HANDWRITING RECOGNITION OF SUNDANESE SCRIPT USING CONVOLUTIONAL NEURAL NETWORK METHOD

Imam Ramadhan[1], Galih Hermawan[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : imamramadhan548@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRACT

Sundanese script is the original font used by the Sundanese to write down words in the Sundanese language Sundanese script is used, it is also quite complicated and unique in writing. Handwriting patterns related to Sundanese script is unique, then the research was undertaken about pattern recognition image of Sundanese script handwriting. In previous research associated with pattern recognition handwriting has not the discovery of optimal results from the use of the methods previously discussed about Handwritting Recognition, therefore it takes a method that is able to improved accuracy in terms of pattern recognition of handwritten. Convolutional Neural Network (CNN) is a method of classification to be used in this research. Previous research on CNN have proven keefektifitasannya in regards to detect Java script writing on the picture scence, and has good accuracy results for recognizing faces in real-time. After testing the system with the Black Box method can be drawn the conclusion that the program works fine. In the process of testing the combination of 3 different tests with a total of 250 characters has been prepared, the program is able to detect the Sundanese script handwriting with the accuracy of 62% in the combination of learning rate 0.1 and epoch 10.

Keyword: Pattern Recognition, Sundanese Script, Convolutional Neural Network, Handwritting, Image Processing.

1. INTRODUCTION

Sundanese script is the original font used by the Sundanese to jot down the words in the Language that is used, in addition to having a unique letter character, Sundanese script is also quite complicated in writing. Associated with pattern writing Sundanese script is unique, then the research was undertaken about pattern recognition image of Sundanese script handwriting.

Handwriting is one of the unique things that belong to everyone because of the writings of someone who is certainly different from one another, although the approach is the same. Based on such differences many studies already conducted related handwriting. Previous research associated with the image recognition can recognize handwritten latin letters used [1] and face recognition in real-time [2].

This research was conducted aiming to find out how accurate the method used in this research to recognize handwriting patterns, because previous studies have not found the optimal results from the use of the previous methods discussing the study of handwriting recognition Handwritting Recognition. The research on get the best accuracy average of 55.7% [1].

Convolutional Neural Network (CNN) is a method of classification to be used in this research, CNN is a variation of the Multilayer Perceptron (MLP) which terinsfirasi of neural network humans [2]. Initial research underlying this research first done by Hubel and Wiesel [3]. In general the type of layer on CNN differenciated into two layers and layer image feature extraction classification. Previous research on CNN has proven to efektifitasannya in regards to detect Java script text on image scence [4] and get 87.48% accuracy to recognize faces in real-time [2].

Then, in this study will be made of the application pattern recognition of handwritten characters using the method of sunda Convolutional Neural Network (CNN). Then thereafter will do the calculation accuracy is done by counting the number of Sundanese script handwriting patterns that can be recognized by the application.

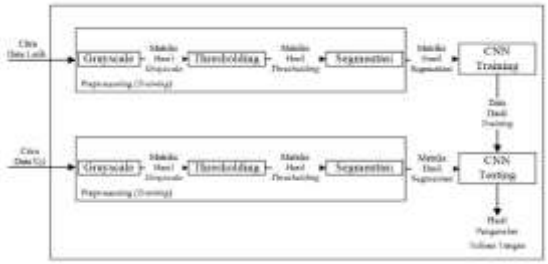
2. CONTENT OF THE STUDY

The problem of the research that is not yet the known accuracy of pattern recognition of handwritten Sundanese language classification method with Convolutional Neural Network (CNN). Previous research based on CNN method has never been fully combined with the method Sauvola Thresholding in the case of Sundanese script handwriting recognition, in this study the method of CNN will be combined with Sauvola method Thresholding algorithm by combining both simultaneously into research, both of these methods are good enough the achievement results of its accuracy. Both of these methods are used to resolve the problems about handwriting recognition letter-shaped print. Thresholding process is very important in this research because of the process that new classification method can do and knowing the accuracy of handwriting recognition [5],

so that the necessary process of Thresholding Sauvola beforehand.

The system will be built that is divided into two processes, namely the process of training and testing, training and testing processes in it there is preprocessing that includes grayscale, threshold, and sauvola segmentation and image after preprocessing new diakukan then finished the training process as well as well as testing. For data input that will be used i.e. handwriting image format .jpg image. The image can be obtained through the scan or photo camera with a resolution of at least 13 megapixels. Image of handwriting used written using pen/marker black in a white paper background with HVS. While the imagery used only Sundanese Script ngalagena only. Here is the general architecture of the system to be built.

Image 1 Overview Of The System



2.1. Data Input

Handwriting image that is used as a trainer is writing image data that is composed of the character ka, ga, nga, ca, it, ja, da, ta, na, pa, ba, ma, ya, ra, la, wa, sa, ha, fa, va, qa, za, xa, kha and sya. Data input or training data are written above the paper with a plain white background. In this study the image of the characters used only three characters, namely ca, ja and used to facilitate researcher in visualizing the values generated from each process.

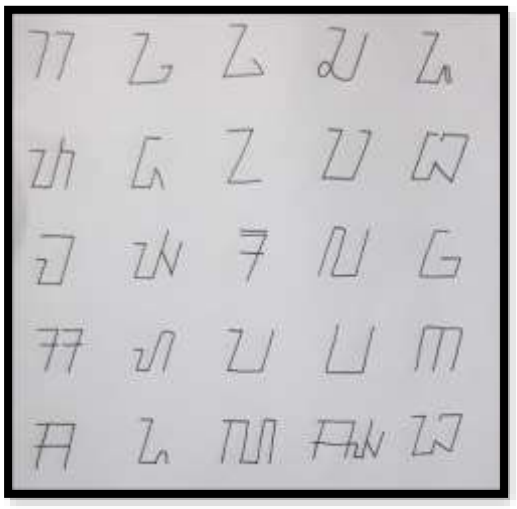


Image 2 Data Input

2.2. Digital Image Processing

Digital image processing means doing two dimensional image processing through a digital computer. image processing is a general term for various techniques that occurrence to manipulate and modify the image in various ways [6]. The photo is an example of a two dimensional image can be processed easily. Each photo in the form of a digital image can be processed using specific software. For example the dark image can be processed so that it becomes more bright, change the format of the color image into grayscale, threshold, binerisasi, segmentation, image resizing and so on.

In digital image processing in research of this bias are called preprocessing. In the process the input data preprocessing image good image trainer as well as a test image will be processed in several stages, namely grayscale, binerisasi, thresholding, segmentation and resizing. This process aims to qualify that have to be fulfilled in this study prior to the training process or testing.

2.2.1 Grayscaling

Grayscale aims to handle the gradations of black and white, and the result of course will change color (RGB) to the effect of the color gray. The color of the image is revealed with the intensity of [7]. In this process, the intensity range from 0 up to 255. A value of 0 is black and 255 States declared white. Change the format of the color to grayscale functions to shrink the range of colors from 0 up to 255. Here is the conversion of colored (RGB) image into grayscale.

$$I = (0.2989*R)+(0.587*G)+(0.1141*B) \quad (1)$$

Description:

I	=	The search function is the value of the scale of the grayish
R	=	The red component value (Red) of a pixel point
G	=	The component values of the Green (Green) from a pixel point
B	=	The component values of the blue (Blue) of a pixel point

The above equation is selected in this study because the human eye is naturally more sensitive to red and green light. Therefore, these colors were given higher weights to ensure that the balance in the relative intensity of the resulting grayscale image is similar to RGB color image [7].



Image 3 Example Result Grayscaling

2.2.2 Thresholding

Sauvola threshold threshold method is capable of detecting image handwriting very well, by calculating the threshold value of the dynamic range of the grayscale image standard deviation value [8]. Sauvola included in local threshold threshold value which is determined by the value of the neighboring pixels, threshold is a process he does aim to simplify the form of the image. Here is the formula that is used in the process of thresholding.

$$T(x,y) = m(x,y) * (1 + k * (\frac{s(x,y)}{R} - 1)) \quad (2)$$

$$m(x,y) = \frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} img(i,j)}{i*j} \quad (3)$$

$$s(x,y) = \sqrt{\frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} (img(i,j)-m(x,y))^2}{(i*j)-1}} \quad (4)$$

Description:

R = The maximum value of the standard deviation (128 to the image grayscale)
k = The kernel with a value between 0.2 – 0.5
m = The function that generates the average value from a number of image pixels
s = Functions that generate the value of the standard deviation from a number of image pixels
T = A function that produces a value of threshold values for the coordinates of the image width in the formula (i)
x = A function that produces a value of threshold values for the coordinates of the image width in the formula (i)
y = The value of the image height in coordinate formula (j)

After being processed then obtained the value threshold of T (x, y), i.e. the next process to enter the results of the above to the following equation.

$$f(x,y) = \begin{cases} 0, & img(x,y) < T(x,y) \\ 255, & img(x,y) \geq T(x,y) \end{cases} \quad (5)$$

Description:

f = A function that produces a value of 0 or 255
Img = Image value grayscale



Image 4 Example Result Sauvola Threshold

2.2.3 Segmentasi

Segmentation is a technique for dividing an image into multiple areas (regions) in which each region has the resemblance of the attribute [7]. Some of the techniques has segmentation include floating, connected component labelling, segmentassi-based cluster, and hough transformation. Focus on image segmentation in this research is to use floating and connected component labelling. In this process, the used input i.e. black and white imagery results sauvola threshold. Further cuts will be made to get the image of the handwritten letter.



Image 2 Segmentation Example

2.2.4 Resize

Resize IE changing the pixel size of an image or as needed. In the process the resize resize pixel not only decrease but can also increase the size of the pixels, besides resizing can also improve the image quality may even exacerbate an image accordingly. On the research results of Normalizing pixel Resizing/segmentation size 6 x 6 be scaled down.

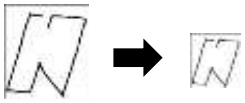
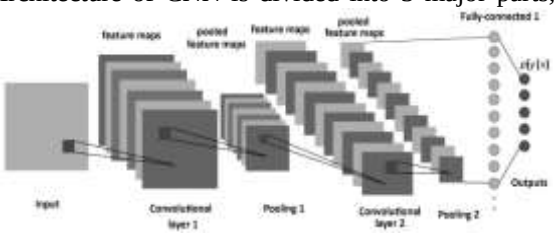


Image 3 Resize Example

2.2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of neural network used in the image data. CNN may be used to detect and identify objects in an image [9]. Architecture of CNN is divided into 3 major parts,



Convolutional Pooling Layer, Layer, and the Fully-Connected Layers. Following convolutional neural network architecture.

Image 7 CNN Architecture

A. Convolutional Layer

Convolutional layer composed of neurons are arranged in such a way that it forms a set of filter length and height (pixels). Each of these filters will be shifted across part of the pictures one by one. Each shift will do the operation "dot" between the inputs

and the value of the filter so as to produce an output or commonly referred to as activation folder or folder feature. On convolutional layer will be made between the konvolusi image input matrix with matrix-matrix filter F. Filter-this filter will be shifted to the entire surface of the picture so that it generates the output matrix featured FM folder.
With the hyperparameter above it will be featured with FM folder generated size obtained from the following formula:

$$\text{featuresize} = \frac{\text{inputsize} - \text{filtersize} + 2\text{pad}}{\text{str}} + 1 \quad (6)$$

Description:
featurezise = Size feature map
inputsize = Matrixs size input
filtersize = Filters size
Pad = zero padding
str = stride

The steps that are performed on a convolutional layer is as follows.

$$FM_{x',y'} = \sum_{x=1}^{size} \sum_{y=1}^{size} (A_{x,y} * F_{x,y}) + bF \quad (7)$$

Description:
$FM_{x',y'}$ = pixels feature map to x', y'
$A_{x,y}$ = pixels matrix input has been given a zero padding ke x, y
$F_{x,y}$ = pixel filter ke x, y
size = Filters size

And then run the activation function on each of the featured folder FM. Each pixel featured FM folders will be included in the activation function ReLU, where each pixel <0 value will be 0, with the formula.

$$\text{ReLU}(x) = \max(0, x) \quad (8)$$

Description:
ReLU(x) = Aktivation fungsion ReLU
x = Pixels

B. Pooling Layer

Pooling layer consists of a filter with a specific size and stride will be shifted on the entire area of the featured folder. Pooling to be used in this study was Max Pooling. The purpose of the use of pooling layer is reducing the dimension of the feature folder (downsampling), thus accelerating the computation because the parameters that should be updated less and less. The workings of the pooling layer is as follows. Put the window on the upper left corner of the featured folder. Here is the formula.

$$P_{x',y'} = \max(R_{x,y}, \dots, R_{x+\text{window},y+\text{window}}) \quad (9)$$

Description:
$P_{x',y'}$ = pixel feature map pooling to x', y'
$R_{x,y}$ = pixel feature map ReLU to x, y
Window = Window size

2.2.6 Fully Connected Layer

Fully Connected Layer at its core is a multilayer perceptron neural netrowk (MLP), which has some hidden layer, activation function, the output layer and the loss function. Fully connected layer-the one who will act for classifying data input. The output resulting from pooling layer still shaped multidimensional arrays, so it will be flatten in advance to change the vector data before input to be made fully connected layers.

A. Input Layer

Input layer X is an amalgamation of the matrix-the matrix feature folder P gained from pooling layer and stretching it into a vector along the number of pixels the entire third featured folder. Total nodes of the input layer is the sum of pixel featured folder P. Values of the input layer node X to be used in the calculation of hidden layer Z.

B. Hidden Layer

Each node in the input layer X will send the input signal to every hidden layer z. each node Xi will be multiplied by the weighting Vj, i then added with bias V0, i.e. to generate the value of the input z_ini. The formula of calculating the input hidden layer z_in is as follows.

$$z_{in_i} = \sum_{i=1}^{n} X_j * V_{j,i} + V_{0,i} \quad (10)$$

Description:
z_in_i = Input for node hidden layer Z to-i with the number of node n
X_j = node X to-j
$V_{j,i}$ = weight V for node Xj and node Zi
$V_{o,i}$ = bias V for node z_in_i

After that calculate the value of the output value by activating input Z z_in using the ReLU activation function, as follows.

$$Z_i = \text{ReLU}(z_in_i) = \max(0, z_in_i) \quad (11)$$

Description:
Z_i = Input to node hidden layer to-i
z_in_i = Input to node hidden layer to-i

C. Output Layer

Each node in the output layer Zj will send the input signal to the output layer Yi. Each node Z will be multiplied by the weighting Wj, i then added with bias W0, i. enumeration of input output layer y_in is as follows.

$$y_{in_i} = \sum_{i=1}^{n} Z_j * W_{j,i} + W_{0,i} \quad (12)$$

Description:

y_in_i = Input to node hidden layer Z to-i with the number of node n

Z_j = node Z ke-j

$W_{j,i}$ = weight W for node Zj and node Yi bias V for node z_in_i

$W_{o,i}$ = bias W for node y_in_i

After that activate the output value Y using the softmax activation function.

$$Y_i = \frac{e^{y_{in_i}}}{\sum_{i=1}^{m} e^{M}} \quad (13)$$

Description:

Y_i = Output for node output layer

y_in_i = Input for node output layer to-i

M = all input for node output layer, numbering m.

2.2.7 Backpropagation

At this stage of training will be done on backpropagation class letters. At this stage of the adjustment process will be performed each weight and bias based on the error obtained at this stage of feedforward. The first gradient will be calculated from the loss function of all parameters (weight and bias) by finding the partial derivative (partial derivative) of the function. After it updates all the parameters by using Stochastic Gradient Descent (SGD).

A. The calculation of the Loss Function

Loss function is a function that measures how good the performance of the neural network in the prediction of conduct towards the target. Loss function will calculate the loss, that is the difference from the value of output with the expected target value. To the problem of classification, the loss function used is cross-entropy loss function :

$$L = -\sum_{i=1}^{m} t_i \log(Y_i) \quad (14)$$

Description:

L = loss / error

t_i = Value vektor to-i one-hot the expected target

Y_i = Value matrix output on stage feedforward to-i with the number of classes m

B. Calculation Of The Gradient Towards The Parameters W_{ji}

After that calculate the gradient of the error parameter Wji weights against, using the chain rule formulas.

$$\frac{\partial L}{\partial W_{ji}} = \frac{\partial L}{\partial Y_i}\frac{\partial Y_i}{\partial y_in_i}\frac{\partial y_in_i}{\partial W_{ji}} \quad (15)$$

$$\frac{\partial L}{\partial Y_i} = \frac{Y_i - t_i}{Y_i(1-Y_i)} \quad (16)$$

$$\frac{\partial Y_i}{\partial y_in_i} = Y_i(1 - Y_i) \quad (17)$$

$$\frac{\partial y_in_i}{\partial W_{ji}} = Z_j \quad (18)$$

Then for the filter W_{ji}

$$\frac{\partial L}{\partial W_{ji}} = Y_i(1 - Y_i)Z_j \quad (19)$$

And for bias $W_{0,i}$

$$\frac{\partial L}{\partial W_{ji}} = Y_i(1 - Y_i) \quad (20)$$

Description:

$\frac{\partial L}{\partial W_{ji}}$ = Weight gradient W_{ji} terhadap loss L

$\frac{\partial L}{\partial W_{0,i}}$ = Gradient bias $W_{0,i}$ against loss L

$\frac{\partial L}{\partial Y_i}$ = Value gradient node Y_i against loss L

$\frac{\partial Y_i}{\partial y_in_i}$ = Gradient value matrix y_in_i against value node Y_i

$\frac{\partial y_{in_i}}{\partial W_{ji}}$ = Weight gradient W_{ji} against value matrix y_in_i

Y_i = Value node output layer to-i

y_{in_i} = Input Value to output layer to-i

Z_j = value node hidden layer to-j

C. Calculating The Gradient Towards The Weighting Parameters V_{kj} dan Bias $V_{0,j}$

After that calculate the gradient of the error parameter Vkj weights against, using the chain rule formula.

$$\frac{\partial L}{\partial V_{kj}} = \sum_{i}^{m} \frac{\partial L}{\partial Y_i}\frac{\partial Y_i}{\partial y_in_i}\frac{\partial y_in_i}{\partial Z_j}\frac{\partial Z_j}{\partial z_in_j}\frac{\partial z_in_j}{\partial V_{kj}} \quad (21)$$

$$\frac{\partial L}{\partial Y_i} = \frac{Y_i - t_i}{Y_i(1-Y_i)} \quad (22)$$

$$\frac{\partial Y_i}{\partial y_in_i} = Y_i(1 - Y_i) \quad (23)$$

$$\frac{\partial y_in_i}{\partial Z_j} = W_{ji} \quad (24)$$

$$\frac{\partial Z_j}{\partial z_in_j} = Z_j\left(1 - Z_j\right) \quad (25)$$

$$\frac{\partial z_in_j}{\partial V_{kj}} = X_k \quad (26)$$

Then for the filter V_{kj}

$$\frac{\partial L}{\partial V_{kj}} = \sum_{i}^{m}(Y_i - t_i)(W_{ji})(Z_j\left(1 - Z_j\right))(X_k) \quad (27)$$

And for bias $V_{0,j}$

$$\frac{\partial L}{\partial V_{kj}} = \sum_{i}^{m}(Y_i - t_i)(W_{ji})(Z_j\left(1 - Z_j\right)) \quad (28)$$

Description:

$\frac{\partial L}{\partial W_{ji}}$ = Gradien weights V_{kj} against loss L

$\frac{\partial L}{\partial W_{0,i}}$ = Gradien bias $V_{0,i}$ against loss L

$\frac{\partial L}{\partial Y_i}$ = Gradien value node Y_i against loss L

$\frac{\partial Y_i}{\partial y_in_i}$ = Gradien matrixs value y_in_i value againtst node Y_i

$\frac{\partial y_{in_i}}{\partial W_{ji}}$ = Gradien weight W_{ji} against the values of a matrix y_in_i

Y_i = Value node output layer to-i

y_{in_i} = Input value output layer to-i

Z_j = Value node hidden layer to-j

D. **Calculating the gradient towards the filter parameters F and bF bias**

After that calculate the gradient of the error against the filter parameter f. According the rules of the chain rule, we have to find gradients to layer-layer earlier in advance. Then calculate the gradient of each filter with a gradient obtained previously.

$$\frac{\partial L}{\partial F} = \frac{\partial L}{\partial X}\frac{\partial X}{\partial P}\frac{\partial P}{\partial F} \quad (29)$$

Description:

$\frac{\partial L}{\partial F}$ = gradien value filter F against p loss L

$\frac{\partial L}{\partial X}$ = gradien valuei node X against loss L

$\frac{\partial X}{\partial P}$ = gradien value feature map P value againts node X

$\frac{\partial P}{\partial F}$ = gradien value filter F value against feature map P

First calculate the value $\frac{\partial L}{\partial X}$. calculate gradien the weighting parameters against errors X_k, using the chain rule formula.

$$\frac{\partial L}{\partial X_k} = \sum_j^n \sum_i^m \frac{\partial L}{\partial Y_i}\frac{\partial Y_i}{\partial y_in_i}\frac{\partial y_in_i}{\partial Z_j}\frac{\partial Z_j}{\partial z_in_j}\frac{\partial z_in_j}{\partial X_k} \quad (30)$$

$$\frac{\partial L}{\partial Y_i} = \frac{Y_i - t_i}{Y_i(1-Y_i)} \quad (31)$$

$$\frac{\partial Y_i}{\partial y_{in_i}} = Y_i(1 - Y_i) \quad (32)$$

$$\frac{\partial y_in_i}{\partial Z_j} = W_{ji} \quad (33)$$

$$\frac{\partial Z_j}{\partial z_in_j} = Z_j\left(1 - Z_j\right) \quad (34)$$

$$\frac{\partial z_in_j}{\partial X_k} = V_{kj} \quad (35)$$

Then

$$\frac{\partial L}{\partial X_k} = \sum_j^n \sum_i^m (Y_i - t_i)(W_{ji})(Z_j\left(1 - Z_j\right))(V_{kj}) \quad (36)$$

Description:

$\frac{\partial L}{\partial X_k}$ = Gradien value node X_k against loss L

$\frac{\partial L}{\partial Y_i}$ = Gradien nilai node Y_i terhadap loss L

$\frac{\partial L}{\partial Y_i}$ = Gradien nilai node Y_i terhadap loss L

$\frac{\partial Y_i}{\partial y_in_i}$ = Gradien matrixs value y_in_i value against node Y_i

$\frac{\partial y_{in_i}}{\partial Z_j}$ = Gradien weight W_{ji} value against matrixs y_in_i

$\frac{\partial Z_j}{\partial z_{in_j}}$ = Gradien nilai matriks z_{in_j} terhadap nilai node Z_j

$\frac{\partial z_{in_j}}{\partial X_k}$ = Gradien value node X_k value against matrixs z_{in_j}

Y_i = Value node output layer to-i

y_{in_i} = Input Value output layer to-i

Z_j = Value node hidden layer to-j

z_{in_j} = Input value hidden layer to-j

X_k = Value node input layer to-k

m = Amount node output layer Y

= Amount node hidden layer Z

After that we calculate the count value $\frac{\partial X}{\partial P}$ by using the value obtained from the calculation $\frac{\partial L}{\partial X}$. This value is obtained from the search of a gradient in layer pooling layer. First we return mention vector shapes $\frac{\partial L}{\partial X}$ become a featured map gradients. For operation max-pooling feature map P', gradien $\frac{\partial X}{\partial P}$ look at the maximum values feature map in accordance with the results of max-pooling on stage feedforward. Gradient $\frac{\partial X}{\partial P}$ for pixels with maximum value is $\frac{\partial L}{\partial X}$, While the gradient for the pixels is 0. As for the bias gradient filters, total the each value on a gradient filter. Do for all filters.

$$\frac{\partial L}{\partial bF} = \sum_x^{size}\sum_y^{size}\frac{\partial P_{x,y}}{\partial F_{x,y}} \quad (37)$$

Description:

$\frac{\partial L}{\partial bF}$ = Gradien bias filters bF against loss L

$\frac{\partial P_{x,y}}{\partial F_{x,y}}$ = Gradient value filter F to-x,y value against feature map P to-x,y

size = Filters Size

E. Update Parameter Values

After all the gradients calculated parameters, update parameters by using Stochastic Gradient Descent. Stochastic Gradient Descent optimization method is used to find the optimal weighting values so obtained a small loss value, where the value of the loss represents a level of error (error) on the system. The search value weighting is done by way of changing the weight gradually while training, using the parameters that are called learning rate, i.e. how fast neural network learning. Here is the calculation to update the value of the parameter weights W '.

$$W_{x,y}' = W_{x,y} - \alpha(\frac{\partial L}{\partial W_{x,y}}) \quad (38)$$

Description:

$W_{x,y}'$ = Weight W to-x,y which has been updated
$W_{x,y}$ = Weight W to-x,y early
$\frac{\partial L}{\partial W_{x,y}}$ = Weirght gradient $W_{x,y}$ against loss L
α = learning rate

Here is the calculation to update parameter values the bias W_0'.

$$V_{x,y}' = V_{x,y} - \alpha(\frac{\partial L}{\partial V_{x,y}}) \quad (39)$$

Description:

$V_{x,y}'$ = Weight V to-x,y which has been updated
$V_{x,y}$ = Weight V to-x,y awal
$\frac{\partial L}{\partial V_{x,y}}$ = Weight gradient $V_{x,y}$ against loss L
α = learning rate

Here is the calculation to update parameter values the bias V_0'.

$$V_{0,y}' = V_{0,y} - \alpha(\frac{\partial L}{\partial V_{x,y}}) \quad (40)$$

Description:

$V_{0,y}'$ = Weight V_0 to-x,y which has been updated
$V_{0,y}$ = Bobot V_0 ke-x,y awal
$\frac{\partial L}{\partial V_{0,y}}$ = Weight gradient $V_{0,y}$ against loss L
α = learning rate

Then here is the calculation for the filter parameter value update F'.

$$F_{x,y}' = F_{x,y} - \alpha(\frac{\partial L}{\partial F_{x,y}}) \quad (41)$$

Description:

$F_{x,y}'$ = Weight F' to-x,y which has been updated
$F_{x,y}$ = Weight F' to-x,y early
$\frac{\partial L}{\partial F_{x,y}}$ = Weight gradient $F_{x,y}$ against loss L
α = learning rate

The following is the calculation for the last update filter parameter values bF'.

$$bF' = bF - \alpha(\frac{\partial L}{\partial bF}) \quad (42)$$

Description:

bF' = Weight bF to-x,y which has been updated
bF = Weight bF ke-x,y early
$\frac{\partial L}{\partial bF}$ = Weight gradient bF against loss L
α = learning rate

Results weighting parameters update V', bias V_0', bobot W', bias W_0', and filters F' and bias filters bF' will be used in the next iteration of feedforward. In the process of testing to find out the results of the accuracy of the calculation is only done introduction to feedforward only.

2.3 Testing Of The Accuracy

On testing the accuracy will be calculated the value of accuracy or suitability of the results of the application of Sundanese script handwriting recognition. Testing done after the data training already done the training process.

This testing is done to 250 data sets. The following is the experiments conducted with different parameters on test images.

Table 1 The Test Results

No.	Testing	Learning Rate	Epoch	Accuracy
1.	1	0,1	50	62,0%
2.	2	0,2	20	56,0%
3.	3	0,5	50	61,6%

Based on the results of testing accuracy against 10 test data that has been made, the evaluation of the results achieved against the accuracy of handwriting recognition that is Sundanese script as follows:

1. Accuracy results that have the highest percentage of IE 1st test with learning rate 0.1 and epoch 50 and obtained an average accuracy of 62%.
2. Accuracy results that have the lowest percentage of IE the 2nd test with learning rate 0.2 and epoch 20 and obtained an average accuracy of 56%.

Charging the learning rate and the epoch affecting how much amount of handwritten characters that can be detected by the program, but the greater the number of rate and learing epoch making the process of training the longer. On test data itself also affect the accuracy, the more presentable on the writing test data will be increasingly easy program to recognize the alphabet of Sunda.

3 COVER

This section contains the results of the study include namely the conclusions and suggestions.

3.1 Conclusion

This research has resulted in a system of Sundanese script handwriting recognition method using CNN. The introduction of 25 types of Sundanese Script output ngalagena.

Based on the results of the testing that has been done, the testing of the system the following handwriting recognition :

1. Handwriting recognition system using CNN obtained the best accuracy of 62%. This accuracy is affected by how many layers are used in the program and is produced in the training process.
2. The amount of learning rate and epoch also affects the accuracy of the testing process.
3. The abundance of handwritten characters not detected properly effected by the Sundanese script writing process on test data far from the word mess.
4. Handwriting recognition systems can recognize the characters of Sunda.

3.2 Advice

The current system was built can still be developed according to the needs of users in getting results better accuracy. Suggestions for future development on the system that can be used as a consideration that is:

1. Using the method of extraction of characteristics and classification of its better to get the best accuracy.
2. Using a different method of preprocessing in order to compare the accuracy with which other research.
3. Multiply training data in order to improve the accuracy of the results to the maximum.
4. Making data work with character inscriptions in price because it will help in the process of testing the accuracy of.

BIBLIOGRAPHY

[1] U. Rohwana and M. I. Irawan, “Pengenalan Tulisan Tangan Huruf Latin Bersambung Secara Real Time Menggunakan Algoritma Learning Vector Quantization,” Jurnal Sains dan Seni POMITS, vol. 2, no. 1, pp. 1-6, 2013.

[2] M. d. B. S.tiyono, “Convolutional Neural Network untuk Pengenal Wajah Secara Real-time,”Jurnal Sains dan Seni ITS, vol. 5, no. 2, pp. 72-77, 2016.

[3] W. Hubel.D, “Receptive fields and Functional Architecture of Monkey striate Cortex,” Journal of Physiology (London), no. 195, pp. 215-243, 1968.

[4] M.Afakh, A.Risnumawan, MAnggraeni et al, “Aksara Jawa Text Detection in Scence Images Using Convolutional Neural Network”, Proceedings - International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Vol.2017-Janua, 2017.

[5] Z.Hadjadj, A.Meziane, Y.Cherfa, “Improved Sauvola's Algorithm for Document Image Binarization,” Reasearch Center in Scientific and Technical Information (CERIST), vol.9730, no.1, pp.737-745, 2016.

[6] B. B. Agarwal, “ S. P. Tayal and M,"Software Engineering & Testing”, Boston:Jones And Bartlett Publishers,2010.

[7] A. Kadir and A. Susanto,"Teori dan Aplikasi Pengolahan Citra", Yogyakarta: ANDI, 2013.

[8] C. Solomon, and T. Breckon,"Fundamental of Digital Image Processing", West Sussex: John Wiley & Sons, Ltd, 2011..

[9] Y.Lecun, Y.Bengio, G.Hinton, “Deep Learning,” International Journal of Science, vol.521, no.7534, p.436-444, 2015.