International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 4
ISSN: 2088-5334

Implementation of Gabor Wavelet and Support Vector Machine for Braille Recognition

Rifki Indra Perwira [a,*], Mangaras Yanu Florestiyanto [a], Indah Reforsiana Nurjanah [a], Heriyanto [a], Dessyanto Boedi Prasetyo [a]

[a] *Faculty of Industrial Engineering, Universitas Pembangunan Nasional Veteran Yogyakarta, Babarsari, Yogyakarta,55281, Indonesia*
*Corresponding author: *rifki@upnyk.ac.id*

***Abstract*— Visually impaired people use braille letters to write and read, which not all people with normal vision can read. It causes parents of children with impaired vision to have difficulties in assisting them in learning at home. At the same time, the involvement of parents in children's learning assistance is needed to monitor their learning progress. In this research, braille letters are identified through images taken using a scanner as a tool to input the images. Then, the Canny edge detection method is used obtain all the edges of each braille dot. Feature extraction is applied to obtain all characteristics of each letter, and the method used is the Gabor Wavelet. The features which are utilized include standard deviation, mean, variance, and median with a theta angle of 0^0, 30^0, 45^0, 60^0, 120^0, 135^0, 180^0 and wavelengths of 3, 6, 13, 28, and 58. These features are combined and used as test data and training data for the Support Vector Machine (SVM) classification stage and produce letters and words in alphabetic letter forms. Braille letters that can be identified in this research are small letters, capital letters, punctuation marks, and numbers. Tests are carried out using a multi-class confusion matrix scenario to determine the level of accuracy, precision, and recall. Based on the results of the tests conducted using 758 braille letters, the accuracy value is 98.15%; the precision value is 97.66%; and the recall value is 98.28%.**

***Keywords*— Braille; image processing; canny edge detection; gabor wavelet; support vector machine.**

Manuscript received 9 Feb. 2021; revised 6 May 2021; accepted 29 Jun. 2021. Date of publication 31 Aug. 2022.

I. Introduction

People with normal eyesight exchange information in visual and printed form. It is different from someone who has visual deficiencies or is blind, causing problems in exchanging information in written form. A visually impaired person should be assisted by their family, in which parental involvement in the learning process also contributes to learning outcomes [1]. Learning is carried out not only at school but also at home, in which parents play a crucial role in helping children understand their learning. A visually impaired person uses braille to write and read. This causes parents to have difficulties assisting children with visual impairments in learning.

Up to this point, the braille script has been translated by matching each braille letter with the alphabet until it is readable. For people who are not fluent in braille, this method is ineffective. Therefore, it is necessary to develop an intelligent system that helps ordinary people to recognize braille easily. Braille is a touch script consisting of raised dots that indicate letters, numbers, and symbols. For visually impaired people, reading braille is done by using the sense of touch by fingertips [2].

This research aims to build a braille identification system. In research by Smelyakov *et al* [3] revealed that the test accuracy of applying the neural network method in identifying Braille characters is 95%. Mousa *et al.* [4] show that the accuracy value obtained is between 94% and 99% in producing text in alphabetical letters. Then, in research by Antonscopoulous and Bridson [5], the average accuracy value obtained is between 94.9% in recognizing a one-sided braille text. Another research uses the contour method with an image slope of 0 degrees to 0.5 degrees [6]. The image classification using the Support Vector Machine (SVM) method for Braille recognition reaches 97,44%ccuracy [7]. Other previous research discussed braille recognition with different methods and extraction features [8]- [13].

The best combination comes from HoG algorithm and SVM with an accuracy of 94.43% in character recognition [14] and is also used for different objects [15], [16]. The Gabor Wavelet extracted features combined with the PCA method to recognize the traffic sign using parameters of 5 scale, and eight orientation gets 95% [17]. Other previous studies using Gabor Wavelet to identify facial images and classified by KNN method obtain 100% accuracy depending on the quality of the braille document [18].

Based on several studies that have been conducted, there are three main contributions of this study. This study proposes Gabor Wavelet as a feature extractor to enhance the recognition of braille images. Also, this study aims to prove that the Support Vector Machine (SVM) with Gabor Wavelet can produce high accuracy of braille recognition. In addition, this study is subject to prove that the SVM with Gabor Wavelet can be used to build a braille recognition system with high accuracy.

II. Material and Method

This study is developed in several stages shown in Fig 1.

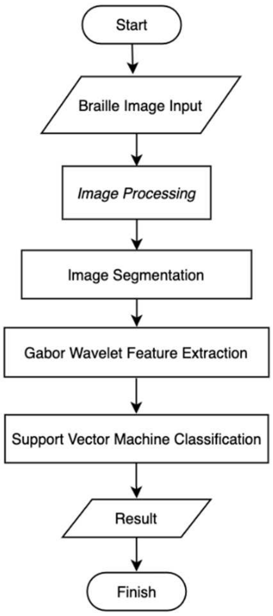

Fig. 1 Research stage

The first stage is inputting process of the braille letter image using a scanner. Afterwards, the images are saved in .jpg, .bmp, and .png formats. The images used in this study are braille images in the word forms of uppercase and lowercase letters, punctuation, and numbers. The braille image processing is subsequently carried out aiming to improve the images and prepare them for the image segmentation process. In the image segmentation process, the images of words will be separated into the images of each letter.

After that, images of each letter will be examined with a feature extraction process using the Gabor Wavelet. This result is saved into training data and test data. The next process is classifying images based on the feature extraction data on the training and test data set. Both data will be processed and produce class identification according to the test data using the Support Vector Machine method. The classes are combined to form a word according to the braille images.

A. Image Processing

Image preprocessing stages include several processes, seen in the following:

1) Grayscale: Grayscale is a process to change the input images into gray-level ones. Each image pixel entered initially has RGB values, then the average value is calculated so that the color pixel value changes and becomes a gray value [19].

2) Median Filter: Median Filter is a process that aims to remove noise in the previous grayscale images. A median value is obtained by calculating the median value of each mask [20]. The median filter is done by replacing the noisy pixel with the median value of the neighbor pixels.

3) Binary Image: Binary Image processing is a process for converting a gray image into a black and white image [21]. The pixel value is changed based on the threshold value, that is, by calculating the average value of the degrees of gray. If the pixel value is greater than the threshold value, it will be converted to 255 (white). If it is smaller than the threshold value, it will be converted to a value of 0 or black [22]. Binary images are used to identify the existence of objects that are represented as regions in the image.

4) Dilation: Dilation is a process used to add pixels to the boundaries of the image. A braille image makes the black dots smaller [23]. It is a process of combining image points that intersect with the element structure and finding the maximum pixel value in the element structure environment [24]. It aims to remove some tiny dots or noise so that the braille images only contain black dots letters and reduce the risk of reading non-braille dots.

5) Erosion: Erosion is a process used to thicken black pixels or remove pixels to the boundaries of a braille image. It will return the braille dot size to its original size. This morphological process uses a small (radius of 1) disk-shaped structure element. It aims to reduce the false positive spots and away extrusions from the spine region [25].

6) Canny Edge Detection: The Canny Edge Detection method is used to extract the edges of objects in images. It is an image processing technique to detect brightness discontinuities, extracting large gradient brightness magnitudes [26]. This method yields the best result than other edge detection methods because it can effectively suppress the noise and detect it well [27].

7) Contour Normalization: Contour Normalization is needed because image contours on the system have different sizes and shapes. In this process, they will be converted into similar shapes and sizes. It consists of several processes. The first is detecting the edges of every dot of the images. This Canny edge detection process aims to mark out the essential input images and produce the circular edges of the braille letters. After that, every contour is figured out to calculate the center point of each contour.

From the center point coordinates, the shape and size of each contour will be equalized by drawing a circle at each dot with the same radius. The next stage is looking for the contour area of the image. This sub-process will crop the image only on the braille texture area. The previous process produces a photo with a black background and white braille dots. Then it inverses the image color to restore it to the original image color.

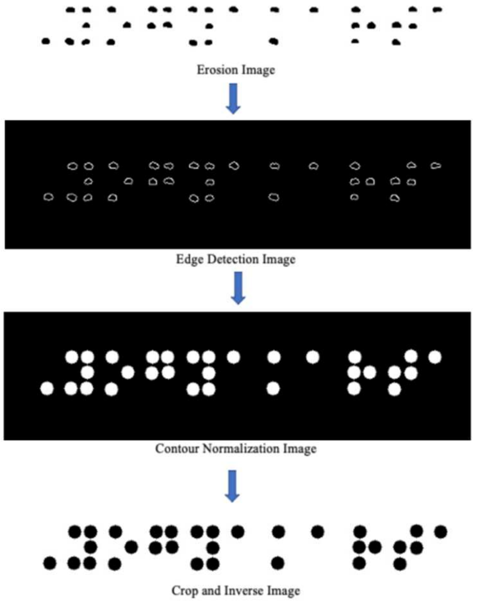

Fig. 2 Contour normalization process

B. Image Segmentation

At this stage, images are grouped based on the letters and are segmented, resulting in a picture for each letter for the feature extraction process. The image segmentation process is carried out in several stages. The first stage is thickening each braille dot by using the erosion process with a kernel size of 100x7 pixels so that the dots of each braille letter that are close will merge into one form of a black segment. The black elements indicate letters, and the white elements indicate blank spaces between letters. The next stage is detecting the edge to find the edge of the line formed on each black element and finding the coordinate value from the edge detection image. Then, the subsequent stage projects it on the previous image and crops it according to the formed line.

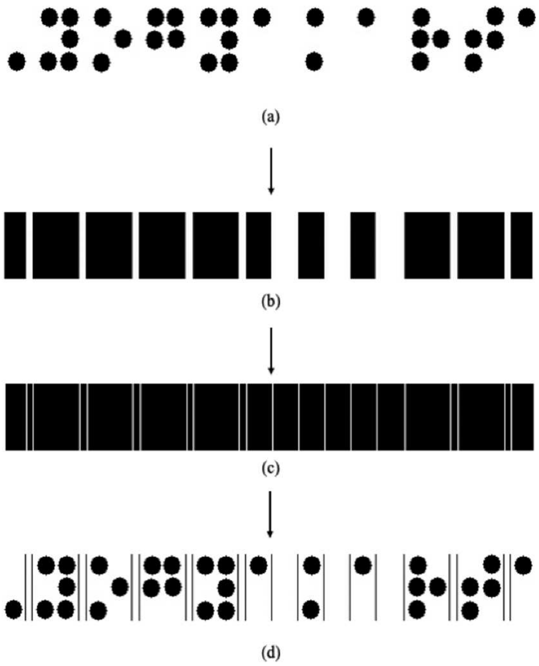

Fig. 3 Segmentation Image process (a) Contour normalization image (b) Erosion for each contour (c) Edge detection image (d) Braille segmentation image

C. Gabor Wavelet Feature Extraction

Feature extraction is a process of retrieving important information in an image. Feature extraction aims to take the specific characteristics of an object for storage. These characteristics are used for comparison in identifying certain objects in an image. This feature extraction is used as a parameter or input value at the classification stage. It is undertaken using the Gabor Wavelet method.

The image convolution obtains the 2-D Gabor Wavelet with a Gabor Wavelet kernel. These wavelet filters are applied to the image that is designed to selectively pass-through regions containing a specific size and orientation of a given shape structure [28]. It is an efficient transformation in compression because it can provide optimal resolution in extracting local features [29]. The image filtering process using the Gabor Wavelet is seen as a convolutional operation between the image matrix and the convolution kernel matrix. Parameters, such as x, y, j, and θ are used to determine the real and imaginary kernel.

This characteristic is a value obtained from combining each orientation angle value θ and the frequency value F. The different varieties of the values of θ and F produce different characteristic values. These feature values represent the elements in an image feature vector. The dimensions of the resulting feature vector depend on the number of combinations of θ and F used [30]. This stage involves calculating the feature value of an image according to the parameters used. The process of the Gabor Wavelet feature extraction process is shown in Fig 4.

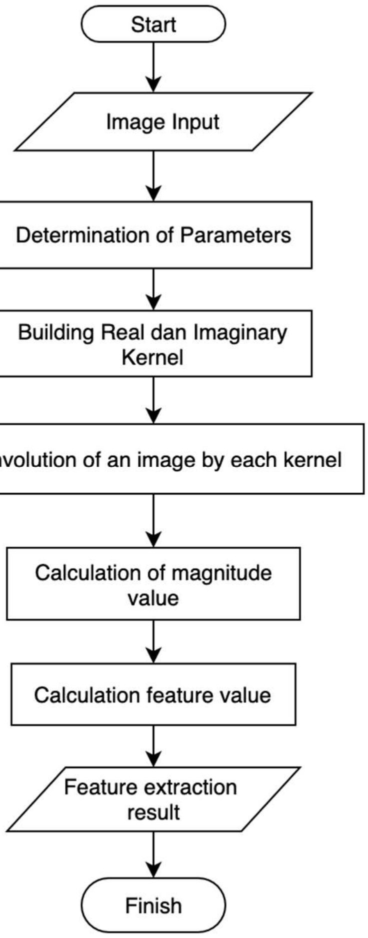
Start
Image Input
Determination of Parameters
Building Real dan Imaginary Kernel
Convolution of an image by each kernel
Calculation of magnitude value
Calculation feature value
Feature extraction result
Finish

Fig. 4 Gabor wavelet process

Feature extraction using the Gabor Wavelet has several processes:

1) Input Process: The image input process is carried out by inputting a previously segmented letter image. Then, the image will be resized to match the size of all images to be extracted.

2) Determine Parameters: In kernel development, several parameters are needed, including λ which represents the wavelength, θ which represents the angle of orientation from normal to the parallel line of the Gabor function, ψ which represents the phase offset, σ which represents the sigma parameter or standard deviation of the Gaussian envelope, and γ which represents the aspect ratio spatially and to determine the ellipticity of the Gabor function support.

3) Build real and imaginary kernels: This is performed from Gabor Wavelets using the following formula [31].

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right) cos(2\pi Fx' + \psi) \quad (1)$$

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right) sin(2\pi Fy' + \psi) \quad (2)$$

$$x' = x\cos\theta + y\sin\theta \quad (3)$$

$$y' = -x\sin\theta + y\cos\theta \quad (4)$$

$$mag = \sqrt{r^2 + im^2} \quad (5)$$

Where F is the frequency, θ is the angle of orientation to the image ranging from 0 to 2p, and σ is the sigma. The equations are divided into the real kernel and the imaginary kernel. Formula 1 is the real kernel, and formula 2 is the imaginary kernel. Meanwhile, Formula 5 is the formula used to calculate the result of convoluting images by the real kernel and the imaginary kernel.

4) Convolution image by each kernel: The next stage is convoluting an image by each kernel that has been built by, which will produce a real convolution image and an imaginary convolution image.

5) Calculate the magnitude value: This calculation is done by using equation 5. A Magnitude output value consists of the square root of the sum of the square of real parts and square of imaginary parts. It is used as the feature of the image.

6) Calculate the statistical features: This calculation is used for the classification process. This research uses the mean, standard deviation, variance, and median statistical values for features to extract.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (6)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (7)$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \quad (8)$$

$$Med = x_{\frac{1(n+1)}{2}} \quad (9)$$

D. Support Vector Machine Classification

Support Vector Machine is a learning method used for binary classification, basically looking for the best hyperplane which works as a separator of two classes in the input field [32]. The SVM method will search and separate the hyperplane into two classes. Hyperplane as the best separator is figured out by finding the hyperplane margin and finding the maximum point value. Margin is the distance between the hyperplane and the pattern that is closest to each class. The closest pattern is called Support Vector.

At first, SVM was developed to solve the problem of two classes. And then, it is developed again to classify more than two classes or multi-class. The multi-class classification is divided into two that are commonly used, namely one against all (OAA) and one against one (OAO) [33]. This study uses the One Against All method with the following formula to determine the class of new data (x) based on the largest hyperplane value using the following equation [34].

$$classx = argmax(\textstyle\sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b \quad (10)$$

This SVM classification stage will produce words in alphabetical letters according to the image entered into the system. The process is shown in Fig 5.

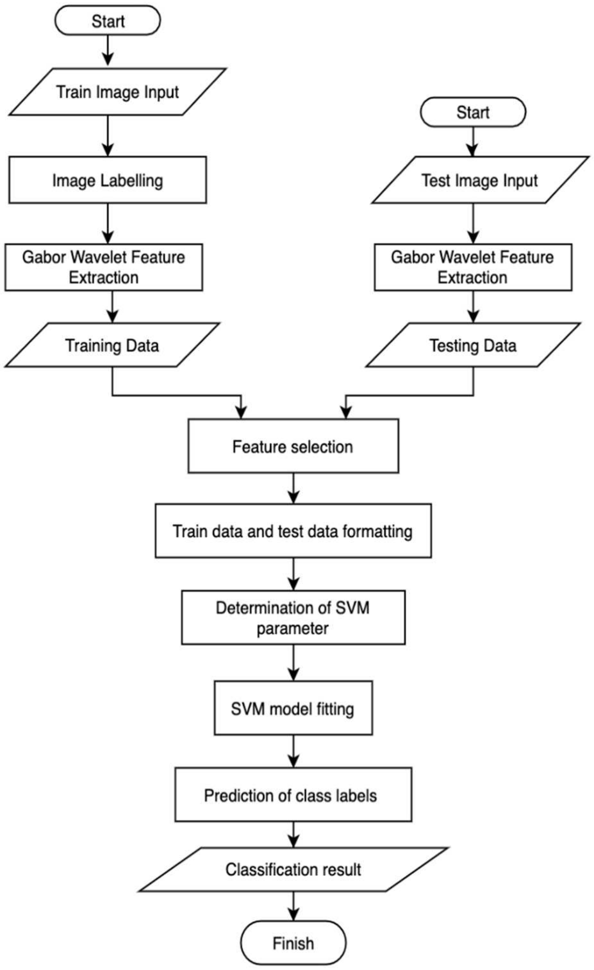
Start
Train Image Input
Image Labelling
Gabor Wavelet Feature Extraction
Training Data
Start
Test Image Input
Gabor Wavelet Feature Extraction
Testing Data
Feature selection
Train data and test data formatting
Determination of SVM parameter
SVM model fitting
Prediction of class labels
Classification result
Finish

Fig. 5 SVM process using training data and testing data

For the classification process using the support vector machine (SVM) method is conducted based on the results from the feature extraction process. In this process, features of test data are compared with features of train data that have been labeled. In this classification, there are 36 classes, 26 of which are letters of the alphabet, 9 punctuations, and 1 space. This study employed one against all methods.

The first step is to read the extraction feature and choose the feature value that will be used for classification. Then perform the test data formatting and training data by determining the features and class labels. Determining the SVM parameters needed in this classification process and then fitting the model according to the training data. After being fitted, the model can be used to predict the test data based on the features that have been selected. Then, the classification result is produced in the form of a class label for each tested feature.

III. Result and Discussion

A. Result

Braille images are processed to become the image of each letter. The letter image will be processed with the feature extraction process using Gabor Wavelet. To simulate the kernel development process, the Gabor kernel to be made has a size of 3x3 with a frequency of $\frac{\sqrt{2}}{2^{58}}$, sigma x is 10, sigma y is 1, and theta (θ) is 90^0.

$$g_{real}(1{,}1) = \frac{1}{2\pi x 10 x 1} exp\left(-\frac{1}{2}\left[\frac{(x\,cos\,9\,0 + y\,sin\,9\,0)^2}{10^2} + \frac{(-x\,sin\,9\,0 + y\,cos\,9\,0)^2}{1^2}\right]\right)$$

$$cos(2\pi)\frac{\sqrt{2}}{2^{58}}(x\,cos\,9\,0 + y\,sin\,9\,0) = 6.46$$

$$g_{im}(1{,}1) = \frac{1}{2\pi x 10 x 1} exp\left(-\frac{1}{2}\left[\frac{(x\,cos\,9\,0 + y\,sin\,9\,0)^2}{10^2} + \frac{(-x\,sin\,9\,0 + y\,cos\,9\,0)^2}{1^2}\right]\right)$$

$$sin(2\pi)\frac{\sqrt{2}}{2^{58}}(-x\,sin\,9\,0 + y\,cos\,9\,0) = -2.67$$

The calculation continues until the last element. Below are the results of building a real kernel and an imaginary kernel.

$$g_{real} = \begin{pmatrix} 6.46 & 3.17 & 1.26 \\ 1.3 & 4.32 & 1.16 \\ 1.18 & 2.63 & 4.76 \end{pmatrix}$$

$$g_{im} = \begin{pmatrix} -2.67 & -1.75 & -8.75 \\ -9 & -3.57 & -1.12 \\ -1.14 & -2.9 & -5.91 \end{pmatrix}$$

After the real and imaginary Gabor kernels are obtained, the next step is to convolute the image with the two kernels. The following is the original image matrix and the convoluted image with each Gabor kernel.

$$Original\,Ima\,ge = \begin{pmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 5 & 5 & 6 & 6 \\ 4 & 5 & 5 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 5 & 5 \end{pmatrix} x \begin{pmatrix} 6.46 & 3.17 & 1.26 \\ 1.3 & 4.32 & 1.16 \\ 1.18 & 2.63 & 4.76 \end{pmatrix}$$

Real image convolution

$$(r) = \begin{pmatrix} 124 & 126 & 122 & 133 & 140 \\ 118 & 114 & 119 & 124 & 137 \\ 117 & 129 & 132 & 147 & 153 \\ 115 & 118 & 129 & 138 & 148 \\ 103 & 114 & 129 & 144 & 149 \end{pmatrix}$$

$$Original\,Ima\,ge = \begin{pmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 5 & 5 & 6 & 6 \\ 4 & 5 & 5 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 5 & 5 \end{pmatrix} x \begin{pmatrix} -2.67 & -1.75 & -8.75 \\ -9 & -3.57 & -1.12 \\ -1.14 & -2.9 & -5.91 \end{pmatrix}$$

$$\text{Imaginary image convolution } (im) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After convoluting the image with each kernel is completed, the convoluted image with the real kernel will be stored in the variable r and the imaginary kernel in the variable m. The results of the two convolutions to get the magnitude value uses the following equation:

$$r^2 = \begin{pmatrix} 16 & 4 & 36 & 25 & 144 \\ 100 & 196 & 81 & 16 & 81 \\ 121 & 1 & 16 & 105 & 113 \\ 169 & 100 & 1 & 100 & 114 \\ 113 & 196 & 1 & 0 & 185 \end{pmatrix}$$

$$im^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$mag^2 = \sqrt{r^2 + im^2} = \begin{pmatrix} 4 & 2 & 6 & 5 & 12 \\ 10 & 14 & 9 & 4 & 9 \\ 11 & 1 & 4 & 10.25 & 10.6 \\ 13 & 10 & 1 & 10 & 12 \\ 10.6 & 14 & 1 & 0 & 13.6 \end{pmatrix}$$

The results of the magnitude are extracted to obtain its feature values, such as the mean, variance, standard deviation, and median, which will be stored in the training data.

$$\mu = \frac{1}{25}\sum_{i=1}^{25} x_i = 7.884$$

$$\sigma = \sqrt{\tfrac{1}{25}\textstyle\sum_{i=1}^{25}(x_i - 7.884)^2} = 4.516$$

$$\sigma^2 = \frac{1}{25}\sum_{i=1}^{25}(x_i - 7.884)^2 = 20.4$$

$$Med = x_{\frac{1(25+1)}{2}} = x_{13} = 10$$

This classification process is based on the results obtained from the feature extraction process. In this process, feature matching is carried out with the features in the existing database. In this classification, there are 27 classes, 26 of which are letters of the alphabet and one of which is space. In this study, multi-class SVM uses one against all methods. For more details, the following is an illustration of the calculation in the SVM classification. In the example of applying the

SVM method using 6 sample data training, each of which has A, B, and C classes.

TABLE I
THE DATA TEST

sample	x1	x2	x3	x4	x5	x6	x7	x8
bx	0.17	0.24	0.29	0.68	0.038	0.058	0.23	0.8

TABLE II
THE TRAINING DATA ON THE DATA FRAME

A1	A2	B1	B2	C1	C2
0.21	0.22	0.18	0.19	0.25	0.22
0.15	0.16	0.24	0.24	0.16	0.16
0.23	0.28	0.27	0.29	0.23	0.20
0.79	0.78	0.69	0.70	0.71	0.71
0.046	0.050	0.036	0.039	0.061	0.048
0.022	0.024	0.060	0.057	0.027	0.027
0.13	0.175	0.23	0.24	0.08	0.09
0.84	0.83	0.8	0.81	0.76	0.77

Then dot product calculation is undertaken with the linear kernel function K (x, y) based on the Table I and Table II above.

TABLE III
DOT PRODUCT TABLE

A1	A2	B1	B2	C1	C2
1.380524	1.394942	1.333448	1.357988	1.260684	1.25669

The sample data B will be tested on each training data for each class. The class that has the highest yield will be the final decision. The values α, y, and b are taken from the training data, while the values *K (x_i, x_j)* are from Table III.

$$\alpha A=\begin{bmatrix}0.11069172\\0.11075006\\0.09595654\\0.09573052\\0.09644951\\0.09650884\end{bmatrix} Bias_A=0.20$$

$$\alpha B=\begin{bmatrix}0.0961934\\0.09620797\\0.11090821\\0.11103067\\0.09681771\\0.09685297\end{bmatrix} Bias_B=0.22$$

$$\alpha C=\begin{bmatrix}0.09557028\\0.09547621\\0.09584603\\0.09569755\\0.11136414\\0.11134695\end{bmatrix} Bias_C=0.24$$

The following are the decision function for each class.

Class A = ((1*0.1106917*1.380524) + (1*0.1107501 * 1.394942) + (-1*0.959565*1.333448) + (-1 *0.0957305*1.357988) + (-1*0.0964495* 1.260684) + (-1 * 0.0964937 * 1.25669)) + 0.2001206
= 0.0066141

Class B = 0.0120955

Class C = -0.002022

Based on the final decision function in each class A, B, and C, the braille image tested is included in class B because the class B classification result is the highest classification result of the other classes.

B. *Testing Result*

Application testing is implemented after the coding stage is completed. The testing stage aims to determine the level of success of the classification process that has been built. System testing is implemented in this research by creating a multi-class confusion matrix and validating the application's data; 15 words will be taken randomly to be read directly by a visually impaired person.

TABLE IV
CONFUSION MATRIX 2 CLASS SCENARIO

		Predicted Class	
		A	B
Actual Class	A	TP	FN
	B	FP	TN

Through confusion matrix testing, accuracy, precision, and recall results can be assessed. In the measurement of confusion matrix multi-class, there are four terms to represent the results of the classification process, namely:

- TP (True Positive) is the number of instances correctly classified by the system like A.
- TN (True Negative) is the number of instances correctly classified by the system as B
- FN (False Negative) is the number of instances incorrectly classified by the system as B that is A
- FP (False Positive) that is the number of instances incorrectly classified by the system as A that is B [31].

The next is to calculate the accuracy obtained from the number of diagonal values of the confusion matrix divided by the total value of the prediction, the precision obtained from the number of correct values of the class divided by the number of predictions in its class, and the recall obtained from the number of correct values of the class divided by the actual number of classes. The test data used are 758 letters consisting of braille letters and punctuation marks. The formula for measuring the three parameters is presented in the following equation.

$$Accuracy=\frac{TP+TN}{TP+FP+FN+TN} \tag{11}$$

$$Precision=\frac{\sum_{i=1}^{n}(\frac{TPi}{FPi+TPi})}{n} \tag{12}$$

$$Recall=\frac{\sum_{t=1}^{n}(\frac{TPi}{FNi+TPi})}{n} \tag{13}$$

The test results using the confusion matrix multi-class are shown in the following Table V. Based on the table, the accuracy value is obtained by calculating the accuracy as in equation 11.

$$Accuracy=\frac{744}{758}x100\%=98.15\%$$

TABLE V
CONFUSION MATRIX MULTI-CLASS

Actual	Prediction																																				Detection Rate(%)
	space	!	Numeric Marker	Capital Marker	,	-	.	/	:	?	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
space	86																																				100
!		10																																			100
Numeric Marker			6																																		100
Capital Marker				11																																	100
,					10																																100
-						10																															100
.							10																														100
/								8																													100
:									9																												100
?										11																											100
a											87																										100
b												25																									100
c													13																								100
d														15																							100
e															38																						100
f																9																					100
g																	23																				100
h							3											15																			83
i																			46																		100
j																				14																	100
k																					29	1															96,7
l																						19															100
m																							27														100
n																								39	1												97,5
o																									16												100
p																										22	1										96
q																											7										100
r							1											1										34									94
s																													16	1							94
t																											1			18							95
u	1																						3								18						92
v																																8					100
w																																	9				100
x																																		8			100
y																																			11		100
z																																				7	100
Overall Accuracy																																					98,15303

TABLE VI
RECALL AND PRECISION VALUES FOR EACH CHARACTER

Class	Recall	Precision
space	1	0,99
!	1	1
Numeric Marker	1	1
Capital Marker	1	1
,	1	1
-	1	1
.	1	0,71
/	1	1
:	1	1
?	1	1
a	1	1
b	1	1
c	1	1
d	1	1
e	1	1
f	1	1
g	1	1
h	0,83	0,94
i	1	1
j	1	1
k	0,97	1
l	1	0,95
m	1	0,9
n	0,97	1
o	1	0,94
p	0,96	1
q	1	0,78
r	0,94	1
s	0,94	1
t	0,95	0,95
u	0,82	1
v	1	1
w	1	1
x	1	1
y	1	1
z	1	1

The precision value used is the average precision for each class of the classification model. The precision value for each class and the average are calculated using equation 12. So, testing using the confusion matrix produces an average value of 98.15% accuracy, 97.66% precision, and 98.28% recall.

$$Precision = \frac{35.157}{36} x100\% = 97.66\%$$

$$Recall = \frac{35.3827}{36} x100\% = 98.28\%$$

C. *Analysis*

From the implementation and testing done before, the Gabor Wavelet can work well when combined with the SVM method, classifying the features obtained in the feature extraction process and identifying braille letters with good results. This can be proven in confusion matrix testing, which produces an average value of 98.15% accuracy, 97.66% precision, and 98.28% recall.

The high accuracy of the classification process method depends on the number of braille character test image data used. The more images used, the higher the accuracy value, and the less the number of test image data, the smaller the accuracy value. Then, the slope in the braille script and the presence of dots or stains on the paper used to write braille can affect the identification results. For example, there is a strip close to the braille dots. The identification results can still produce letters according to the letters whose characteristic values are relative to those of the previous training data.

The failure in identifying letters is due to the value resulting from feature extraction in the image in the classification process that tends to be close to other classes, compared to the actual letter class. In this research, the braille letters identified only can be done in 2 forms meaning that if numbers are to be identified, a word only consists of numbers. Then, words in braille only contain letters and punctuation without numbers.

In braille identification application, identifying an image of a word in braille into the alphabet and the voice takes 4 seconds on average, but in a braille image that is relatively longer in numbers, producing voice output takes on average 11 seconds. Afterwards, the voice will be pronounced in Indonesian, which is provided by Google, so an internet connection is required to get the voice output. But, if there is no internet connection, its application still runs and generates identification results in words of alphabetical letters.

IV. Conclusion

The results of the study show that the Gabor Wavelet and Support Vector Machine (SVM) is successfully applied to identify braille into the alphabet in digital image processing. This method can be implemented to identify braille letters in uppercase, lowercase, numbers, and punctuation, giving the test results of the confusion matrix method with an average value of 98.15% accuracy, 97.66% precision, and 98.28% recall. On average, it takes to identify a braille image is 4 seconds, and identifying numbers needs 11 seconds. Testing is done by using random test data of 758 data. Stains influence the incorrect classification results in this study on the paper around the braille word images and the slope of the braille image. For further research, it is recommended to identify the difference between dots of braille and stains on paper.

References

[1] D. Berthelsen and S. Walker, "Parent's Involvement in Their Children's Education," *Family Matters,* no. 79, pp. 34-41, 2008.

[2] M. Khochen-Bagshaw, "Reading through touch, importance and challenges," in *World Congress Braille 21*, Germany, 2011.

[3] K. Smelyakov, A. Chupryna, A. Sakhon, D. Yeremenko and V. Polezhai, "Braille Character Recognition Based on Neural Networks," *IEEE Second International Conference on Data Stream Mining & Processin,* pp. 509-513, 2018.

[4] A. Mousa, H. A. R. Hiary and L. Alnemer, "Smart Braille System Recognize," *IJCSI International Journal of Computer Science Issues,* vol. 10, no. 6, pp. 52-60, 2013.

[5] A. Antonscopoulous and D. Bridson, "A Robust Braille Recognition System," *Springer Lecture Notes in Computer Science,* pp. 533-545, 2004.

[6] J. Subur, T. A. Sardjono and R. Mardiyanto, "Braille Character Recognition Using Find Contour Method," in *the 5th International Conference on Electrical Engineering and Informatic*, Bali, Indonesia, 2015.

[7] S. Ibrahim, N. A. Tarmizi, N. Sabri, N. F. M. Johari and A. F. A. Fadzil, "Braille Image Recognition for Beginners," in *2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC 2018)*, Shah Alam, Malaysia, 2018.

[8] B. Nurgroho, I. Ardiyanto and H. A. Nugroho, "Review of Optical Braille Recognition Using Camera for Image Acquisition," in *2nd International Conference on Biomedical Engineering (IBIOMED)*, 2018.

[9] V. V. Murthy and M. Hanumanthappa, "Improving Optical Braille Recognition in Pre-processing stage," in *2018 International Conference on Soft-computing and Network Security (ICSNS)*, 2018.

[10] N. D. Silva and S. Vasanthapriyan, "Optical Sinhala Braille Documents Convertion Methodology for Sinhala Language," in *2018 National Information Technology Conference (NITC)*, Colombo, Sri Lanka, 2018.

[11] R. Li, H. Liu, J. X. X. Wang and Y. Qian, "Optical Braille Recognition Based on Semantic Segmentation Network with Auxiliary Learning Strategy," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[12] K. Rączy, M. Czarnecka, D. Zaremba, K. Izdebska, M. Paplińska, G. Hesselmann, A. Knops and M. Szwed, "A shared code for Braille and Arabic digits revealed by cross-modal priming T in sighted Braille readers," *Acta Psychologica,* 2020.

[13] A. Sharma, S. Devi and J. K. Verma, "Braille Book Reader using Raspberry Pi," in *2020 International Conference on Computational Performance Evaluation (ComPE)*, India, 2020.

[14] M. R. Phangtriastu, J. Harefa and D. F. Tanoto, "Comparison Between Neural Network and Support Vector Machine In Optical Character Support Vector Machine In Optical Character Recognition," *Procedia Computer Science,* pp. 351-357, 2017.

[15] A. A. Chandini and U. M. B., "Improved Quality Detection Technique for Fruits Using GLCM and MultiClass SVM," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, 2018.

[16] M. F. Zuandi, B. Hidayat and S. Sitam, "Granuloma Image Detection Through Periapical Radiograph by Using Gabor Wavelet Method and Support Vector Machine Classification," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, 2018.

[17] I. Wicaksono, H. Kusuma and T. A. Sardjono, "Traffic Sign Image Recognition Using Gabor Wavelet and Principle Component Analysis," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, Yogyakarta, Indonesia, 2019.

[18] D. Kathuria and J. Yadav, "An Improved Illumination Invariant Face Recognition Based on Gabor Wavelet Transform," in *2018 Conference on Information and Communication Technology (CICT)*, Jabalpur, India, 2018.

[19] Y. Wang, H. Liu, D. Wang and D. Liu, "Image processing in fault identification for power equipment based on improved super green algorithm," *Computers & Electrical Engineering,* vol. 87, 2020.

[20] A. Prasetio and P. M. Hasugian, "Improving the Quality of Digital Images Using the Median Filter Technique to Reduce Noise," *Journal Publications & Informatics Engineering Research,* vol. 4, no. 1, pp. 143-148, 2019.

[21] V. Mohan, R. K. Durga, S. Devathi and K. S. Raju, "Image Processing Representation Using Binary Image; Grayscale, Color Image, and Histogram," in *Proceedings of the Second International Conference on Computer and Communication Technologies*, New Delhi, 2016.

[22] Y. Zhu, L. Wang, Y. Behnamian, S. Song, R. Wang, Z. Gao, W. Hu and D.-H. Xia, "Metal pitting corrosion characterized by scanning acoustic microscopy and binary image processing," *Corrosion Science,* vol. 170, 2020.

[23] K. Somasundaram, J. H. Mercina, S. Magesh and T. Kalaiselvi, "Brain Portion Extraction Scheme using Region Growing and Morphological Operation from MRI of Human Head Scans," *International Journal of Computer Sciences and Engineering,* vol. 6, no. 4, pp. 41-51, 2018.

[24] Q. Wang, N. Gui, Y. Liu, S. Peng, X. Yang, J. Tu and S. Jiang, "A morphological image processing method for simultaneous scrutinization of particle position and velocity in pebble flow," *Annals of Nuclear Energy,* vol. 148, no. 107704, 2020.

[25] I. S. Masad, A. Al-Fahoum and I. Abu-Qasmieh, "Automated measurements of lumbar lordosis in T2-MR images usingdecision tree classifier and morphological image processing," *Engineering Science and Technology,an International Journal,* vol. 22, pp. 1027-1034, 2019.

[26] T. R. Fujimoto, T. Kawasaki and K. Kitamura, "Canny-Edge-Detection/Rankine-Hugoniot-conditions unified shock sensor for inviscid and viscous flows," *Journal of Computational Physics,* vol. 396, pp. 264-279, 2019.

[27] U. A. Nnolim, "Automated crack segmentation via saturation channel thresholding, areaclassification and fusion of modified level set segmentation with Cannyedge detection," *Heliyon,* vol. 6, no. 12, 2020.

[28] G. Ayalew, Q. U. Zaman, A. W. Schumann, D. C. Percival and Y. K. Chang, "An investigation into the potential of Gabor wavelet features for sceneclassification in wild blueberryfields," *Artificial Intelligence in Agriculture,* vol. 5, pp. 72-81, 2021.

[29] D. A. H. Al-Fayadh, H. R. Mohammed and R. S. Al-shimsah, "Gabor Wavelet Transform in Image Compression," *Journal of Kufa for Mathematics and Computer,* vol. 1, pp. 107-113, 2012.

[30] Adiwijaya, B. Purnama, A. Hasyim, M. D. Septiani, U. N. Wisesty and W. Astuti, "Follicle Detection on the USG Images to Support Determination of Polycystic Ovary Syndrome," *Journal of Physics: Conference Series,* vol. 622, 2015.

[31] C. Adak, "Gabor Filter and Rough Clustering Based Edge Detection," in *International Conference on Human Computer Interactions (ICHCI)*, Chennai, India, 2014.

[32] E. Gul, N. Alpaslan and M. E. Emiroglu, "Robust optimization of SVM hyper-parameters for spillway type selection," *Ain Shams Engineering Journal,* 2021.

[33] K. Thirumala, S. Pal , T. Jain and A. C. Umarikar , "A classification method for multiple power quality disturbances using EWT based adaptive filtering and multi-class SVM," *Neurocomputing,* vol. 334, pp. 265-274, 2019.

[34] Z. Wang and X. Xue, "Multi-Class Support Vector Machine," *Ma Y., Guo G. (eds) Support Vector Machines Applications,* pp. 23-48, 2014.