JOIN (Jurnal Online Informatika)
Volume 6 No. 2 | December 2021: 128-137
DOI: 10.15575/join.v6i2.653
p-ISSN: 2528-1682
e-ISSN: 2527-9165

Application of VGG Architecture to Detect Korean Syllables Based on Image Text

Irma Amelia Dewi [1],Amelia Shaneva[2]
[1,2]Department of Informatics, Institut Teknologi Nasional Bandung, Indonesia

Article Info

Article history:

Received October 20, 2020
Revised September 06, 2021
Accepted October 13, 2021
Published December 26, 2021

Keywords:

CNN
Image Processing
Korean Language
Korean Syllables
VGG

ABSTRACT

Korean culture began to spread widely throughout the world, ranging from lifestyle, music, food, and drinks, and there are still many exciting things from this Korean culture. One of the interesting things to learn is to know Korean letters (Hangul), which are non-Latin characters. If the Hangul letters have been learned, the next thing that lay people must learn is the Korean syllables, which are different from the Indonesian syllables. Because of the difficulty of learning Korean syllables, understanding a sentence needed a system to recognize Korean syllables. Therefore, in this study designing a system to acknowledge Korean syllables, the method used is Convolutional Neural Network with VGG architecture. The system performs the process of detecting Korean syllables based on models that have been trained using 72 syllable classes. The tests on 72 Korean syllable classes obtain an average accuracy of 96%, an average precision value of 96%, an average recall value of 100%, and an average F1 score of 98%.

Corresponding Author:

Irma Amelia Dewi,
Department of Informatics,
Institut Teknologi Nasional Bandung,
Jl. PH.H. Mustofa No.23, Bandung, Indonesia
Email: irma_amelia@itenas.ac.id

1. INTRODUCTION

Korean culture, commonly known as Hallyu, began to spread in East Asian countries, Southeast Asia, and even worldwide. One of the Southeast Asian countries affected by this Korean cultural phenomenon in Indonesia. Korean culture itself is known in Indonesia, such as dramas, boy bands (male music groups), girl groups (female music groups), traditional clothes and clothing styles, food and drinks, which are the favorites of Indonesian people today [1].

The Korean language also has its characteristics. The distinctive feature of Korean writing is in the language writing system, also known as Hangul. Hangul does not have Latin writing like Indonesian and only in characters (such as 응 which reads yes or eung). Based on the shape of the characters and the way of writing, the Hangul vowel is divided into two groups, namely the standing vowel (vertical) and the sitting vowel (horizontal). The writing method in one syllable is written to the right side (vertical) ㄴ + ㅏ = 나 and the writing method in one syllable is written downward (horizontally) ㅁ + ㅗ = 모. And for beginners who want to learn Korean, after learning vowels and consonants, the next thing that must be learned is Korean syllables.

This study designed a system to recognize syllables in Korean using deep learning technology using a Convolutional Neural Network (CNN). Deep Learning is one of the areas of Machine Learning that utilizes artificial neural networks to implement problems with large datasets. Deep Learning has two of the most popular methods used, specifically Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN is commonly used in image-related processes. CNN has several types of layers that can be used, namely subsampling layer, convolutional layer, loss layer, and fully connected layer. CNN also has several architectures, including AlexNet, GoogleNet, VGG, and others. [2]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual competition held by the ImageNet organization that competes for classifications based on image data available on ImageNet. Every year, various types of CNN architectures become state-of-the-art for classification problems with 1000 classes. With AlexNet as the winner in 2012, ZF Net in 2013, GoogleNet as the first winner,VGG as the second winner

in 2014, and ResNet in 2015 [3]. Based on previous research conducted [4], monitoring of Hangul letters using Convolutional Neural Network by comparing the performance of AlexNet architecture and GoogleNet architecture. The success rate obtained in classifying Hangul letters is 90.12% using the AlexNet architecture and 89.14% using the GoogleNet architecture.

Another study has compared the prediction process of the internal work mechanisms of the AlexNet and VGG architectures through analysis of visual information stored in various layers. This research was conducted by [5], that the VGG architecture has better representation capabilities and can remove unnecessary background information than the AlexNet architecture, which stores unrelated background information that interferes with the final prediction on the last layer.

From the two previous studies, the Visual Geometry Group (VGG) architecture was taken based on the research conducted [5], whether the VGG architecture still produces good performance with different parameters and subjects, bringing the subject of research [4] in the form of Hangul letters.

Therefore, based on the literature review and the background of the problem, in this study, the CNN architecture used to recognize Korean syllables is the VGG architecture. This study aims to measure architectural performance by calculating the value of accuracy, precision, recall, and F1 score to detect Korean syllables based on image text.

2. METHOD

This system development stage implements the agile system development method [6]. This method was chosen to make the system creation process more orderly and controlled according to the schedule. The stage carried out in this development is analysis, system development, system testing stage, and program maintenance stage.

2.1. General Design

The system built in this study is described in Figure 1, and shows how the whole system works simply.

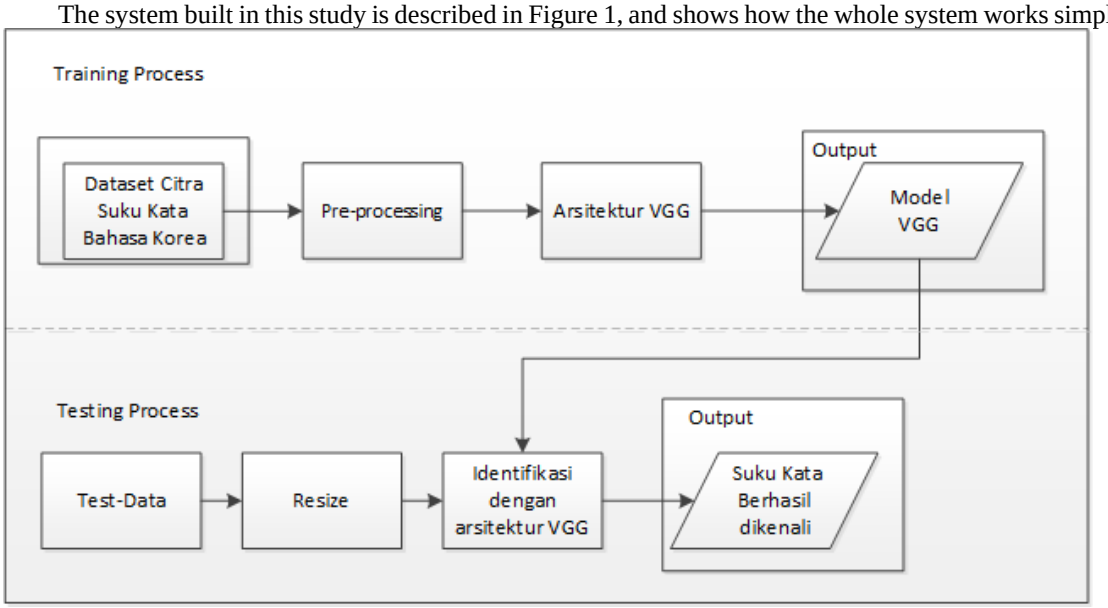

Figure 1. General System Block Diagram

The system built is a syllable identification system in Korean consisting of several steps, specifically the training process and testing process. In the training process, a new model was created with better recognition capabilities based on the VGG architecture of the Hangul image file dataset that has been augmented.

In the syllable identification process, the initial stage is pre-processing by resizing the input image to 224x224. Several architectures carry out the image resolution used to classify ImageNet, especially for the VGG architecture. Graph files and labels generated from the training data process will be entered into the system to recognize syllables properly. If the syllable recognition process is successful, the resulting output will be the detection result of Korean syllables.

2.2. Training Data Collection Process

The training data obtained is a syllable image obtained from the Naver website. From the Naver website, training data is received in image text types for Korean syllables.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Table 1. Table 72 Classes of Korean Syllables

No	Class	No	Class	No	Class	No	Class	No	Class	No	Class	No	Class	No	Class
1	Ba (바)	10	Dae (대)	19	Geu (그)	28	Hi (히)	37	Ji (지)	46	Li (리)	55	Mo (모)	64	Sa (사)
2	Beu (브)	11	Deo (더)	20	Gi (기)	29	Hing (힝)	38	Jjae (째)	47	Lo (로)	56	Mu (무)	65	Seo (서)
3	Beo (버)	12	Deu (도)	21	Go (고)	30	Ho (호)	39	Jo (조)	48	Lu (루)	57	Na (나)	66	Seu (스)
4	Bi (비)	13	Di (디)	22	Goj (곶)	31	Hu (후)	40	Ju (주)	49	Ma (마)	58	Neo (너)	67	Si (시)
5	Bis (빗)	14	Do (도)	23	Gu (구)	32	I (이)	41	Kkyu (뀨)	50	Maess (맸)	59	Neu (느)	68	So (소)
6	Bo (보)	15	Du (두)	24	Gug (국)	33	Ja (자)	42	Kyeo (켜)	51	Mang (망)	60	Ni (니)	69	Su (수)
7	Bu (부)	16	Ga (가)	25	Ha (하)	34	Jad (잗)	43	La (라)	52	Meo (머)	61	No (노)	70	Teo (터)
8	Cho (초)	17	Gass (갔)	26	Heo (허)	35	Jeo (저)	44	Leo (러)	53	Meu (므)	62	Nu (누)	71	U (우)
9	Da (다)	18	Geo (거)	27	Heu (흐)	36	Jeu (즈)	45	Leu (르)	54	Mi (미)	63	Pya (퍄)	72	Yu (유)

This study uses 30 fonts that have been downloaded on the Naver website. The overall training image used is 2160, with the total number of classes used are 72 classes.

2.3. Pre-processing

There is a pre-processing process in the training and testing process, along with a pre-processing flowchart shown in Figure 2.

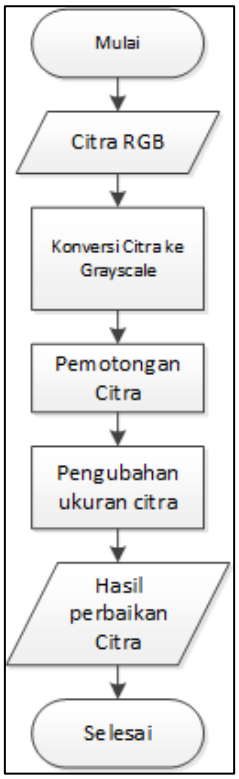

Figure 2. Pre-processing Flowchart

Furthermore, it is explained how the system that has been made on the flowchart works as follows:

1. RGB Image Detection.
2. Grayscaling. This stage is the initial process in the image pre-processing process. At this stage the input image is converted into a grayscale image using Equation 1 [7] so that the image has an intensity of gray pixel values (0 -255).

$$Grayscale = 0.21 * R + 0.72 * G + 0.07 * B \quad (1)$$

Where R is the intensity of the pixel is red, G is the intensity of the pixel in green, and B is the intensity of the pixel in blue. In order to perform the grayscale calculation phase, an input image is required. Assume that the input image is class Ba which can be seen in Figure 3, and is given a calculation case study by sampling an image size 10x10 from its original size, as in Table 2.

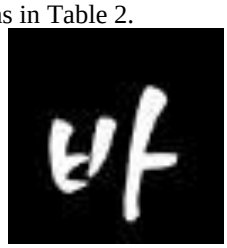

Figure 3. Image of the Ba Syllable [8]

Table 2. Table RGB Image Matrix

R=184	R=250	R=247	R=255	R=232	R=255	R=255	R=129	R=0	R=0
G=184	G=250	G=247	G=255	G=232	G=255	G=255	G=129	G=0	G=0
B=184	B=250	B=247	B=255	B=232	B=255	B=255	B=129	B=0	B=0
R=215	R=255	R=255	R=240	R=255	R=241	R=213	R=12	R=5	R=6
G=215	G=255	G=255	G=240	G=255	G=241	G=213	G=12	G=5	G=6
B=215	B=255	B=255	B=240	B=255	B=241	B=213	B=12	B=5	B=6
R=247	R=255	R=249	R=255	R=250	R=238	R=44	R=4	R=0	R=6
G=247	G=255	G=249	G=255	G=250	G=238	G=44	G=4	G=0	G=6
B=247	B=255	B=249	B=255	B=250	B=238	B=44	B=4	B=0	B=6
R=255	R=250	R=255	R=255	R=250	R=165	R=0	R=16	R=4	R=0
G=255	G=250	G=255	G=255	G=250	G=165	G=0	G=16	G=4	G=0
B=255	B=250	B=255	B=255	B=250	B=165	B=0	B=16	B=4	B=0
R=255	R=251	R=255	R=238	R=255	R=85	R=6	R=0	R=2	R=0
G=255	G=251	G=255	G=238	G=255	G=85	G=6	G=0	G=2	G=0
B=255	B=251	B=255	B=238	B=255	B=85	B=6	B=0	B=2	B=0
R=254	R=255	R=254	R=255	R=235	R=44	R=0	R=0	R=0	R=6
G=254	G=255	G=254	G=255	G=235	G=44	G=0	G=0	G=0	G=6
B=254	B=255	B=254	B=255	B=235	B=44	B=0	B=0	B=0	B=6
R=249	R=255	R=246	R=255	R=155	R=0	R=0	R=0	R=8	R=0
G=249	G=255	G=246	G=255	G=155	G=0	G=0	G=0	G=8	G=0
B=249	B=255	B=246	B=255	B=155	B=0	B=0	B=0	B=8	B=0
R=255	R=253	R=255	R=246	R=27	R=4	R=0	R=6	R=2	R=3
G=255	G=253	G=255	G=246	G=27	G=4	G=0	G=6	G=2	G=3
B=255	B=253	B=255	B=246	B=27	B=4	B=0	B=6	B=2	B=3
R=251	R=255	R=255	R=236	R=0	R=1	R=0	R=0	R=0	R=0
G=251	G=255	G=255	G=236	G=0	G=1	G=0	G=0	G=0	G=0
B=251	B=255	B=255	B=236	B=0	B=1	B=0	B=0	B=0	B=0
R=253	R=255	R=251	R=144	R=1	R=3	R=4	R=4	R=12	R=7
G=253	G=255	G=251	G=144	G=1	G=3	G=4	G=4	G=12	G=7
B=253	B=255	B=251	B=144	B=1	B=3	B=4	B=4	B=12	B=7

In Table 2 the coordinates (0,0) have a pixel value of R = 184, G = 184, B = 184. By using Equation 1, the results of the grayscale image calculation are as follows:

Pixel count (0.0) :

Grayscale = 0,21 * 184 + 0,72 * 184 + 0.07 * 184

Grayscale = 38,64 + 132,48 + 12,88

Grayscale = 184

p-ISSN: 2528-1682
e-ISSN: 2527-9165

The calculation process is carried out up to coordinates (10,10), resulting in a grayscale image with grayscale pixel values, as shown in Figure 4.

184	250	247	255	232	255	255	129	0	0
215	255	255	240	255	241	213	12	5	6
247	255	249	255	250	238	44	4	0	6
255	250	255	255	250	165	0	16	4	0
255	251	255	238	255	85	6	0	2	0
254	255	254	255	235	44	0	0	0	6
249	255	246	255	155	0	0	0	8	0
255	253	255	246	27	4	0	6	2	3
251	255	255	236	0	1	0	0	0	0
253	255	251	144	1	3	4	4	12	7

Figure 4. Grayscale Image Matrix

3. Resizing. This stage is done by changing the size of the test image according to the training data, which is 96x96 pixels.

2.4. VGG Process

The results of the pre-processing process are then processed in the VGG architecture. VGG Network is a CNN architecture designed by [9]. This architecture was created to participate in the 2014 ImageNet Challenge competition and successfully achieved top localization and classification rankings. The input used is an RGB image measuring 224 × 224 pixels. There are two types of Convolutional layers used in this architecture, specifically the Convolutional Layer with a filter size of 3x3 (conv3) and a filter size of 1x1 (conv1). There are various sizes of Convolutional Layer used, specifically 64x64, 128x128, 256x256, and 512x512.

VGG-16 consists of 13 Convolution Layer, 5 Maxpooling Layer, and 3 Dense Layers, which produces 21 but only 16 weight layers [9]. Figure 5 shows a flowchart of the VGG-16 architecture. The training process was carried out with three different epochs, which are 10, 50, and 100.

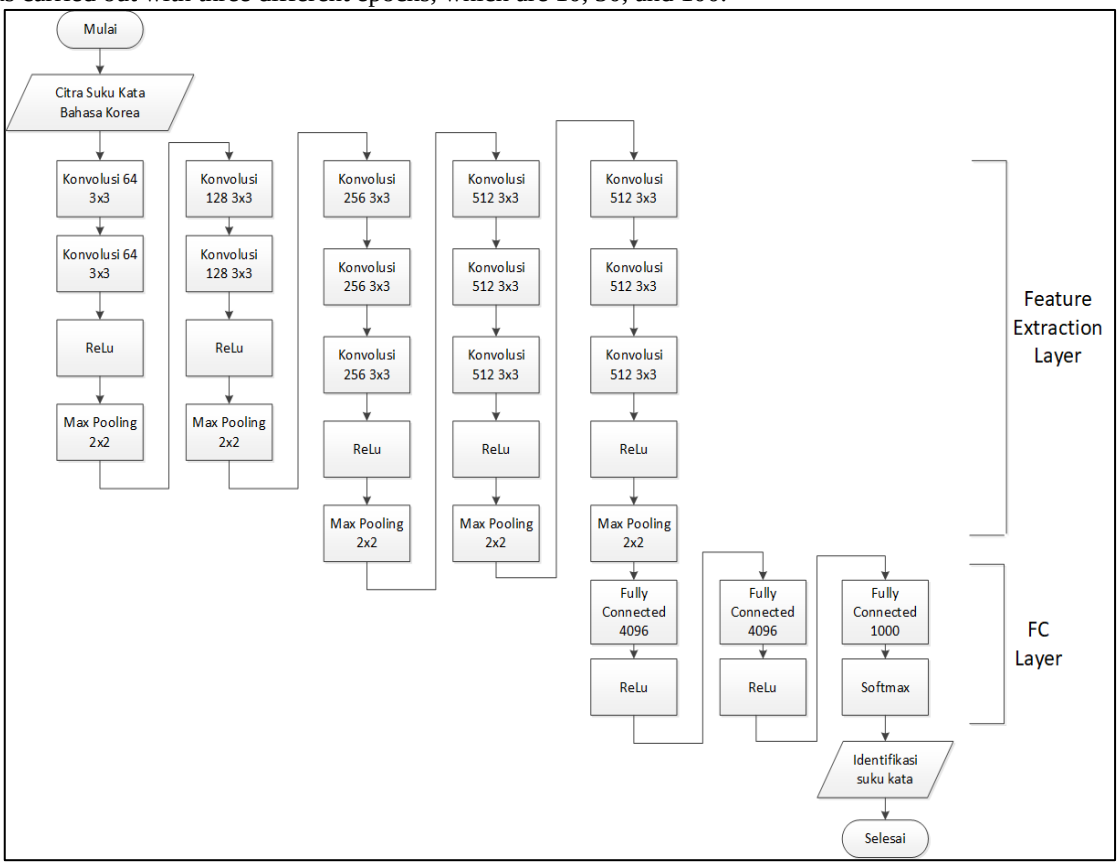

Figure 5. VGG-16 Architecture Flowchart

Furthermore, it is explained how the system that has been made on the flowchart works as follows:
1. Enter the image of Korean syllables that have gone through the pre-processing process.
2. The convolution process is carried out on the image of the Korean syllable against the kernel to get features or filters on the image. The first group starts with 64 filters. As in the case of grayscale in the Preprocessing chapter, the results of the class Ba matrix with a 10x10 matrix are taken and shown in Figure 4. A zero padding process is carried out from the matrix by adding a value of 0 on each side of the matrix. In Figure 6, the matrix of the results of the zero padding is shown.

0	0	0	0	0	0	0	0	0	0	0	0
0	184	250	247	255	232	255	255	129	0	0	0
0	215	255	255	240	255	241	213	12	5	6	0
0	247	255	249	255	250	238	44	4	0	6	0
0	255	250	255	255	250	165	0	16	4	0	0
0	255	251	255	238	255	85	6	0	2	0	0
0	254	255	254	255	235	44	0	0	0	6	0
0	249	255	246	255	155	0	0	0	8	0	0
0	255	253	255	246	27	4	0	6	2	3	0
0	251	255	255	236	0	1	0	0	0	0	0
0	253	255	251	144	1	3	4	4	12	7	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. Zero Padding Result Matrix

The convolution process is carried out using the 3x3 kernel, as shown in Figure 7.

0	-1	0
-1	4	-1
0	-1	0

Figure 7. 3x3 Matrix

To find out the results of the convolution operation, here are the steps to multiply the convolution process between the 10x10 matrix in Figure 6 with the 3x3 filter in Figure 7 :
a) To get the value in row 1 column 1 in Figure 11, add (0 * 0) + (0 * -1) + (0 * 0) + (0 * -1) + (184 * 4) + (250 * -1) + (0 * 0) + (215 * -1) + (255 * 0) = 271. This process is shown in Figure 8. The red box in Figure 8 is multiplied against the 3x3 filter next to it.

1

0	0	0	0	0	0	0	0	0	0	0	0
0	184	250	247	255	232	255	255	129	0	0	0
0	215	255	255	240	255	241	213	12	5	6	0
0	247	255	249	255	250	238	44	4	0	6	0
0	255	250	255	255	250	165	0	16	4	0	0
0	255	251	255	238	255	85	6	0	2	0	0
0	254	255	254	255	235	44	0	0	0	6	0
0	249	255	246	255	155	0	0	0	8	0	0
0	255	253	255	246	27	4	0	6	2	3	0
0	251	255	255	236	0	1	0	0	0	0	0
0	253	255	251	144	1	3	4	4	12	7	0
0	0	0	0	0	0	0	0	0	0	0	0

*

0	-1	0
-1	4	-1
0	-1	0

Figure 8. Illustration of Convolution Process Step 1

b) To get the value in row 1 column 2 in Figure 11, add (0 * 0) + (0 * -1) + (0 * 0) + (250 * -1) + (247 * 4) + (255 * -1) + (255 * 0) + (255 * -1) + (240 * 0) = 336. This calculation process is shown in Figure 9. The red square in Figure 9 is multiplied by the 3x3 matrix next to it. When compared with Figure 8, it can be seen that a red box shift of 2 strides or 2 columns on the 10x10 matrix. [10]

p-ISSN: 2528-1682
e-ISSN: 2527-9165

0	0	0	0	0	0	0	0	0	0	0	0
0	184	250	247	255	232	255	255	129	0	0	0
0	215	255	255	240	255	241	213	12	5	6	0
0	247	255	249	255	250	238	44	4	0	6	0
0	255	250	255	255	250	165	0	16	4	0	0
0	255	251	255	238	255	85	6	0	2	0	0
0	254	255	254	255	235	44	0	0	0	6	0
0	249	255	246	255	155	0	0	0	8	0	0
0	255	253	255	246	27	4	0	6	2	3	0
0	251	255	255	236	0	1	0	0	0	0	0
0	253	255	251	144	1	3	4	4	12	7	0
0	0	0	0	0	0	0	0	0	0	0	0

✱

0	-1	0
-1	4	-1
0	-1	0

Figure 9. Illustration of Convolution Process Step 2

c) To get the values in row 1 of column 3, 4, 5 and 6, do the same with step 2 with a shift of 2 strides.
d) To get the matrix value on row 2 column 1 in Figure 11, add (0 * 0) + (215 * -1) + (255 * 0) + (0 * -1) + (247 * 4) + (255 * -1) + (0 * 0) + (255 * -1) + (250 * 0) = 371. This calculation process is shown in Figure 10. The red square in Figure 10 is multiplied by the adjacent 3x3 matrix. If it is compared with Figure 9, it can be seen that 2 strides of the red box are shifting down on the 10x10 matrix.

0	0	0	0	0	0	0	0	0	0	0	0
0	184	250	247	255	232	255	255	129	0	0	0
0	215	255	255	240	255	241	213	12	5	6	0
0	247	255	249	255	250	238	44	4	0	6	0
0	255	250	255	255	250	165	0	16	4	0	0
0	255	251	255	238	255	85	6	0	2	0	0
0	254	255	254	255	235	44	0	0	0	6	0
0	249	255	246	255	155	0	0	0	8	0	0
0	255	253	255	246	27	4	0	6	2	3	0
0	251	255	255	236	0	1	0	0	0	0	0
0	253	255	251	144	1	3	4	4	12	7	0
0	0	0	0	0	0	0	0	0	0	0	0

✱

0	-1	0
-1	4	-1
0	-1	0

Figure 10. Illustration of Convolution Process Step 3

e) To get the value in row 2, columns 2, 3, 4, 5, and 6, shift two strides to the last position of the red box in Figure 12, as done in step 2.
f) To get the values in rows 3 and 4, do as in steps 4 and 5.
g) The result of the 10x10 matrix convolution process against a 3x3 filter of 2 strides produces a 6x6 matrix output as in Figure 11.

271	336	163	423	-130	0
371	-24	2	-279	-19	-6
260	22	212	-61	4	0
232	-35	103	0	30	0
241	23	-265	-5	-14	0
-253	-251	-1	-4	-12	0

Figure 11. Matrix of Convolution Operation Results with a 3x3 Filter of 2 strides

3. The activation function uses ReLU or Rectified Linear Unit, which is the activation layer on CNN by applying the function f (x) = max (0, x), if x ≤ 0 then x = 0 and if x> 0 then x = x. This process is carried out for thresholding with a zero value against the pixel value in the image input [11]. The ReLU activation process is carried out on each matrix element from the results of the 10x10 matrix convolution process shown in Figure 11 to produce the matrix in Figure 12.

271	336	163	423	0	0
371	0	2	0	0	0
260	22	212	0	4	0
232	0	103	0	30	0
241	23	0	0	0	0
0	0	0	0	0	0

Figure 12. ReLu Operation Result Matrix

4. Reducing the size of the matrix that has been obtained during convolution by using max pooling [12] by taking a maximum value of 2x2 2 strides is carried out on the matrix of Figure 12, which is shown in Figure 13 by taking the largest value in each red box.

271	336	163	423	0	0
371	0	2	0	0	0
260	22	212	0	4	0
232	0	103	0	30	0
241	23	0	0	0	0
0	0	0	0	0	0

Figure 13. Max Pooling Operation

The results of the max pooling operation are shown in Figure 14.

371	423	0
260	212	30
241	0	0

Figure 14. Result of Max Pooling Operation

5. The convolution process in the second group starts with 128 filters using a 3x3 size. This process is carried out twice. The calculation process is the same as in point 2. b.
6. The activation function uses ReLU. The calculation process is the same as point 2.c.
7. Reducing the size of the matrix by using max-pooling of 2x2. The calculation process is the same as in point 2.d.
8. The convolution process in the third group begins with the number of filters 256 using a size of 3x3. This process is carried out twice. The calculation process is the same as in point 2.b.
9. The activation function uses ReLU. The calculation process is the same as in point 2.c.
10. Reducing the size of the matrix by using max-pooling of 2x2. The calculation process is the same as in point 2.d.
11. The convolution process in the fourth and fifth groups begins with the number of filters 512 using a size of 3x3. This process was carried out twice in the fourth and fifth groups. The calculation process is the same as in point 2.b.
12. The activation function uses ReLU. The calculation process is the same as in point 2.c.
13. Reducing the size of the matrix by using max-pooling of 2x2. The calculation process is the same as in point 2.d.
14. After the Korean syllable image passes feature extraction, the fully connected layer [13] process is carried out. The reduced image at max pooling is converted into one dimension resulting in a map feature. The process of matching test data to training data is carried out so that the data can be classified linearly with the classification process that occurs. The image is converted into a vector measuring 4096 × 1. This process is carried out again after activating ReLu again.
15. After the fully connected 4096 processes and the second ReLu activation, the final fully connected layer process is carried out by converting the image into 1000 × 1 vector.
16. Activate Softmax [14] to handle multiclass classification cases because usually the output layer has more than one neuron. In this softmax section, an image classification process is carried out on the label of the Korean syllable, where the probability calculation of the similarity of the test image and the training image is carried out. The Softmax function is able to convert log [2.0, 1.0, 0.1] into probability [0.7, 0.2, 0.1], and the probability is 1.
17. Identified Korean syllables.

3. RESULTS AND DISCUSSION

3.1. Testing Training Set

Training set testing is done by separating the dataset with a ratio of 80:20, where 80% is for the train image and 20% is for the validation image. Validation Set or Validation Image is a set of data used to train artificial intelligence (AI) to find and optimize the best model to solve a given problem. The training process is carried out with epoch, learning rate, batch size, image dimensions, and different optimizers. What is measured for this study is the shape of the Hangul letter, epoch, learning rate, batch size, image dimensions, and the optimizer.

For case (I), the training process is carried out as many as 100 epochs with a period of about 1382s every epoch, and in one epoch, it reaches 720 steps, so that it takes three days for the training process. With Adam's optimizer, the batch size is 32, the learning rate is 1e-3 (0.001) and the dimensions of Figure (96, 96, 3). From the training results obtained using VGG, the loss and accuracy values are obtained. The loss value is

used to compare and measure the prediction results and estimate errors. The accuracy percentage the case 1 is 97%, val accuracy 98%, loss as much as 8%, val loss is 3%, and a span of 2 seconds to perform each step.

For case (II), the training process is carried out as many as ten epochs with a period of about 184s every epoch, and in one epoch, it reaches 360 steps. With the RMSProp optimizer, the batch size is 64. The learning rate is 1e-1 (0.1) and the dimensions of Images (32, 32, 3). The accuracy percentage for case II is 68%, val accuracy 79%, loss 1.11, val loss 2.57 and a period of 511 ms to perform each step.

For case (III), the training process was carried out as many as 50 epochs with a period of about 323s every epoch, and in one epoch it reached 180 steps. With the Adagrad optimizer, the batch size is 128, learning rate 1e-2 (0.01) and Image dimensions (64, 64, 3). The accuracy percentage for case III is 91%, Val accuracy as much as 96%, loss as much as 29%, val loss as much as 13%, and a period of 2 seconds to perform each step.

For the case study (IV) the training process is carried out as many as 10 epochs with a period of about 154s every epoch and in one epoch it reaches 360 steps. With the SGD optimizer, the batch size is 64, the learning rate is 1e-4 (0.0001), and the dimensions of Images (32, 32, 3). The percentage for case IV is the accuracy of 0.02, val accuracy of 0.03, loss of 5.67, val loss of 4.30, and a span of 428ms to perform each step.

3.2. System Performance Testing

Measure the system's work and test by calculating the value of accuracy, precision, recall,nd F1 score [15]. The system can recognize the syllables obtained using the equation for the accuracy, precision, recall, and F1 score.

$$Accuracy = y = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} x\ 100\% \quad (3)$$

$$Recall = \frac{TP}{TP+FN} x\ 100\% \quad (4)$$

$$F1\ Score = 2\ x\ \frac{(Presisi\ x\ Recall)}{(Presisi+Recall)} \quad (5)$$

Where :
TP (True Positive), namely the number of positive data classified correctly by the system.
TN (True Negative), which is the amount of negative data that is classified correctly by the system.
FN (False Negative), which is the amount of negative data but is classified incorrectly by the system.
FP (False Positive), which is the number of positive data but is classified incorrectly by the system.

After getting the accuracy, precision, recall, and F1 score, what needs to be done is to calculate the average value with the following equation:

$$\bar{x} = \frac{x1+x2+\cdots+x_n}{n} \quad (6)$$

Where :
$\bar{x}$ is the calculated average.
$x1, x2 \ldots x_n$ is the amount of data.
n is total of data.

3.2. Testing Analysis

In the testing process, tested 72 classes shown in table 1, where the tests were carried out to produce accuracy, precision, recall, and F1 score. From the 4 cases, the total mean value for accuracy using Equation 2, precision using Equation 3, recall using Equation 4, and F1 score using Equation 5 was sought using Equation 6. From this equation, for the case, I obtained an average value of total accuracy of 96%, an average value of total precision of 96%, an average value of total recall of 100%, and a total value of an F1 score of 98%. For case II, it was obtained an average total accuracy value of 84%, an average total precision value of 84%, an average total recall value of 100%, and a total value of an F1 score of 91%. For case III, the mean value of total accuracy is 93%, the average value of total precision is 93%, the average value of total recall is 100%, and the total value of the F1 score is 96. For case IV, the mean value is obtained. The average total accuracy is 4%, the total precision average value is 4%, the total recall value is 4%, and the F1 score is 7%.

Table 3. System Performance Testing Table

No	Testing	Result			
		Case I	Case II	Case III	Case IV
1	Accuracy	96%	84%	93%	4%
2	Precision	96%	84%	93%	4%
3	Recall	100%	100%	100%	44%
4	F1 Score	98%	91%	96%	7%

Based on the four cases, the case (I) with Adam's optimizer, a batch size of 32, a learning rate of 1e-3 (0.001) and the dimensions of Image (96, 96, 3) resulted in a greater average value than the 3 cases. The remaining percentage of failure is obtained because some syllables cannot be converted according to their respective labels because the type and size of the syllables used as input are different from the template. On the other hand, the lack of training data representing the same syllables causes some syllables to be converted into the wrong label.

4. CONCLUSION

The conclusion that can be drawn from this research is based on testing on four cases, implementing the Convolutional Neural Network method using the VGG-16 architecture with Adam's optimizer, batch size of 32, learning rate 1e-3, and image dimensions (96, 96, 3). The percentage is quite good, with an average value of total accuracy of 96%, an average value of total precision of 96%, an average value of total recall of 100%, and a total value of an F1 score of 98%. Of the 72 classes, almost all have a match with the existing data. There is no underfitting or overfitting due to the balanced condition between testing data tests and training testing.

5. REFERENCES

[1] RAHOTNI SITIO, "FENOMENA HALLYU PADA KOMUNITAS KOREAN CULTURAL CENTRE MEDAN (KCCM) DI KOTA MEDAN," *Universitas Negeri Medan*, 2017.

[2] M. Yulius Harjoseputro S.T., "CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK PENGKLASIFIKASIAN AKSARA JAWA," *Universitas Atma Jaya Yogyakarta*, 2018.

[3] Olga Russakovsky, J. Deng., Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 2014.

[4] Sang-Geol Lee, Yunsick Sung, Yeon-Gyu Kim and Eui-Young Cha, "Variations of AlexNet and GoogLeNet to Improve Korean Character Recognition Performance," *Journal Of Information Processing Systems*, 2018.

[5] Wei Yu, Kuiyuan Yang, Yalong Bai, Tianjun Xiao, Hongxun Yao and Yong Rui, "Visualizing and Comparing AlexNet and VGG using Deconvolutional Layers," *arXiv:1412.6631*, 2014.

[6] K. Andri, Analisa Sistem Informasi, Yogyakarta: Graha Ilmu, 2004.

[7] G. T. Situmorang, A. W. Widodo and M. A. Rahman, "Penerapan Metode Gray Level Cooccurence Matrix (GLCM) untuk Ekstraksi Ciri pada Telapak Tangan," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2019.

[8] A. Hwa, C. Yong, R. Adinda, S. Agung and F. Hutagalung, "Bahasa Korea Terpadu untuk Orang Indonesia," *Yu Hyun-seok: Seoul*, 2013.

[9] Karen Simonyan and Andrew Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," *University of Oxford*, 2014.

[10] I. Suartika E. P, A. Wijaya and Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101," *Jurnal Teknik ITS*, 2016.

[11] D. Pedamonti, Comparison of non-linear activation functions for deep neural networks on MNIST Classification Task, Machine Learning, 2018.

[12] D. Nielsen, "Deep Learning Cage Match: Max Pooling vs Convolutions," 9 September 2018. [Online]. Available: https://medium.com/@duanenielsen/deep-learning-cage-match-max-pooling-vs-convolutions-e42581387cb9.

[13] Christopher Albert Lorentius, Rudy Adipranata and Alvin Tjondrowiguno, "Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network," *Universitas Kristen Petra*, 2019.

[14] A. T. Wicaksono, "Prediksi Kepribadian Berdasarkan Tulisan Tangan Dengan Metode Convolutional Neural Network," *Universitas Komputer Indonesia*, 2019.

[15] C. D. Manning, P. Raghavan and H. Schütze, An Introduction to Information Retrieval, Cambridge: Cambride University Press, 2009.