99	0.99	0.99	0.99
4.	Parent character with sub-letters below	0.99	0.99	0.99	0.99

Table 5 shows the evaluation results of the Normal CNN model and Table 6 shows the results of the DenseNet CNN model evaluation. The level of comparison of the evaluation results for the two models is different. The performance of the DenseNet CNN model is much better than that of the Normal CNN. It can be seen from the accuracy, precision, recall, and f1-score. So the CNN DenseNet model is feasible to use for Lampung script recognition.

3. 3 Benchmark CNN Normal And CNN DenseNet Handwriting Lampung Script

In this study, the CNN Normal and CNN DenseNet models were benchmarked from the results of training and testing. The benchmark aims to determine whether the CNN DenseNet model used in Lampung script recognition has a high accuracy than the CNN Normal model. Table 7 shows the training and testing benchmarks for the CNN Normal and CNN DenseNet models.

Table 7. Benchmark CNN Normal and CNN DenseNet

No.	Data	Training		Testing	
		CNN Normal	CNN Densenet	CNN Normal	CNN DenseNet
1.	Parent character	0.60	0.57	0.80	0.62
2.	Parent letter with sub-letter above	0.79	0.46	0.86	0.43
3.	Parent character with sub-letter on the side	0.87	0.87	0.84	0.91
4.	Parent character with sub-letters below	0.93	0.68	0.90	0.68

Table 7 shows the benchmark results between the CNN Normal model and the CNN DenseNet model. The learning speed results of the CNN Normal model are better than the CNN DenseNet. The reason is the difference in the number of layers where CNN Normal has 16 layers while CNN DenseNet has 128 layers. The level of comparison of accuracy in training and testing is also different. Benchmark results show that CNN Normal is superior to CNN DenseNet.

3. 4 Evaluation of CNN Normal and CNN DenseNet Handwriting Lampung Script

In this study, the CNN Normal and CNN DenseNet models were evaluated from the results of the confusion matrix. Table 8. shows the evaluation of the CNN Normal model and Table 9. shows the evaluation of the CNN DenseNet model.

Table 8. CNN Normal evaluation results

No	Data	Accuracy	Precision	Recall	F1-score
1.	Parent character	0.73	0.78	0.72	0.72
2.	Parent letter with sub-letter above	0.87	0.88	0.87	0.86
3.	Parent character with sub-letter on the side	0.88	0.88	0.88	0.86
4.	Parent character with sub-letters below	0.96	0.98	0.96	0.96

Table 9. CNN DenseNet evaluation results

No	Data	Accuracy	Precision	Recall	F1-score
1.	Parent character	0.68	0.79	0.68	0.68
2.	Parent letter with sub-letter above	0.51	0.43	0.51	0.44
3.	Parent character with sub-letter on the side	0.88	0.91	0.88	0.88
4.	Parent character with sub-letters below	0.64	0.60	0.64	0.59

Table 8 shows the evaluation results of the Normal CNN model and Table 9 shows the results of the DenseNet CNN model evaluation. The level of comparison of the evaluation results for the two models is different. The performance of the CNN Normal model is much better than that of CNN DenseNet. It can be seen from the accuracy, precision, recall, and f1-score. Furthermore, the CNN DenseNet model was carried out by introducing Lampung script handwriting. The recognition aims to determine the extent to which the model recognizes the handwritten image of Lampung script.

3. 5 Lampung Script Recognition Results

In this study, the image of Lampung script will be recognized according to the character class. This process aims to determine whether the evaluated model can recognize the image of Lampung script or not. Table 10 shows the results of the introduction of Lampung script.

Table 10. Results of Lampung script recognition

No.	Data	Recognition
1.	Parent character	All parent character data can be identified according to the character class.
2.	Parent letter with sub-letter above	All parental data with subletters above can be recognized according to their character class.
3.	Parent character with sub-letter on the side	All parental data with sub-letters on the side can be recognized according to the character class.
4.	Parent character with sub-letters below	All parental data with sub-letters below can be recognized according to their character class.

Table 10 shows the results of the recognition of Lampung script. The CNN DenseNet model that has been evaluated is used for the image recognition process of Lampung script. The recognition results show that all data can be recognized according to the character class. So that the CNN DenseNet model that we chose in this study was successful for image recognition of Lampung script.

3. 6 Lampung Script Handwriting Recognition Results

In this study, the handwritten image of Lampung script will be recognized according to the class of the script. This process aims to determine whether the evaluated model can recognize the handwritten image of Lampung script or not. Table 11 shows the results of handwriting recognition of Lampung script.

Table 11. Results of handwriting recognition of Lampung script

No.	Data	Recognition
1.	Parent character	Only 14 handwritten data of the main character can be recognized according to the character class.
2.	Parent letter with sub-letter above	Only 47 handwritten data of the parent script with the above sub-letters can be recognized according to the character class.
3.	Parent character with sub-letter on the side	Only 45 handwritten data of parent characters with subletters on the side can be recognized according to their character class.
4.	Parent character with sub-letters below	Only 45 handwritten data of parent characters with lower letters can be recognized according to their character class.

Table 11 shows the results of the recognition of Lampung script. The CNN DenseNet model that has been evaluated is used for the process of image recognition of Lampung script handwriting. The recognition results show that all data cannot be recognized according to the character class. So that the CNN DenseNet model that we chose in this study has not been successful for image recognition of Lampung script handwriting.

4. CONCLUSIONS

In this study, the CNN DenseNet architecture got a much better accuracy than the CNN Normal architecture on the image of Lampung script document. In the handwritten image of Lampung script, the CNN Normal architecture gets much better accuracy than the DenseNet architecture. In the image of Lampung script documents, CNN Densenet architecture managed to recognize all data according to the script class. In the handwritten image of Lampung script, DenseNet architecture has not succeeded in recognizing all data according to its class. The possible cause is the lack of a dataset on the handwritten image of Lampung script.

ACKNOWLEDGEMENTS

REFERENCES

[1]   Fitriawan, H. & Setiawan, H., 2016, Neural Networks for Lampung Characters Handwritten Recognition, *2016 International Conference on Computer and Communication Engineering (ICCCE)*, 485–488.

[2]   ALWI, A., & Wardoyo, R. (2010). Pengenalan pola huruf-huruf lontara Bugis - Makassar dengan menggunakan jaringan syaraf tiruan metode backpropagation (Doctoral dissertation, [Yogyakarta]: Universitas Gadjah Mada).

[3]   SUDARMAN., & Hartati, S. (2019). Pengenalan Aksara Lontara Menggunakan Convolutional Neural Network (Doctoral dissertation, [Yogyakarta]: Universitas Gadjah Mada).

[4]   Dewa, C.K., Fadhilah, A.L. & Afiahayati, A., 2018, Convolutional Neural Networks for Handwritten Javanese Character Recognition, *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 12, 1, 83.

[5]   ARYA, I. K., & Harjoko, A. (2017). Pengenalan Tulisan Tangan Aksara Bali Menggunakan Convolutional Neural Network (Doctoral dissertation, [Yogyakarta]: Universitas Gadjah Mada).

[6]   Widiarti, A.R., Harjoko, A., Marsono & Hartati, S., 2017, The model and implementation of javanese script image transliteration, *Proceedings – 2017 International Conference on Soft Computing, Intelligent System and Information Technology: Building Intelligence Through IOT and Big Data, ICSIIT 2017*, 2018-January, 51–57.

[7]   Iskar., & Basyar, M., 2012, Kamus Bahasa Lampung, CV. Smart Cipta Intelekta

[8]   Krizhevsky, A., Sutskever, I. dan Hinton, G.E., 2012, ImageNet Classification with Deep Convolutional Neural Networks, F Pereira, C J C Burges, L Bottou, dan K Q Weinberger (ed.), Advances in Neural Information Processing Systems 25, [Online], Curran Associates, Inc., hal. 1097–1105, tersedia di http://papers.nips.cc/paper/4824-imagenet-classification-withdeep-convolutional-neural-networks.pdf.

[9]   Arrofiqoh, E.N. & Harintaka, H., 2018, Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi, Geomatika, Uniiversitas Gadjah Mada, 24, 2, 61.

[10] Christian, 2020, Max Pooling, Average Pooling, Global Max Pooling and Global AveragePooling, tersedia di https://www.machinecurve.com/index.php/2020/01/30/what-are-maxpooling-average-pooling-global-max-pooling-and-global-average-pooling/.

[11] Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K.Q., 2017, Densely connected convolutional networks, Proceedings - 30th IEEE Conference on Computer Vision and

Pattern Recognition, CVPR 2017, 2017-Janua, 2261– 2269.

[12]   Masías, V.H., Valle, M., Morselli, C., Crespo, F., Vargas Schüler, A. dan Laengle, S., 2016, Modeling Verdict Outcomes Using Social Network Measures: The Watergate and Caviar Network Cases, PloS one, [Online] 11e0147248, tersedia di DOI:10.1371/journal.pone.0147248.