REAL TIME TEXT RECOGNITION USING BLOB DETECTION AND BACKPROPAGATION NEURAL NETWORK

Afifatul Mukaroh
afifatul.mukaroh@gmail.com
201883597

I. INTRODUCTION

Self-driving car or walking robot is highly developed these days. For that, real time text recognition has been a certain topic that attract many researchers. As real time recognition is needed, it's not just accuracy but also processing time becomes an issue here. Text recognition can be implemented as robot vision for guiding the direction or location. This project develops real time text recognition using mobile device.

II. BACKGROUND THEORY

A. Blob Analysis

Blob Analysis is a fundamental technique of machine vision based on analysis of consistent image regions [1]. Blob analysis is most basic method for analyzing the shape features of an object, such as the presence, number, area, position, length, and direction of lumps. Blob also can be interpreted as a response of a particular detector like color detector, face detector, motion detector, or edge detector. Blob detection can be used for object extraction, object removal, compositing, or others.

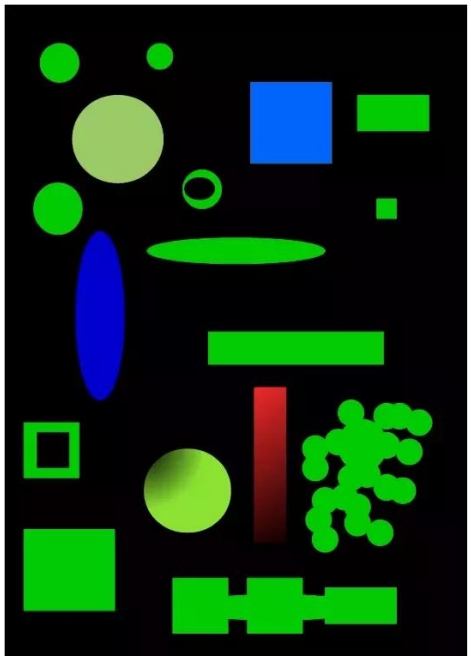

Figure 1. Example of blobs

B. Histogram

Histogram of an image is graphic that represents the value of pixel distribution from that image or spesific part of that image [2]. Histogram of image could be calculated by equation 1.

$$h_i = \frac{n_i}{n} \quad , i = 0,1, \dots, L-1 \quad \text{.....(1)}$$

n_i is the number of pixels with grayscale *i*. *n* is the total number of pixels in the image. *L* is grayscale value of the image, started from 0 to *L* – 1 (for example in the image with grayscale quantization of 8-bit, the value of grayscale woud be from 0 to 255).

C. Backpropagation Neural Network

Backpropagation is one of methods in Artificial Neural Networks and has supervised training which uses weight adjustment patterns to achieve the minimum error between predictions output and real output [3].

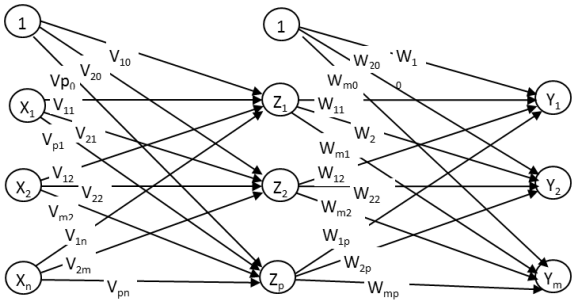

Figure 2. Backpropagation Architecture

The training steps according to backpropagation architecture in Figure 2 which has input node $X_i(\ i = 1,2,3,\dots,n)$, hidden node $Z_j\ (j = 1,2,3,\dots,p)$, output node $Y_k(k = 1,2,3,\dots,m)$, weight connecting input layer and hidden layer V_{pn}, and weight connecting hidden layer and output layer W_{mp} is explained below [4] :

1. Initialize the weights (Generalize the initial weights with the smallest random values).
2. While the condition has not met, do steps bellow:

Feedforward

a. Every input is saved as x_i and pass it into all nodes in the next layer (hidden layer).
b. Calculate all the output in hidden nodes (Z_j):

$$z\_net_j = V_{jo} + \sum_{i=1}^{n} X_i V_{ji} \quad \text{.....(2)}$$

Then calculate the output using activation function sigmoid,

$Z_j = f\left(z\_net_j\right) = \frac{1}{1+e^{-z\_net_j}}$ .....(3)

save it as the value of hidden nodes and send it into all nodes in the next layer (output layer).

c. Calculate all the output in output nodes (Y_k)
$y\_net_k = W_{ko} + \sum_{j=1}^{p} X_j V_{kj}$ .....(4)

Then if using activation function sigmoid, calculate with :

$Y_k = f\left(y\_net_k\right) = \frac{1}{1+e^{-y\_net_k}}$ .....(5)

and save it as the value of output nodes.

Backward

d. Output nodes y recieve pattern target based on each pattern input that it also receive, calculate the error:
$\delta_k = (t_k - Y_k) f'\left(y_{net_k}\right) = (t_k - Y_k) Y_k\,(1 - Y_k)$ .....(6)

Calculate the weight rate W_{kj} with learning rate α

$\Delta W_{kj} = \alpha\, \delta_k Z_j$ .....(7)

Pass δ to the previous nodes.

e. Calculate factor δ of hidden nodes based on the error in every hidden nodes Z_j
$\delta\_net_j = \sum_{k=1}^{m} \delta_k W_{kj}$ .....(8)

Factor δ hidden nodes :

$\delta_j = \delta\_net_j\, f'(z\_net_j) = \delta\_net_j Z_j\,(1 - Z_j)$ .....(9)

Calculate the weight rate of V_{ji}

$\Delta V_{ji} = \alpha\, \delta_j X_i$ .....(10)

f. Every output nodes corrects the weights :
The new weight that connected output nodes:

$W_{kj}\,(new) = W_{kj}\,(old) + \Delta W_{kj}$ .....(11)

The new weight that connected hidden nodes:

$V_{ji}\,(new) = V_{ji}(old) + \Delta V_{ji}$ .....(12)

III. APPROACHES

In this project, methods that being used to recognize text consists of three stages. Those are blob detection, segmentation, and character recognition. These all are implemented as android application. But, before it is developed as android application, there is training program developed using Matlab. The purpose training program is to obtain a model for text recognition. The framework for matlab program shown in Figure 3, while the framework for android application shown in Figure 4.

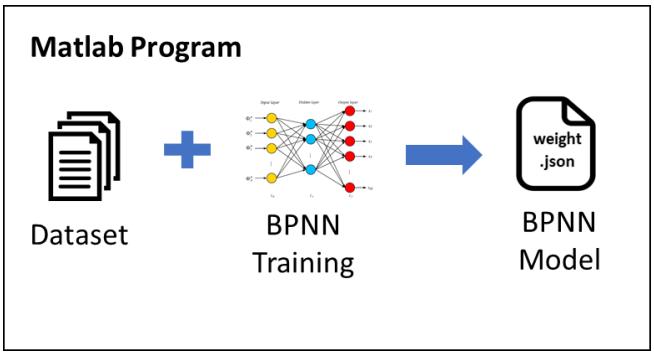

Figure 3. Matlab Program Framework

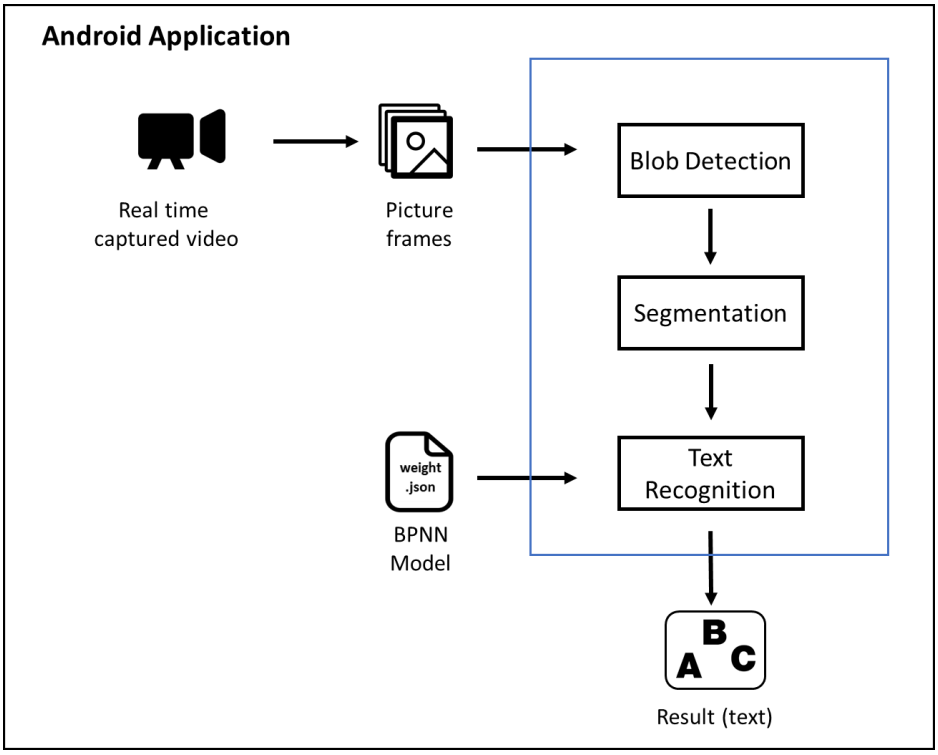

Figure 4. Android Application Framework

A. Blob Detection

In this stage, region that contains text is detected. Because text that is going to be recognized is located inside green box, the approach to detect the blob here is based on the color. The picture is converted into HSV to detect the green color.

Blob region here is rectangle, marked with four coordinates, minimum x, maximum x, minimum y, and maximum y. Pixel by pixel is read at each frame. Every time green pixel is found, it will be marked as blob region. The flow chart of this blob detection process can be seen in Figure 5.

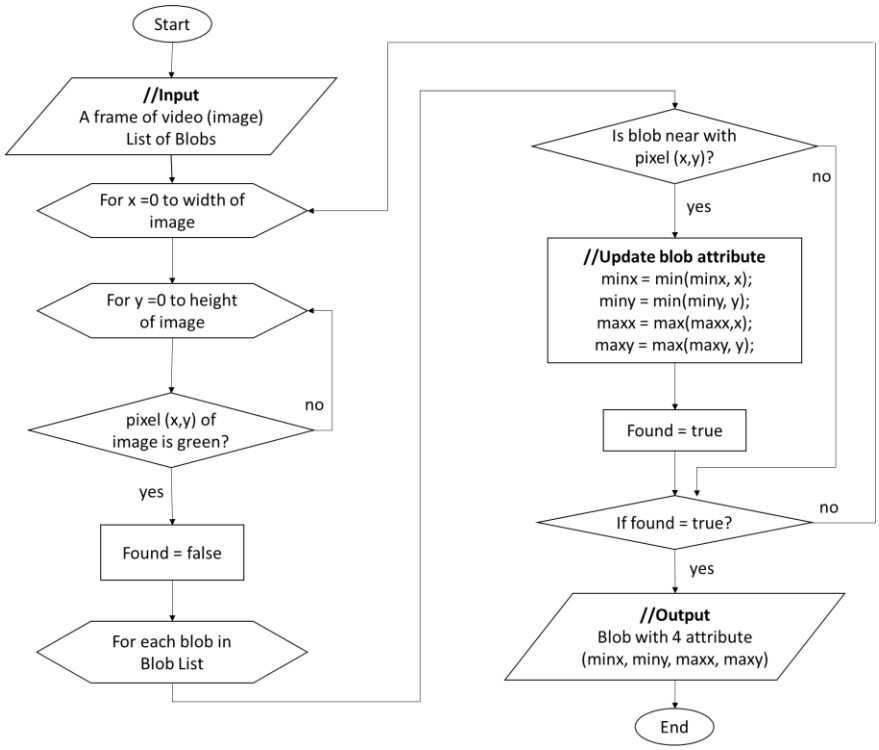

Figure 5. Blob Detection Flowchart

B. Segmentation

Segmentation stage aims to crop the blob into every alphabet character. The process of character segmentation was also using rectangle-shaped of ROI (Region of Interest). The approach that being used in this stage is by reading the histogram of white pixel distribution of the blob. First, the blob region is processed into binary image. Then, cropping boundaries are determined by calculating the histogram distribution of white pixels vertically (based on axes y). Then the histogram is read from left to right (based on axes x). If white pixel is found, then it is considered as a “character image”. If black pixel is found, then it is considered as the spaces between a “character image”. In the other words, a character is identified if a range in the histogram has value not equal to zero (In Figure 6, it is shown by red highlight marks).

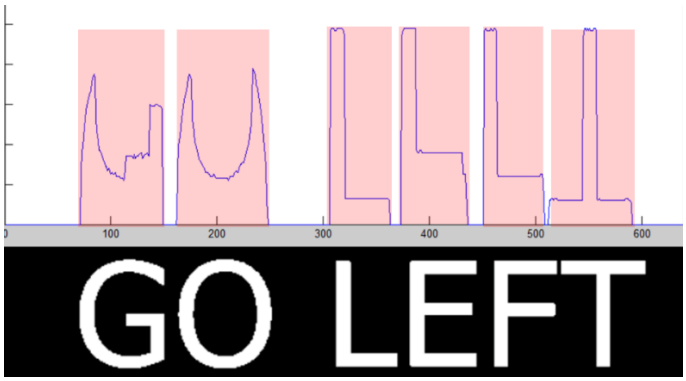

Figure 6. Histogram-based segmentation

C. Character Recognition

After alphabet characters are segmented, it is recognized with Artificial Neural Network (ANN). ANN that being chose for this case is Backpropagation Neural Network (BPNN). The feature for this neural network is pixel of binary images of each character. First, every character is resized into 14x8 size, so that 112 of pixel one and zero can be obtained and used as features. Figure 7 shows the architecture of BPNN used here. The input layer consist of 112 nodes X = {X_1, X_2, … , X_{112} }. The hidden layer consists of 75 nodes Z = {Z_1, Z_2, … , Z_{75} }. The output layer consists of 26 nodes Y = {Y_1, Y_2, …, Y_{26}}. This output nodes represents 26 alphabet characters. If alphabet character is A, then Y_1 will have value 1, while the others are 0, If alphabet character is B, then Y_2 will have value 1, while the others are 0, and so on. V_{ij} is weight that connects input layer and hidden layer where i = 1, 2, …, 75 and j = 0, 1, …, 112. W_{mn} is weight that connects hidden layer and output layer where m = 1, 2, … 26 and n = 0,1, …, 75. Both weights are obtained from training process in Matlab program.

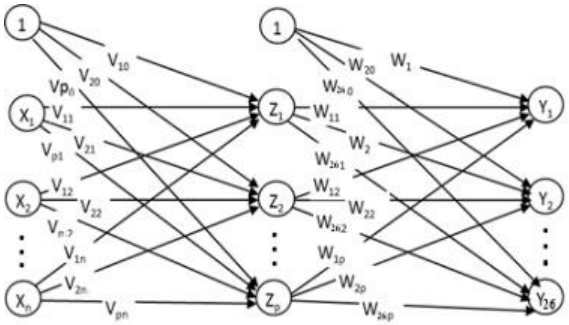

Figure 7. BPNN Architecture

IV. EXPERIMENTAL RESULT

In order to mimic as real robot vision, this text recognition system is implemented as android application. It's developed using Android Studio to develop the application but using Matlab to generate the neural network model.

A. Dataset

Dataset that being used in this project consists of 130 pictures of alphabet character with 5 pictures of each character. The alphabet characters here are used specific font and only uppercase character. Figure 8 shows some of training dataset.

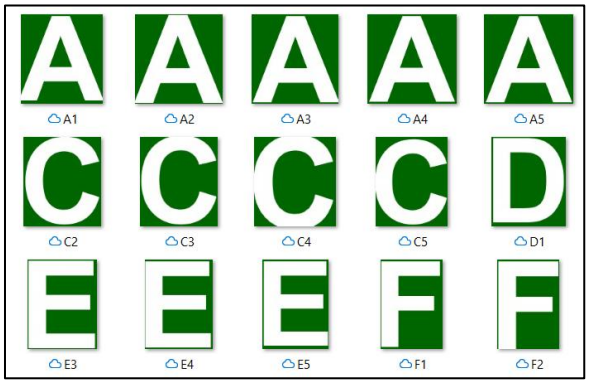

Figure 8. Overview of Training Dataset

B. Training

Training aims to generate neural network model. The output of training is weights. It used learning rate 0.1. It stopped when epoch reached 4785 and loss value 0.0001. Figure 7 shows the training curve for this training.

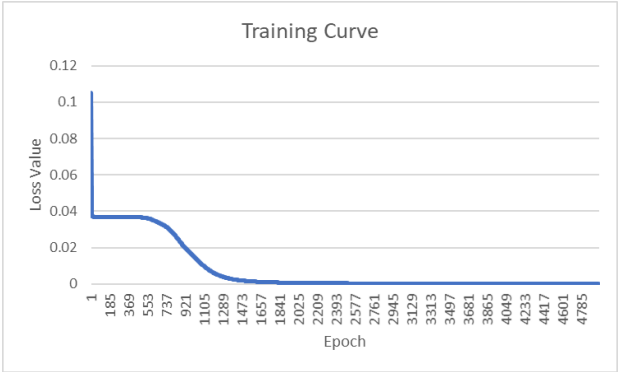

Figure 7. Training Curve based on Epoch

C. Implementation

After training, the obtained weight is embedded to android program in form of json file. With this file, android can do neural network prediction. Figure 8 shows the result of this prediction. A white rectangle in green area is the result of blob detection, six blue rectangles among each character are the result of segmentation, while red text in the upper screen is the result of text recognition.

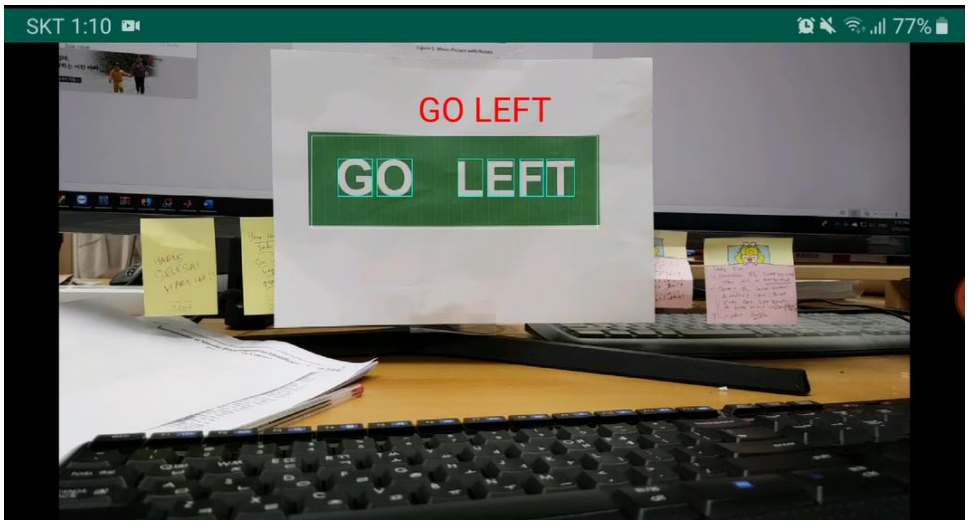

Figure 8. Implementation in Android Application

V. CONCLUSION

As robot vision, real time text recognition is needed for location guiding. This project develops real time text recognition based on Android application. This approach used are blob detection, segmentation, and artificial neural network (ANN). The recognition using ANN is good, but problem happens in segmentation. Because it processes every frame in video that always moves, the segmentation is not stable,

VI. REFERENCES

[1] Adaptive Vision. *Blob Analysis*. https://docs. adaptive-vision.com /current/studio/ machine_vision_guide/ BlobAnalysis.html [Accessed 20 May 2019].

[2] Munir, R., 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Penerbit Informatika.

[3] Suhandi, K. F., 2009. Penerapan Jaringan Syaraf Tiruan Untuk Memprediksi Jumlah Pengangguran di Provinsi Kalimantan Timur Dengan Menggunakan Algoritma Pembelajaran Backpropagation. [Online] Available at: http:// informatikamulawarman. files. wordpress. com/ 2010/ 02/08-jurnal-ilkom-unmul-v-5-1-0.pdf [Accessed 19 March 2015].

[4] Fausett, L., 1994. Fundamentals of Neural Network, Architecture, Algorithms and Applications. New Jersey: Prentice Hall.