BISINDO Alphabets Sign Language Classification using Convolutional Neural Network

Faiz Khansa Adrika
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta Indonesia
faiz.k.a@mail.ugm.ac.id

***Abstract*—An innovation, a scientific term called "sign language recognition" (SLR) is a leading topic in computer vision. This helps the deaf and the mute to communicate with others. This research aims to create an image classification model for Bahasa Isyarat Indonesia (BISINDO) alphabet signs. BISINDO is chosen due to its usability that is more natural because they are created from actions and not grammatical language. The main feature of this research is to use advanced preprocessing such as Otsu Thresholding and Canny Edge Detection to highlight the best feature of an image with a hand sign. The model will be developed using Convolutional Neural Network (CNN) using Keras library. The CNN training went smoothly, the early stopping was triggered on epoch number 26 after 2 hour and 10 minutes of training with the accuracy in the validation set of 97.93%. We can further develop a recognition system using object detection based on the model. The sign language recognition system can be implemented in mobile devices**

(*Abstract***)**

Keywords—BISINDO, Sign Language, Sign Language Recognition, Edge Detection, CNN

I. INTRODUCTION

The sign language is the communication tool for society with hearing and and verbal disability, their visual language is the way out. It uses hand gestures, shape, and expression to describe its meaning [1]. The complexion of sign language is that terms could differ with a slight change of the sign aspects in their parts of body which are used to articulate gestures. According to World Health Organization (WHO) per 1 March 2020, 5% of the world population, --or around 466 million people-- has disabling hearing loss. Which the current data represent 432 million adults and 34 million children within the circle of disability[2]

Verbal speakers already have enough trouble for speaking to other verbal speakers. Even a larger problem are handled by the deaf and dumb, the differences make it more difficult to blend in with the majority of people [3]. Being disabled in this case, hearing and verbal, has a lot of downside. A judgement of their capability to work and socialize have made them discriminated or excluded from the society [4].

The solution for verbal speakers to be able to communicate with sign language have never been easier. Innovation after innovation, a scientific term called "sign language recognition" (SLR) is now a leading research field in computer vision for good [5]. Sign language recognition (SLR) enables the user to conveniently translate gestures naturally and freely without the disturbance of devices, since it is moderately affected by small movements [1], [6].

Indonesia mainly has 2 conventional sign languages. The first is called "Sistem Bahasa Isyarat Indonesia" (SIBI), created by Anton Widyatmoko, which already has support from the government by the distribution of SIBI dictionaries to schools. Unfortunately, the use of SIBI is considered impractical compared to the second sign language, "Bahasa Isyarat Indonesia" or BISINDO for short, regarding the ease of cultural context is more natural than SIBI that use grammatical context of spoken Bahasa [3], [7].

The research will create sign language recognition by image classification of preprocessed BISINDO alphabets using convolutional neural network (CNN). The research mainly analyses the classification accuracy of image thresholding and CNN implementation that will be further developed to sign language recognition system in the next research.

II. RELATED WORKS

Bin L, Huann G, and Yun L in IEEE ICSIPA 2019 Malaysia, publicize their paper with the focus study of CNN in static ASL recognition[8]. The research will capture image with smartphone camera. 24 ASL letters gestures captured and 200 images for each, performed by two different users with various background and lighting. CNN components built are one input layer, two cascaded convolutional layer including max pooling and dropout, one flattening layer, and fully connected layer with dropout, and output layer with softmax. Also, 25% dropout rate set to prone overfitting. The overall accuracy is 95%, but there is a fact that the alphabet W and X received 60% accuracy because of the sign gesture similarity. The limitation to the research is the noise reduction for segmentation and hand tracking, also an image occlusion.

Harini and group proposed a webcam based sign language translation system[9]. The image dataset captured from internal computer camera and pre-processed by converting to grayscale image and background substraction, also resized to 28*28 size for the ROI. CNN multilayers namely convolution, pooling, flattening, and connected layer also with ReLu activation function classifies the images into translations. The system able to translate sign language with the accuracy of 99.91%. The evaluation for the system is the image should be captured in the best way possible. Although facial expressions was not featured, the result of CNN method in the model is promising.

Alom also received a high percentage of accuracy by using CNN and SVM. 98.2% and 98.3% accuracy obtained for ASL dataset and SLD dataset, respectively[10]. The dataset are sign digits from ASL dataset and standard database (SLD). The ASL dataset filled with 700 image of 0-9 digits, 100 each, performed from 5 different individuals. The data are pre-processed using bilinear interpolation and noise removal by median filter. Trained Inception-v3 deep CNN architecture

proposed for the feature extraction, transfer learning method to preserve weight value between layers also applied, and lastly the high level feature extracted by softmax. SVM was chosen for the classification for inseparable problem.

The research by Bantupalli and Xie on ASL recognition using deep learning and computer vision, achieved 99% accuracy in training[11]. The dataset was independently created from the Asl dataset curated by Neidel et. al. where 60 fps and 720p videos recorded using iPhone 6 camera, the videos trimmed to 300 frames. Inception CNN model was used, but with 2 different approach of classification: softmax layer and global pool layer. The segmented gesture identified by CNN and classified by LSTM. Softmax layer acquire the average accuracy of 91.5%. But with pool layer the accuracy range was lower, 56.5%. The problems the model faced are facial feature and skin tones, in some cases the accuracy would drop. Background was also an aspect of evaluation, variation of clothing affected and reduced the accuracy. The solution to the problem could be by using ROI to focus the frame to the hand only.

Russian Sign Language (RSL) Dactyl Recognition research conducted by Makarov, the SLR will be applied in mobile phones[12]. The dataset is RSL dactyl which contains 33 gestures, and 26 gestures are static. The dataset is self manufactured using images taken using smartphones and youtube videos. CNN with simple structure of several convolutional and pooling layers with ReLu activations and dense layer is implemented. All the networks trained on 100 epochs using ADAM optimizer and 0.003 learning rate. The result accuracy varies between 70-80%, but having larger dataset by augmentation would increase the accuracy to over 98%. The lower accuracy result from the small-sized self-manufactured dataset, even though the researcher concluded the system may work better than the complex laboratory data, the dataset cannot be generalized on real-world considering the size and variance.

III. METHOD

A. Dataset

The dataset will be 26 BISINDO alphabets sign language with addition of 'nothing' category. The sign language will be collected from public dataset in Kaggle by Riestiya Zain. The dataset is stored in the form of 3000x3000 pixel image dimension, but will be resized into 300x300 dimension.

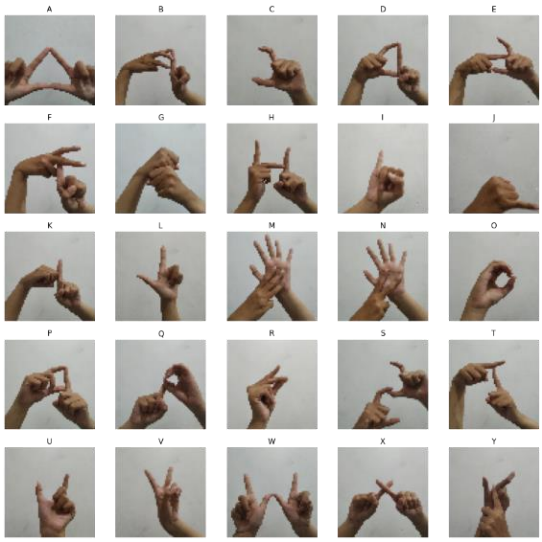

Fig. 1. Dataset visualization

B. Data Preprocessing

The preprocessing method will have the flow of an image below:

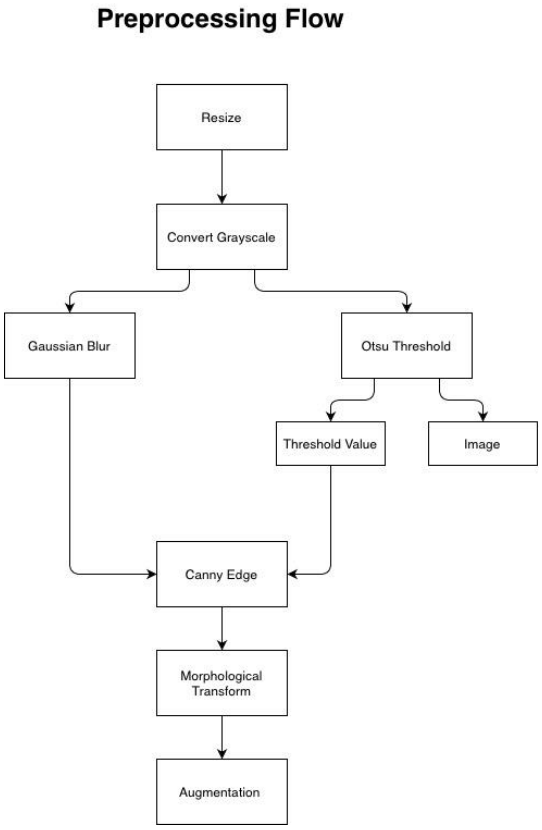

Fig. 2. Preprocessing flow

- Gaussian Blurring

  Gaussian Blurring is done by using Gaussian kernel with the function cv.GaussianBlur() in which the width and height of the kernel should be positive and odd. Standard deviation in the X and Y directions, sigmaX and sigmaY are also needed to be specified. If in the case that only sigmaX is specified, then sigmaY is taken as the same as sigmaX. If both are specified as zeros, they are calculated from the kernel size. This method is adequate to remove Gaussian noise in an image.

- Otsu Thresholding

  Otsu Thresholding determines optimal global threshold value from the image histogram. Unlike global thresholding which uses an arbitrary chosen value as a threshold, this method doesn't choose a value and determines it automatically. To do it, the cv.threshold() function is used. The algorithm finds the optimal threshold value which is returned as the first output and the thresholded image for the second output. The use of the Otsu thresholding is to find the optimal threshold value which later will be used to measure the threshold values parameter of the Canny Edge.

- Canny Edge Detection

  Canny Edge Detection focuses on only the existent edges in the picture. In the process, the distance

between edge pixels are detected so then real edge pixels have to be minimized. To minimize response, only one detector response is considered per edge. The canny edge take lower threshold and upper threshold as the parameter, the upper threshold is equal to the threshold value from Otsu thresholding and the lower threshold takes 0.3 value ratio from the threshold value. Even though it is enough, the edge detection in some cases were still too detailed. Minimizing the detail, morphological transform is used.

- Morphological Transform

  Morphological Transform is a simple operation that based on shape of image. It is conventionally performed on binary images. This method needs two inputs which are original image and kernel that decides the operation. The process includes two basic morphological operators, Erosion and Dilation. Then variant forms such as Opening, Closing and Gradient are called on. In this case, we first use dilation to thicken the edge, then we add opening operation to reduce holes and details within the edges.

The step-by step result of the preprocessing displayed on the figure below:

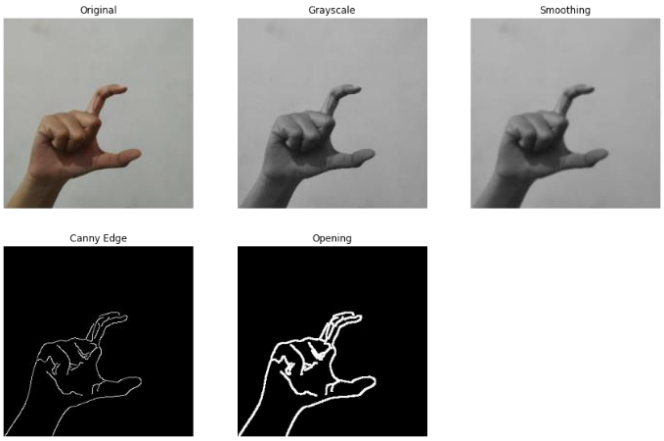

Fig. 3. Preprocessing results

After going through the image processing using OpenCV, the dataset will be further preprocessed by image augmentation using Keras' ImageDataGenerator. Hopefully, this preprocessing will ease the model fitting by normalizing the pixels and give variance of the image in the batches.

C. Classification

The modelling for image classification will need several specific methods that would help in the prediction. These are the methods that will be used:

- Convolutional Neural Network

  The CNN model is specified using Sequential() of Keras library, creating the model by stacking linear layers of convolution functions of your desired model. The CNN consist of 5 main layers: input layer, convolutional layer for convolution operation, pooling layer for matrix discretization to down-sample the input using MaxPooling, flattening layer, and dense/fully connected layer. The activation functions that will be used are ReLU and Softmax algorithm. The detailed layers of the sequential() model described in Figure

- Activation Function

  Rectified Linear Unit (ReLU) Rectified Linear Unit, ReLU for short, is one of the most common and most used activation function in deep learning. Activation function determines the output of neural network as it is connected to each neurons in the network, also as an activation of the neuron function according to the relevant prediction. Activation function technically normalize neurons' output to 0-1 or -1 to 1. Whenever ReLU receives negative input, it will returns 0, and when the input is positive it will returns back the positive value itself. ReLU activation function defined as the following function :

$$F(x) = max(0,x) \quad (1)$$

- The Architecture

  The architecture of the model is detailed in the figure below:

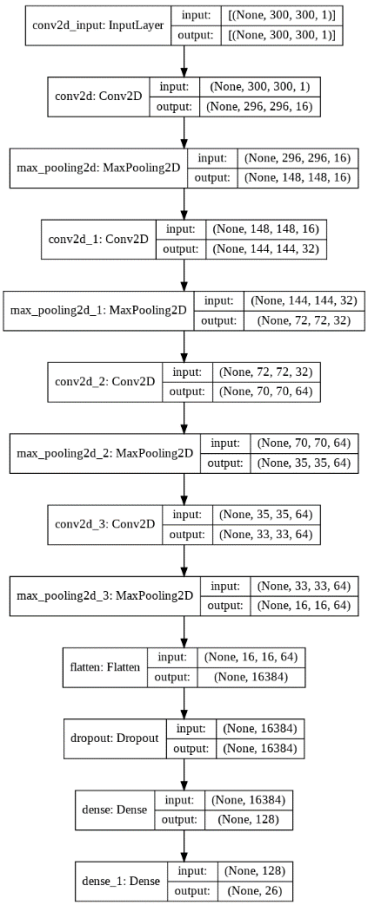

Fig. 4. Model architecture

- Others

  The optimizer used is ADAM optimizer with no further manual parameter on the learning rate. The callback used to prevent overtraining is early stopping with monitor parameter of 'val_loss' and patience of five. It means that the model will stop earlier if the val_loss variable has not progressed (going smaller) in the last 5 epoch

IV. Result and Discussion

The training went smoothly, the early stopping was triggered on epoch number 26 after 2 hour and 10 minutes of training with the accuracy in the validation set of 97.93%. The progress of the result are shown below:

TABLE I. Model training

Epoch	Performance			
	Loss	***Val_loss***	***Acc***	***Val_acc***
1	3.0412	2.1045	10.67%	33.83%
5	0.6449	0.2061	80.82%	93.05%
10	0.2977	0.1367	90.69%	94.74%
15	0.1545	0.1396	94.73%	95.11%
20	0.1182	0.0241	96.43%	99.44%
21	0.1310	0.0239	96.00%	99.25%
22	0.1125	0.0503	96.38%	98.50%
23	0.1004	0.0609	96.90%	98.31%
24	0.1117	0.0484	96.99%	99.06%
25	0.0974	0.0336	97.18%	99.06%
26	0.1004	0.0656	96.76%	97.93%

Fig. 5. Model training progress

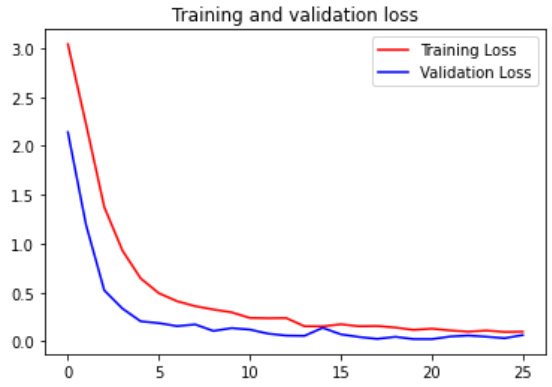

Fig. 6. Accuracy plotting

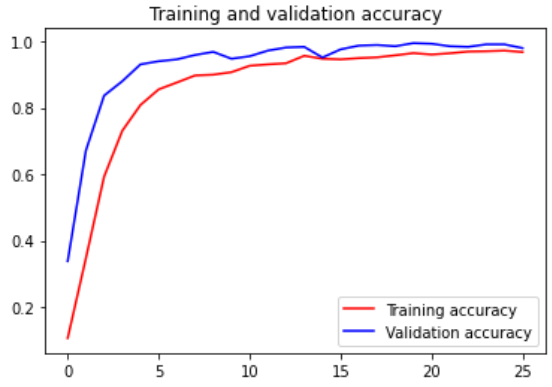

Fig. 7. Loss plotting

Analyzing the graph, the loss descended smoothly over time and the accuracy converge just in time, meaning the model is not overfit or underfit.

TABLE II. Model evaluation

	Test Set 1	**Test Set 2**
Accuracy	97.93%	96.99%
Loss	0.0656	13.2146

Fig. 8. Model evaluation

The model evaluation was conducted using two test dataset with difference in preprocessing. The first test set applies normalization which divides the pixel value by 255.0 into a float value between 0 and 1 and the second is without the normalization

V. Conclusion and Future Work

The conclusion of this research is that the CNN model with adequate preprocessing of thresholding and edge detection achieve a good result of 96.76% accuracy in the training and 97.93% in the validation set. By the analysis of the history, the model also proven to converge nicely without the problem of underfitting and overfitting

Even though the research achieves a good result, there are more room to grow through this classification. From here we can further develop a sign language recognition system using object detection based on the model. The sign language recognition system can also be implemented in mobile devices due to the light sized-model and the flexibility in tensorflow and keras' model export.

References

[1] S. He, "Research of a Sign Language Translation System Based on Deep Learning," in *Proceedings - 2019 International Conference on Artificial Intelligence and Advanced Manufacturing, AIAM 2019*, Oct. 2019, pp. 392–396. doi: 10.1109/AIAM48774.2019.00083.

[2] World Health Organization, "Deafness and hearing loss," Mar. 01, 2020. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Oct. 14, 2020).

[3] T. Handhika, R. I. M. Zen, Murni, D. P. Lestari, and I. Sari, "Gesture recognition for Indonesian Sign Language (BISINDO)," in *Journal of Physics: Conference Series*, Jun. 2018, vol. 1028, no. 1. doi: 10.1088/1742-6596/1028/1/012173.

[4] A. Riadi and P. Aditia, "Illustrated Book of Indonesian Sign Language for Deaf Children," Dec. 2017.

[5] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in *2018 Conference on Signal Processing And Communication Engineering Systems, SPACES 2018*, Mar. 2018, vol. 2018-January, pp. 194–197. doi: 10.1109/SPACES.2018.8316344.

[6] H. D. Yang, S. Sclaroff, and S. W. Lee, "Sign language spotting with a threshold model based on conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1264–1277, 2009, doi: 10.1109/TPAMI.2008.172.

[7] G. Gumelar, H. Hafiar, and P. Subekti, "Bahasa Isyarat Indonesia Sebagai Budaya Tuli Melalui

Pemaknaan Anggota Gerakan Untuk Kesejahteraan Tuna Rungu," *INFORMASI*, vol. 48, no. 1, p. 65, Jul. 2018, doi: 10.21831/informasi.v48i1.17727.

[8] L. Y. Bin, G. Y. Huann, and L. K. Yun, "Study of Convolutional Neural Network in Recognizing Static American Sign Language," in *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Sep. 2019, pp. 41–45. doi: 10.1109/ICSIPA45851.2019.8977767.

[9] R. Harini, R. Janani, S. Keerthana, S. Madhubala, and S. Venkatasubramanian, "Sign Language Translation," in *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, Mar. 2020, pp. 883–886. doi: 10.1109/ICACCS48705.2020.9074370.

[10] Md. S. Alom, Md. J. Hasan, and Md. F. Wahid, "Digit Recognition in Sign Language Based on Convolutional Neural Network and Support Vector Machine," Dec. 2019. doi: 10.1109/STI47673.2019.9067999.

[11] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," in *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, Jan. 2019, pp. 4896–4899. doi: 10.1109/BigData.2018.8622141.

[12] I. Makarov, N. Veldyaykin, M. Chertkov, and A. Pokoev, "Russian Sign Language Dactyl Recognition," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2019, pp. 726–729. doi: 10.1109/TSP.2019.8768868.