

CONVOLUTIONAL NEURAL NETWORK IMPLEMENTATION ON HANDWRITTEN RECOGNITION

Hamzah Nur Al Falah¹, Ken Kinanti Purnamasari²

^{1,2} Indonesia Computer University

Dipati Ukur St. No.112-116, Lebakgede, Coblong, Bandung City, West Java 40132

E-mail: hamzahnuralfalah@gmail.com¹, ken.kinanti@email.unikom.ac.id²

ABSTRACT

Previous research has been conducted on Intelligence Character Recognition (ICR) using Convolutional Neural networks (CNN) with accuracy reaching 97.6% on Handwritten Arabic Character Dataset. Other research on CNN is said to be superior in case the introduction of the image using the data. Traces of such research, conducted research with the primary focus to measure the ability of CNN on the case of ICR alphabet United Kingdom with data derived from the National Institute of Standards and Technology (NIST) as much as training data and 3.100 15.872 test data with a class of characters include A-Z, a-z, and 0-9. Preprocessing is done before the training and testing of CNN which consists of stage grayscale, smoothing, thresholding, segmentation, and scaling. Stage feedforward CNN consists of layers layer extraction and classification. After feedforward, backpropagation is done to improve the value of the weights and biases when training so that it can be used for testing. With test parameters in the form of learning rate which consists of 0.001, 0.002, 0.003, 0.004, and 0.005 and epoch as much as 28 obtained accuracy of 72.48% in testing phase.

keywords: Artificial Intelligence, Neural Network, Convolutional Neural Network, CNN, Intelligence Character Recognition, ICR, Handwritten Character Recognition

1. INTRODUCTION

Previous research has some missing things, specially on recognizing the characters with similar traits [1]. This relates to the ability of the method in extracting characteristic of image.

Related research has been done using ICR Artificial Neural Network (ANN) and gradient feature extraction with accuracy reaching 86.2% [1]. In other studies obtained accuracy reach 85.62% using the method of multilayer feed forward neural network [2], and other studies obtained accuracy of 74 – 78% use ANN [3].

CNN is the method that utilizes the convolution feature for extracting characteristic of image, it is a distinguishing method of CNN with ANN. prior studies have been conducted on the ICR uses CNN with accuracy reaching 97.6% using Arabic Handwritten Character Dataset [4]. On the research of

other retrieved mencapai 89% accuracy with facial recognition in realtime case [5] , 96.1 – 97.9% superior to the eight other methods including SVM and RBF is the introduction of gender based on face [6]. With the achievement of a good degree of accuracy from earlier research, then the unambiguous case CNN to ICR alphabet United Kingdom with data obtained from NIST [7].

2. STUDY CONTENT

This chapter will explain the study of associated research methods, system architecture, digital image processing, algorithms of CNN, and CNN test results.

2.1 Research Method

This research was conducted by observing the flow of research here.

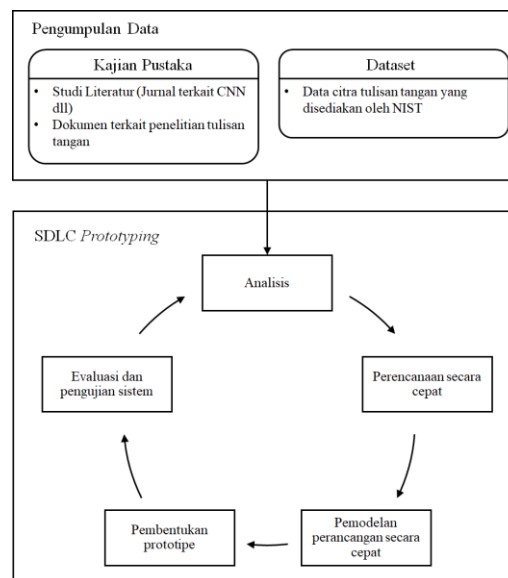


Image 1. Research Flow

This research begins with the data collection in order to know the achievements of previous research, avoid mistakes that had ever been done, to obtain advice and guidance in order for the research to be performed can be run with smoothly and resulted in better results. Data collection is like the related journal methods used, find handwritten dataset.

System Development Life Cycle (SDLC) prototyping model, was used that consisted of the analysis, planning, modeling in a quick planning, the establishment of rapid prototyping and evaluation of

testing the system. This is detailing of every step that was done on this reaserch.

- Analysis, review some methods that will be used in the system are using the books – books and journals.
- Fast planning, decisive the stages of the process that will be used in the software, which includes the selection of the method.
- Modeling design of construction started quickly, making the look and shape of the program. Planning in a quick, decide the stages of the process that will be used in the software, which includes the selection of the method.
- Prototype establishment, that was implemented all of fast planning for being an software.
- Testing and evaluation of the system, ‘Il find the error code or error of logic from the prototype that has been developed. Evaluation of the result from using CNN metode parameters , searching of the weaknesses also missing things from the prototype that was used such as choosing the parameters or the used metode.

2.2 Digital Image Processing

Digital image processing is the branch of informatics studies that aims to improve the quality of the image, to suit your needs. In this research, image processing is performed on the stage of preprocessing. The main stages of the preprocessing is grayscaling, smoothing with an integral image, thresholding using Sauvola Threshold combined with image segmentation using integrals, Connected Component Labeling (CCL), and scaling. Here is a block diagram for preprocessing is used.

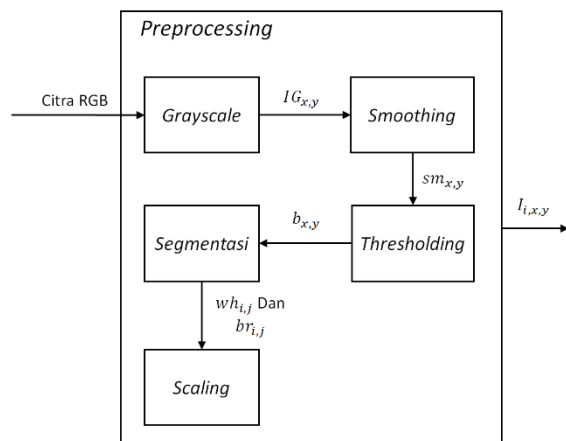


Image 2. Preprocessing Stage Architecture

2.2.1 RGB Image

RGB Image or colored image has three kinds of color channel i.e. Red, Green, and Blue. This image has 24 bits of color for each pikselnya with a range of 0 to 255.

2.2.2 RGB Image Convert To Grayscale

This step aims to reduce computational complexity, by changing the RGB color range on the

image into the degree of keabuan with a single color channel [8]. As for the architecture of this phase can be seen as follows.

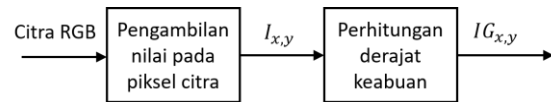


Image 3. Block Diagram Of Grayscale Stage

On the grayscaling step consists of two phases, the first is the retrieval of the value of the RGB image pixels, and the second stage is to convert that value to grayscale degree or grayscale image.

On the image recognition, the complexity will have an effect on the performance of the system, so the use of grayscale image compared to the RGB image has an execution time and lower complexity. moreover, with the use of a single channel, this color can be determined also the number of bits that will be made to facilitate the use of color.

2.2.3 Integral Image

The concept of the integral image is add up each pixel image with its neighbors to save the information value the total intensity of the image [9]. Here is block diagram of the phase integral image.

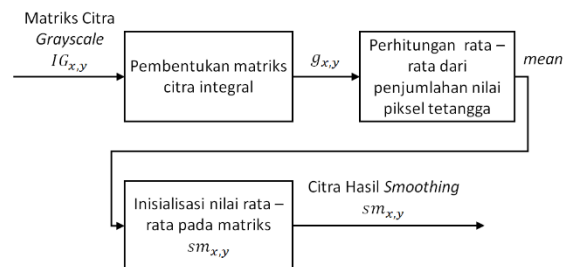


Image 4. Block diagram of integral image step

Establishment of the integral image of steps is like:

- Making data integral image matrix $g_{x,y}$ of the grayscale image. Stages of formation of the integral image matrix using Equation 1.

$$g_{x,y} = \sum_{i=0}^x \sum_{j=0}^y I_{i,j} \quad (1)$$

$$x = 0,1,2, \dots, M - 1$$

$$y = 0,1,2, \dots, N - 1$$

The value of the x and y based on the size of the matrix is represented with a grayscale image $M \times N$.

- Determine the dimensions of the mask for the range of median calculation of integral image align on the next step.
- calculation of averages of integral image on the dimension mask range.

$$mean(x, y) = \frac{(g_{x+dx-1, y+dy-1} + g_{x-dx, y-dy}) - (g_{x-dx, y+dy-1} + g_{x+dx-1, y-dy})}{wx \times wy}$$

$$dx = round\left(\frac{wx}{2}\right), \quad dy = round\left(\frac{wy}{2}\right),$$

$$x = 0, 1, 2, \dots, M - 1, \quad y = 0, 1, 2, \dots, N - 1$$

(2)

This calculation was carried out based on the range of the specified mask. A value from the range of mask will affect the value of wx and wy thus affecting the value of the integral image matrix is used.

Basically the use of integral image was similar to mean filtering method, that is one of the smoothing method that can gain value through the average value of the sum – pixels in range mask. In contrast to the mean filtering which gained value by counting one by one the value range of the integral image mask, utilizing the mathematical calculations on coordinates an integral image matrix to obtain the value of the median.

In this study, equation 2 is used for smoothing phase is represented by $sm(x, y)$, and at this stage of thresholding as one variable calculation of sauvola threshold.

2.2.4 Binary Image

A binary image is an image with a single channel color measuring 2 bits. This image consists only of value zero (0) and one (1). Cell in the matrix of the binary image of the white color represents one value, while zero represents the color black. Binary matrix is used as the input data at this stage of segmentation. Binary image itself is obtained by utilizing the threshold value as a limitation and a determinant of value. Here is a block diagram of thresholding phases.

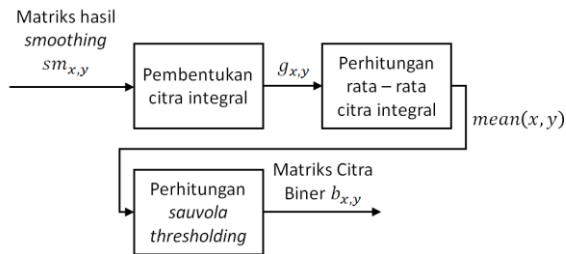


Image 5. Block diagram of thresholding step

A matrix is used as the input at this stage is the result of smoothing ($sm_{x,y}$). In this studies using data on the image with the background is black, so the binary image calculation done by the rules in the following equation.

$$b_{x,y} = \begin{cases} 1, & sm_{x,y} \leq T \\ 0, & sm_{x,y} > T \end{cases} \quad (3)$$

Matrix of $b_{x,y}$ is a representation of the binary image, where the T variable is threshold values used. Thresholding is the value that used as a reference in deciding a case. In this case to determine whether the value of the smoothing results on matrix $sm_{x,y}$ at certain coordinates are white or black. The given value on these variables will affect the resulting binary matrix. Sauvola threshold method is one of Locally Adaptive Thresholding methods, i.e., methods that are able to determine the value of the threshold that automatically based on the values of the pixels on the neighbouring range the dimensions of the mask and combined with an integral image [9]. Here is the equation that was used.

$$T(x, y) = mean(x, y) \left[1 + k \left(\frac{\partial(x, y)}{1 - \partial(x, y)} - 1 \right) \right]$$

$$\partial(x, y) = sm(x, y) - mean(x, y)$$

$$\partial(x, y) = \begin{cases} -1, & \partial(x, y) = 1 \\ \partial(x, y), & \partial(x, y) \neq 1 \end{cases} \quad (4)$$

Descriptions:

- $T(x, y)$ = The value of the sauvola threshold results
- $mean(x, y)$ = The result of Integral image with matrix input $sm(x, y)$
- $\partial(x, y)$ = Standard deviation
- k = Values are biased with ranges [0.1, 0.9], on this case using 0.1

2.2.5 Segmentasi Citra

This research used Connected Component Labeling (CCL) method. CCL is able to segment images based on relationships between pixels by utilizing the mask dimensions at the search stage, in this study a 5×5 dimension mask is used. Here is block diagram of the CCL stage.

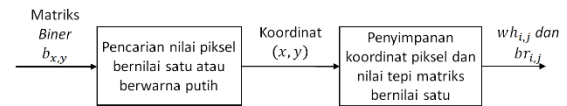


Image 6. Block diagram of segmentation step

The variables that need to be defined from the segmentation stage are like.

Table 1. Definition of the segmentation step

No	Variabel	Descriptions
1	$wh_{i,j}$	A matrix that stores white pixel coordinates (x, y), with i variable as an index for character objects and j as an index for pixel coordinate values.
2	$br_{i,j}$	Matrix that will store the coordinates of the outer limit of the object variable with characters as characters and objects index j as the index value of the coordinates. The variable

		j is composed of four grades, the first index represents the top limits, the second index to limit the right of the third index, for the lowest limit, and the limit for most of the fourth index left.
--	--	---

From Image 6 that can be developed and explained in the following stages.

- Search process to find pixel with one (1) as value starting from the coordinates (0,0) in the matrix of binary image $b_{x,y}$. If it then save the coordinates of the data in a matrix $wh_{i,j}$, then set the value of that pixel on $b_{x,y}$ with zero.
- Do the filling value matrix $br_{i,j}$. If the matrix $br_{i,j}$ not yet have a value are initialized, then simply x coordinate on the first and third index, the index on the y coordinate of the second and fourth. In addition to the first condition, the applicable provisions on the following equation.

$$\begin{aligned}
 br_{i,0} &= \begin{cases} x, & \text{if } (x < br_{i,0}) \\ br_{i,0}, & \text{etc.} \end{cases} \\
 br_{i,1} &= \begin{cases} y, & \text{if } (y > br_{i,1}) \\ br_{i,1}, & \text{etc.} \end{cases} \\
 br_{i,2} &= \begin{cases} x, & \text{if } (x > br_{i,2}) \\ br_{i,2}, & \text{etc.} \end{cases} \\
 br_{i,3} &= \begin{cases} y, & \text{if } (y < br_{i,3}) \\ br_{i,3}, & \text{etc} \end{cases} \quad (5)
 \end{aligned}$$

- The next Search is started based on the dimensions of the mask, with the center of the mask is at coordinates that have been stored in the prior matrix $wh_{i,j}$. As in phase b, if found pixels worth one (1) on the matrix $b_{x,y}$ then save the coordinate in matrix $wh_{i,j}$ and set the value of a matrix $b_{x,y}$ becomes zero (0). Do again phase c and d until there are not the pixels worth one on the dimensions of the mask for the matrix $b_{x,y}$. The search for the next character of the object, starting from the first coordinates stored on the matrix $wh_{i,j}$.

2.2.6 Scaling

Scaling on this research aimed to uniform the size of image segmentation the matrix results, i.e. $wh_{i,j}$ and $br_{i,j}$. The value of the coordinates of the image with the size of these diverse needs to be converted in order to be uniform, i.e. by 28×28 . Here is block diagram of scaling step.

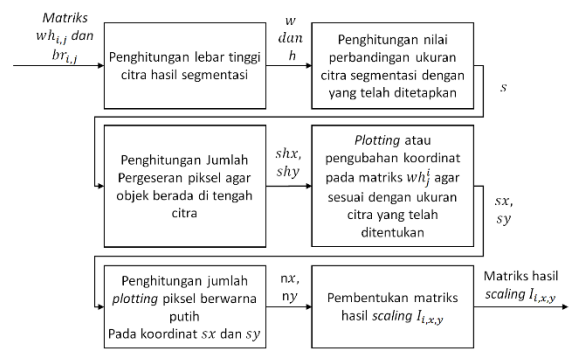


Image 7. Block diagram of scaling step

For the work steps are like :

- Calculate the width (w) and height (h) image segmentation results using the following equation.

$$\begin{aligned}
 w &= br_{i,1} - br_{i,3} + 1 \\
 h &= br_{i,2} - br_{i,0} + 1 \\
 i &= 1,2..n \quad (6)
 \end{aligned}$$

- Calculate the value of size comparison image segmentation result (s) and size ($u = 28$).

$$s = \begin{cases} \frac{u}{w}, & \text{if } (w \geq h) \\ \frac{u}{h}, & \text{if } (w < h) \end{cases} \quad (7)$$

- Count the number of shifts each pixel on the x and y coordinates for the image scaling results.

$$\begin{aligned}
 shx &= \begin{cases} \text{round}\left(\frac{u}{2}\right) - \text{round}\left(\frac{s \times h}{2}\right), & \text{if } (w \geq h) \\ 0, & \text{if } (w < h) \end{cases} \\
 shy &= \begin{cases} \text{round}\left(\frac{u}{2}\right) - \text{round}\left(\frac{s \times w}{2}\right), & \text{if } (w < h) \\ 0, & \text{if } (w \geq h) \end{cases} \quad (8)
 \end{aligned}$$

Descriptions:

- shx = Sum of the shifting pixel on coordinates row
 shy = Sum of the pixel shift on the column coordinate

- Scaling of coordinates, the idea at this stage is to convert existing coordinate values in a matrix $wh_{i,j}$ against the matrix size 28×28 .

$$\begin{aligned}
 sx &= (x - br_{i,0}) \times s + shx \\
 sy &= (y - br_{i,3}) \times s + shy \quad (9)
 \end{aligned}$$

Descriptions:

- sx = X coordinate scaling results
 sy = y coordinate scaling results
 x = X coordinate of the matrix value $wh_{i,j}$
 y = Y coordinate of the matrix value $wh_{i,j}$

- Calculate the amount of pixel is white begins plotting of sx and sy . This stage aims to determine

how many pixels are white obtained from one coordinate in matrix $wh_{i,j}$ when measuring 28×28 .

$$nx = \begin{cases} 2, & \text{if}(sx \bmod 1 > 0 \text{ and } sx > 1) \\ 1, & \text{lainnya} \end{cases} \quad (10)$$

$$ny = \begin{cases} 2, & \text{if}(sy \bmod 1 > 0 \text{ and } sy > 1) \\ 1, & \text{lainnya} \end{cases}$$

Description:

- nx = Plotting the number of pixels are white, starting from sx
- ny = Plotting the number of pixels are white, starting from sy

2.3 Algoritma CNN

CNN has two major layers, called convolution layer and classification layer. Convolution layer consisting of convolution and average stage of pooling. Among convolution stage with pooling there is the use the activation function. There are many variations of the activation function used in the Neural Network, but in this study the activation function used is Leaky Rectified the Linear Unit (LReLU). Here are the equations used in the LReLU.

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \frac{x_i}{a_i} & \text{if } x_i < 0 \end{cases} \quad (11)$$

Descriptions:

- y_i = The calculation result of the activation function LReLU
- x_i = The value convolution results
- a_i = The value of artificial with a range $[1, \infty]$

On this studies the value that was used is $a_i = 5.5$ referring to research by Bing Xu et al. [10].

2.3.1 CNN Architecture

The CNN architecture used in this study has two stages on convolution and average pooling as shown in Image 8. The image becomes the input image is CNN preprocessing results measuring 28×28 value is at the range of one or zero.

As for the number of neurons in the first convolution, second convolution, and fully connected each neuron is 6 (Index j), 12 neurons (k Index), and the 62 neurons (the index). The matrix of weights that are used for each layer measuring 5×5 convolution, while on a layer of fully connected measuring 192×62 . An array of bias for each layer the same size with the number of neurons in the layer.

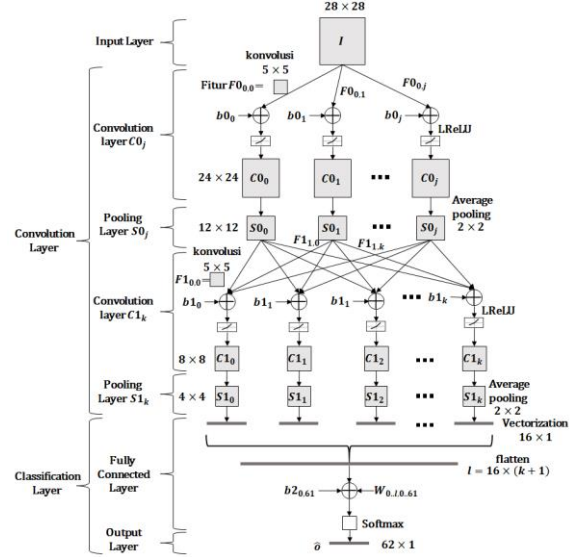


Image 8. CNN Architecture

Convolution performed without using padding. So by using a feature or weights 5×5 would reduce the size of the input image, into a 24×24 in convolution first. At each stage of convolution, performed the shifting feature or weighting as much as one stride.

The results obtained from the convolution calculated using the equation of the function activation LReLU. Next do average-sized 2×2 pooling with a displacement of 2 stride for rows and columns.

After the first stage of the average, the system of pooling would gain matrix size 12×12 and will be the input for the second convolution stages. On the second convolution will be generated a matrix of size 8×8 which will be the input for the second stage average pooling. Then the last stage of the convolution layer, i.e., the average of the second matrix that produces the pooling measures 4×4 .

Error in the calculation method used Cross Entropy Error Function which also serves to reduce the level of error weights and biases to the stage of backpropagation.

2.3.2 Introduction the variables and Initializing

The early stages of CNN is the initialization value weighting matrix of random, and biased with array zero value (0). Here are the variables used in the architecture of CNN.

Table 2. Descriptions Variable on The CNN

Layer	Variable	Descriptions
convolution I	$F0_{0,j,x,y}$	Weights
	$b0_j$	Bias
	$C0_{j,x,y}$	Convolution Result I
Pooling I	$S0_{j,x,y}$	Pooling Result I
Convolution II	$F1_{j,k,x,y}$	Weights
	$b1_k$	Bias

	$C1_{k,x,y}$	Convolution Result II
Pooling II	$S1_{k,x,y}$	Pooling Result II
Fully Connected	$W_{l,m}$	Weights
	$b2_m$	Bias
	o	Fully Connected Result
Output	\hat{o}	Softmax Result

2.3.3 Feedforward CNN

Feedforward is forward flow on Neural Network (NN) that allows the system to run in accordance with the module being used. At this stage carried out several stages which can be seen in diagram block follows.

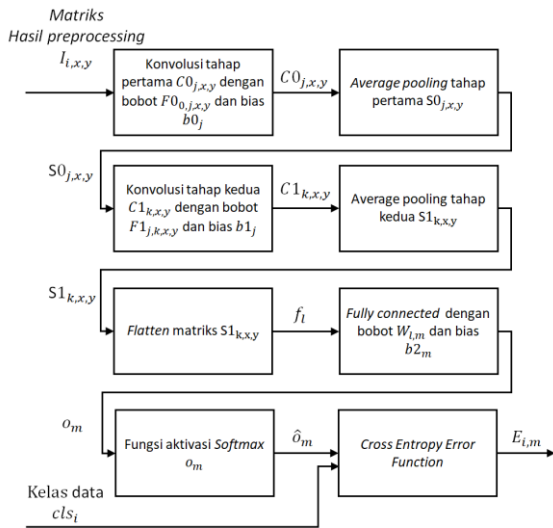


Image 9. Block Diagram of Feedforward

2.3.3.1 First Convolution

First convolution on CNN aims to extract the characteristics of the input image size 28×28 to 24×24 representing by $C0_{j,x,y}$ that use weights $F0_{0,j,x,y}$ size $fx \times fy$ (5×5) and bias $b0_j$ with stride of the weights 1×1 . Next do the calculation of the activation function (ReLU Leaky LReLU).

2.3.3.2 First Average Pooling

First average pooling represent by $S0_{j,x,y}$ with input matrix $C0_{j,x,y}$ from the first convolution with stride $stx \times sty$ (2×2). This process will reduce the matrix from 24×24 to 12×12 .

2.3.3.3 Second Convolution

Second convolution representing by $C1_{k,x,y}$ with input matrix $S0_{j,x,y}$ stride 1×1 . This process will reduce the matrix from 12×12 to 8×8 use the weight $F1_{j,k,x,y}$ size $fx \times fy$ (5×5) and bias $b1_k$. Next do the calculation of the activation function (ReLU Leaky LReLU).

2.3.3.4 Second Average Pooling

Second average pooling represent by $S1_{k,x,y}$. Used stride $stx \times sty$ (2×2) size of matrix 8×8 will reduce to 4×4 on the end of this process.

2.3.3.5 Flatten

Flatten is the stage of incorporation the matrix $S1_{k,x,y}$ size 4×4 changed first into vector so that totaled 16 data value. On the second layer of pooling data pooling there are 12 based on the number of neurons are used, so the neurons at the flatten would amount to 192 neurons are represented by array f_l .

2.3.3.6 Fully Connected and Softmax

Fully Connected (o_m) This research aims to combine each neuron flatten against a number of output neurons, whereas the softmax (\hat{o}_m) aims to quantify the probability of results of feedforward.

2.3.3.7 Cross Entropy Error Function

This method was chosen to calculate and reduce error obtained from the feed forward coaching CNN. This method is not only able to reduce error in value of weight that characterize a class that is being trained, but also change the values for other class [11].

2.3.4 Backpropagation on The CNN

Backpropagation is used as a stage for the reduction of error against each value weights and biases. Here is a diagram of the block of backpropagation.

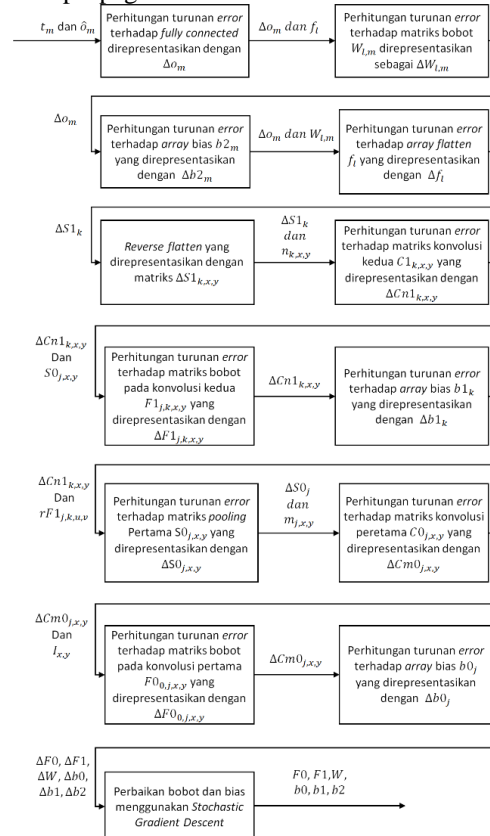


Image 10. Block Diagram of Backpropagation

This stage requires the system to calculate the error of the cross entropy until the earliest of the feedforward process. This stage will result in a value of weights and biases which are updated and will only be done on the training of CNN. Weights and bias that has been fixed will be used for the purposes of testing.

2.3.5 Testing Result of CNN

In the current study used the data as much of the training data and 3.100 15.872 test data that is derived from the NIST datasets with training parameters on table 2.

Table 3. Training Parameters

No	Parameter	Value
1	Epoch	28
2	Target error	0.00000001
3	Learning rate	0.001, 0.002, 0.003, 0.004, 0.005

From the test results obtained for individual learning accuracy rate as follows.

Table 4. Comparison of The Accuracy Of Some Of The Learning Rate

Epoch	Accuration based on Learning Rate				
	0.001	0.002	0.003	0.004	0.005
1	48.71%	46.74%	45.39%	47.06%	43.1%
4	57.39%	57.35%	58.61%	58.87%	54.74%
8	59.03%	59.58%	61.87%	59.19%	59.23%
12	61.19%	62.19%	60.65%	60.94%	58.61%
16	61.81%	61.97%	59.45%	60.35%	56.58%
20	61.03%	61.68%	61.26%	61.52%	58.81%
24	62.39%	62.16%	61.06%	61.45%	56.81%
28	72.48%	66.13%	65.23%	58.94%	57.81%

In table 4, it can be seen that the cell colored keabuan is obtaining the highest accuracy for each group learning rate. Whereas if compared, highest accuracy is obtained by using the learning rate = 0.001 on the 28th epoch.

3. Concluding

Based on the test results, the use of methods of CNN on ICR United Kingdom character by using the dataset obtained the highest accuracy for NIST learning rate = 0.001 on the 28th epoch.

High low accuracy obtained is influenced by a variety of factors that still has not been resolved completely. Things that can be developed for further research is to use different architectures such as the use of different methods, either on CNN or in the

preprocessing to obtained a level of accuracy that is better.

BIBLIOGRAPHY

- [1] R. A. Rahim, L. U. A. Khalik dan M. N. S. Zainudin, "Handwritten English Character Recognition Using Gradient Feature Extraction," International Journal For Advance Research In Engineering and Technology, vol. 3, 2015.
- [2] A. Choudhary, R. Rishi dan S. Ahlawat, "Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique," AASRI Conference on Intelligent Systems and Control, vol. 4, 2013.
- [3] R. K. Mandal dan N. R. Manna, "Handwritten English Character Recognition using Hoof Segmentation of Image Matrix (HSIM)," AMSE Journals, vol. 57, 2014.
- [4] K. S. Younis, "Arabic Handwritten Character Recognition Based On Deep Convolutional Neural Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, 2017.
- [5] M. Zufar dan B. Setiyono, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time," SAINS DAN SENI ITS, vol. 5, p. 6, 2016.
- [6] J. Lemley, S. Abdul-Wahid, D. Banik dan R. Andonie, "Comparison of Recent Machine Learning Techniques for Gender Recognition from Facial Images," MAICS, pp. 97 - 102, 2016.
- [7] NIST, "NIST," NIST, 27 08 2010. [Online]. Available: <https://www.nist.gov/srd/nist-special-database-19>. [Diakses 29 04 2018].
- [8] P. Hidayatullah, Pengolahan Citra Digital Teori dan Aplikasi Nyata, Bandung: Informatika, 2017.
- [9] T. R. Singh, S. Roy, O. I. Singh, T. Sinam dan K. M. Singh, "A New Local Adaptive Thresholding Technique in Binarization," International Journal of Computer Science Issues (IJCSI), vol. 8, no. 6, p. 2, 2011.
- [10] B. Xu, N. Wang, T. Chen dan M. Li, "Empirical Evaluation of Rectified Activations in Convolution Network," arXiv, 2015.
- [11] D. Jurafsky dan J. H. Martin, Speech and Language Processing; An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd penyunt., California: Stanford University, 2018.