Indonesia Sign Language Recognition using Convolutional Neural Network

Suci Dwijayanti*, Hermawati, Sahirah Inas Taqiyyah, Hera Hikmarika, Bhakti Yudho Suprapto
Department of Electrical Engineering, Universitas Sriwijaya
Indralaya, Indonesia

***Abstract*—In daily life, the deaf use sign language to communicate with others. However, the non-deaf experience difficulties in understanding this communication. To overcome this, sign recognition via human-machine interaction can be utilized. In Indonesia, the deaf use a specific language, referred to as Indonesia Sign Language (BISINDO). However, only a few studies have examined this language. Thus, this study proposes a deep learning approach, namely, a new convolutional neural network (CNN) to recognize BISINDO. There are 26 letters and 10 numbers to be recognized. A total of 39,455 data points were obtained from 10 respondents by considering the lighting and perspective of the person: specifically, bright and dim lightning, and from first and second-person perspectives. The architecture of the proposed network consisted of four convolutional layers, three pooling layers, and three fully connected layers. This model was tested against two common CNNs models, AlexNet and VGG-16. The results indicated that the proposed network is superior to a modified VGG-16, with a loss of 0.0201. The proposed network also had smaller number of parameters compared to a modified AlexNet, thereby reducing the computation time. Further, the model was tested using testing data with an accuracy of 98.3%, precision of 98.3%, recall of 98.4%, and F1-score of 99.3%. The proposed model could recognize BISINDO in both dim and bright lighting, as well as the signs from the first-and second-person perspectives.**

Keywords*—Indonesia sign language (BISINDO); recognition; CNN; lighting

I. Introduction

Humans use language to communicate with others. However, a communication disorder may occur because of various factors that cause an impairment in understanding oral speech [1]. Such factors can arise from a hearing disorder or deafness. Thus, deaf people use sign language or hand gestures to communicate. However, most non-deaf people experience difficulties in understanding sign language. A computerized sign recognizer could be employed as an important tool to enable mutual understanding between deaf and non-deaf people.

Various studies have been proposed to recognize hand gestures or sign languages in different countries because each country has a different sign, such as the American sign language [2], Arabic sign language [3], Bengali sign language [4], Peruvian sign language [5], and Chinese sign language [6] using various methods.

Indonesia has two sign languages: Indonesia Sign Language System (SIBI) and Indonesia Sign Language (BISINDO). In 1994, SIBI became the language used in formal schools for students with impairments. However, the deaf prefer to use BISINDO instead of SIBI in their daily lives.

Certain studies have been performed to recognize the SIBI. Hand gestures recognition approaches can be divided into vision based and sensor-based [7]. In vision-based approaches, images are acquired through a video camera. Meanwhile, sensor-based recognition needs an instrument to capture the motion, position, or velocity of the hands. Studies in Indonesian sign languages implemented the vision-based approach. A. Anwar et al. used a leap motion controller to recognize Indonesian sign language using feature extraction captured from hand movement [8]. In [9], a Myo Armband tool was used, which has five sensors, namely accelerator, gyroscope, orientation, orientation Euler, and electromyography (EMG). Both vision and sensor-based approaches need the data acquisition and classification stages. Various classification methods have been proposed to recognize patterns carried by input data. The k-nearest neighbor classification method was used to recognize the SIBI [10]. In this study, the distance between the coordinates of each bone distal to the position of the palm was measured using Euclidean distance. Meanwhile, Khotimah et al. implemented weighted k-nearest neighbor classification for dynamic sign language recognition [11]. Rosalina et al. used artificial intelligence to recognize SIBI [12]. Other studies utilized Hidden Markov Model [13] and Naïve Bayes [14] methods. Meanwhile, [15] used the generalized learning vector quantization model to recognize BISINDO and [16] utilized Scale Invariant Features Transform (SIFT) algorithm to recognize Indonesian Sign Language numbers. Iqbal et al. implemented a mobile device using a Discrete Time Warping for recognizing SIBI [17].

Most studies above discussed SIBI; however, BISINDO is the most common sign language used by the deaf in Indonesia. Thus, this study aims to convert hand gestures to text in BISINDO to improve communications between deaf and non-deaf people. In addition, the methods used in other studies depended on feature extraction. To improve performance, this study proposes a method to recognize BISINDO using a convolutional neural network (CNN) which uses the convolution layer as the feature extraction layer [18]. In other studies, a CNN was used by [2] to recognize American Sign Language. They employed a CNN to extract the features from the sign images, and the classifier used was a multiclass support vector machine. Hayani et al. also utilized a CNN coupled with an Adam optimizer to recognize Arabic sign

*Corresponding Author.
E-mail: sucidwijayanti@ft.unsri.ac.id

language [3]. Hossen et al. used a deep convolutional neural network to recognize Bengali sign language [4].

However, not many previous works have addressed converting BISINDO sign language to text. Furthermore, there is a need to develop CNN models that have lower computation costs for converting sign language to text. This study addressed both needs by developing a new CNN architecture to perform the BISINDO hand gesture to text, and reduced computation costs by using fewer parameters than the common CNN architectures. The experimental research objective of this study was to compare the BISINDO recognition performance of this simplified CNN model to AlexNet and VGG-16 which are other architectures commonly used in CNNs. We tested the performance using BISINDO standard hand signs recorded by a webcam under bright and dim lightning, and from first and second-person perspectives.

This paper is organized as follows: Section II provides a brief summary of BISINDO, followed by a description of the CNN architecture in Section III. The methods used in this study are described in Section IV. The results and discussion are presented in Section V. Finally, the paper is concluded in Section VI.

II. Indonesian Sign Language

Sign language is a language that is expressed using body gestures and facial expressions as a symbol of the meaning of spoken language [19]. The sign languages of Indonesia can be categorized into two types: SIBI and BISINDO. SIBI was adopted from American Sign Language and is used as the formal sign language in schools for deaf students. However, the deaf prefer to use BISINDO instead of SIBI owing to its better applicability. The signs for the letters and numbers in the BISINDO language are shown in Fig. 1.

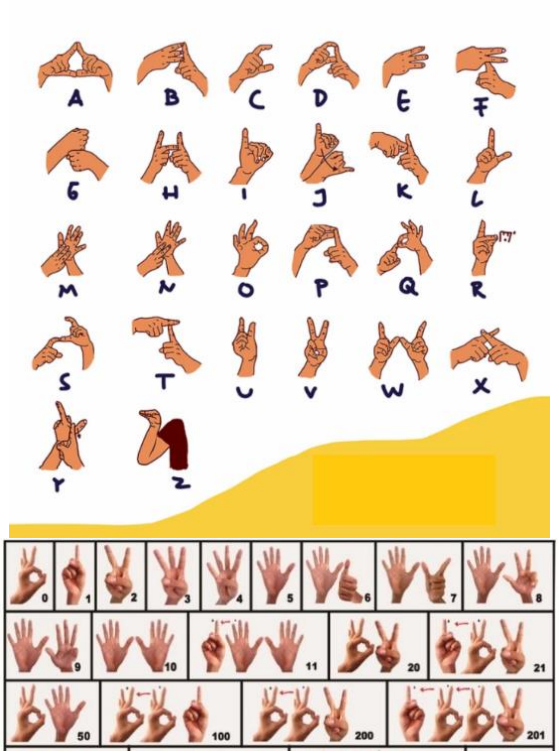


Fig. 1. BISIINDO Alphabets [20] and Numbers [21].

III. Convolutional Neural Network (CNN)

A CNN is typically used to detect or recognize images. It has an architecture that consists of a feature extraction layer and a fully connected layer. The feature extraction layer comprises a convolution layer and pooling layer. The general architecture of the CNN is illustrated in Fig. 2.

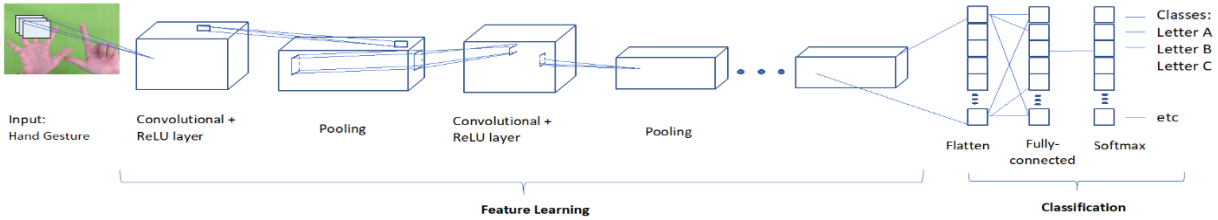


Fig. 2. Architecture of the CNN

The convolution layer extracts the features of images. This results in a linear transformation from the input, which is suitable for the spatial information of the filter. The weights in this layer determine the kernel convolution. Thus, kernel convolution can be trained based on the CNN input. The pooling layer comprises a filter with a stride and a certain size that passes through the path in the feature map. It aims to reduce image size. There are two types of pooling layers: max pooling and average pooling. In this study, max pooling was utilized by determining the maximum value in the vector dimension. After passing the convolution and pooling layers, the output of this process is used as the input to the fully connected layer. However, before this process, the input must be converted into one dimensional data. Finally, the process is performed using Softmax. Softmax calculates the probabilities for all target classes to determine the classes based on the input [22].

IV. Methods

This section provides detailed descriptions of several steps used in our methods. This study was performed using primary data obtained from people who had no prior knowledge of sign language. Here is an overview of the steps. A webcam was used to gather sets of hand sign data from ten people to use as training data. The data were obtained by considering two conditions: lighting and perspective of the person. Then, a new CNN model was designed and trained, which was named model C. For comparison, we trained modified versions of AlexNet and VGG-16. Then, the three models were tested and evaluated against the test data.

A. Data

The data used in this study were obtained using the webcam Logitech C922 with a resolution of 1080p and 30 fps. Ten respondents were asked to perform hand gestures, which consisted of 26 letters and numbers from 1 to 10, adhering to the BISINDO standard. Data were acquired 30 cm from the camera, as shown in Fig. 3. A green screen was placed as a background to minimize noise. Data were obtained by considering two conditions: lighting and perspective of the person.

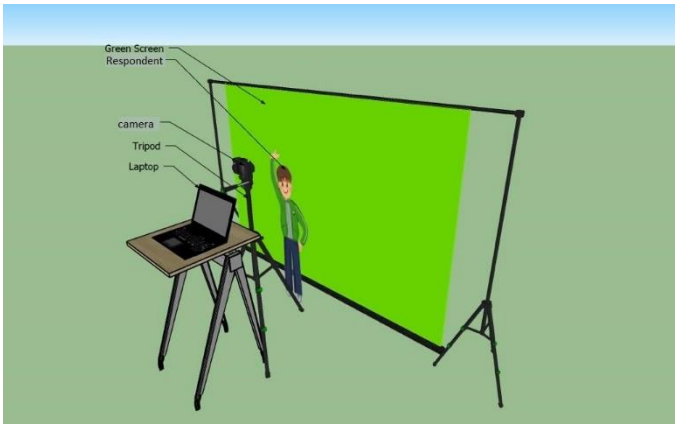


Fig. 3. Overview of the Data Retrieval Process.

B. Architecture of CNN

The CNN architecture used in this study consisted of three architectures, namely, models A, B, and C. Model A was a modified version of AlexNet [23]. The original AlexNet has 24,884,005 parameters, whereas the modified one has 1,432,261. Model B was a modified architecture of the VGG-16 [24]. It was modified to 2,140,405 parameters from its original value of 33,748,837. AlexNet and VGG-16 were chosen because they are the most common architectures used in CNNs. The architectures of models A and B are shown in Fig. 4 and 5, respectively.

This study proposed a new architecture, namely model C. Model C is a simpler architecture that consists of convolutional layer 1, max pooling 1, convolutional layer 2, convolutional layer 3, max pooling 2, convolutional layer 4, max pooling 3, flattened layer, and 3 fully connected layers. The visualization of model C is shown in Fig. 6.

C. Evaluation

This study utilized accuracy, precision, recall, and F1 scores to evaluate the performance of the three models. These parameters were calculated as follows:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

where True Positive (TP) is the number of positive data correctly predicted as positive, true negative (TN) is the number of negative data correctly predicted as negative, false positive (FP) is the number of negative data incorrectly predicted as positive, and false negative (FN) is the number of positive data incorrectly predicted as negative.

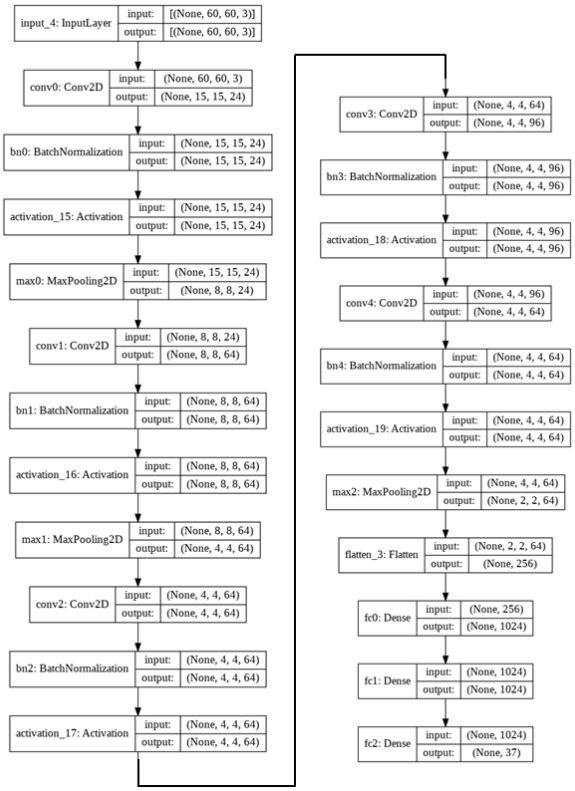


Fig. 4. Architecture of Model A.

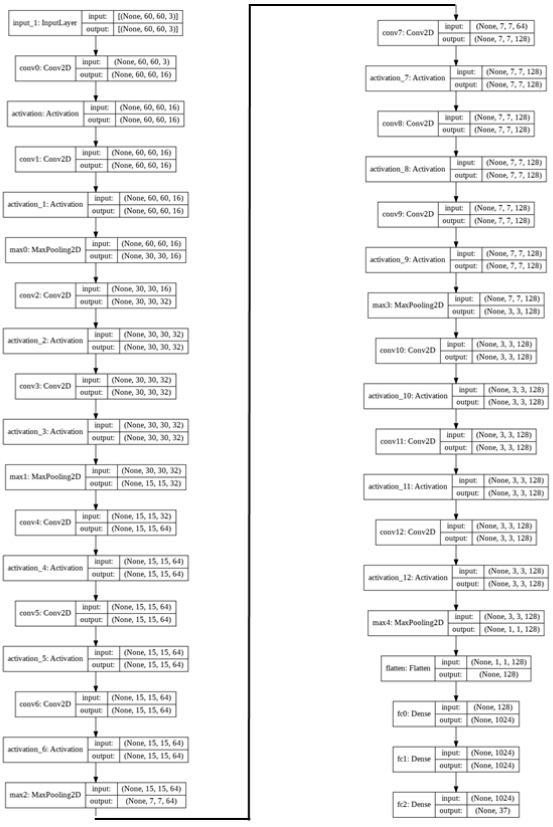


Fig. 5. Architecture of Model B.

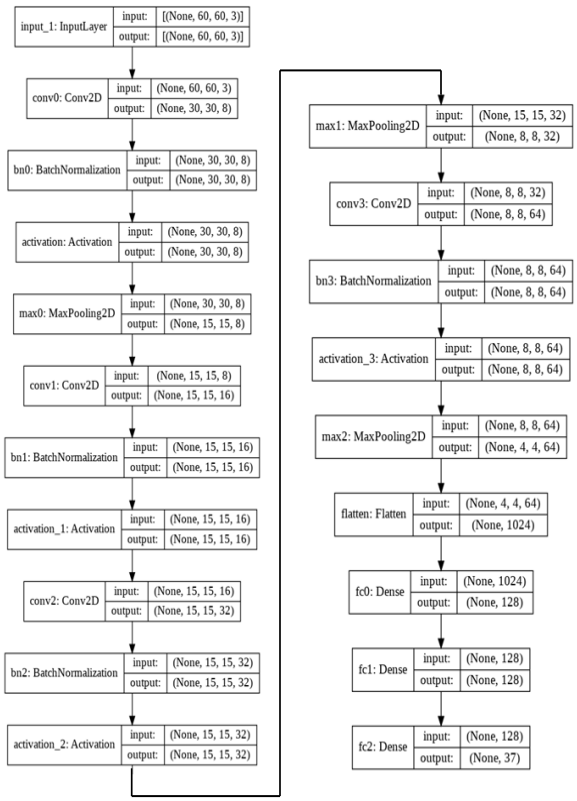


Fig. 6. Architecture of the New Model C.

In addition, precision and recall are also utilized as evaluation parameters. These can be calculated as.

$$precision = \frac{TP}{TP+FP} \quad (2)$$

and

$$recall = \frac{TP}{TP+FN} \quad (3)$$

The balance between precision and recall is determined using the F1-Score, which is obtained as follows.

$$F1\ score = 2\left(\frac{precision \times recall}{precision+recall}\right) \quad (4)$$

V. Result and Discussion

A. Image Dataset

The data used in this study were obtained from 10 respondents under two lighting conditions: dim and bright conditions. The position of the camera was also considered to be from the direction of the object considered (first-person perspective) and from the directions of others who observe the hand gesture (second-person perspective). Both lighting and viewpoints were considered in this study because illumination and viewpoints are challenges in gesture recognition [7]. Each respondent performed 37 hand gestures, consisting of 26 letters and 11 numbers (0–10). The data were recorded in a video format (.mp4) to obtain multiple data varieties. Subsequently, the data obtained were converted into images in the format of .jpg. The total data obtained through this process comprised 39,455 data points. Examples of the data are shown in Fig. 7.

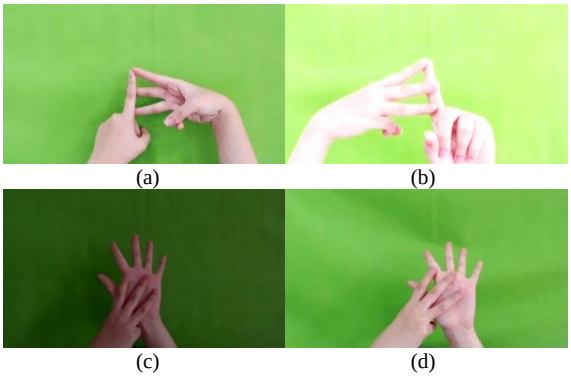


Fig. 7. Examples of Hand Gestures Obtained, (a) From the First-Person Perspective, (b) From the Second-Person Perspective, (c) Images Captured in Dim, and (d) Images Captured in Light.

B. Data Preprocessing

Before using the data in the CNN, the image data were preprocessed. This stage was performed by resizing the image and scaling the features. The image was resized to the same size of 60 × 60 pixels. Thereafter, feature scaling was performed by dividing the values at each point in the image by 255 such that the data value interval in the image was 0–1. Fig. 8 shows the preprocessed results of the image data.

C. Data Split

The preprocessed data were then fed as input to the CNN. In total 39,455 data were obtained, which was further divided using the stratified shuffle split method into three parts: training data, validation data, and test data. The division of the data was: 60 % training data, 20 % validation data, and 20 % test data, as shown in Fig. 9.

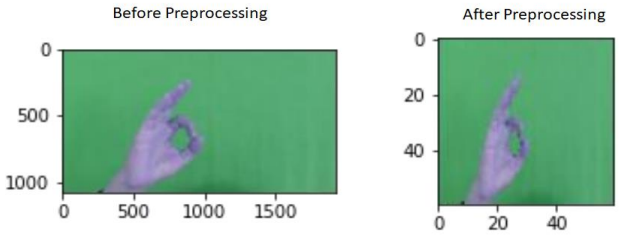


Fig. 8. Example of Preprocessed Result of Image Data.

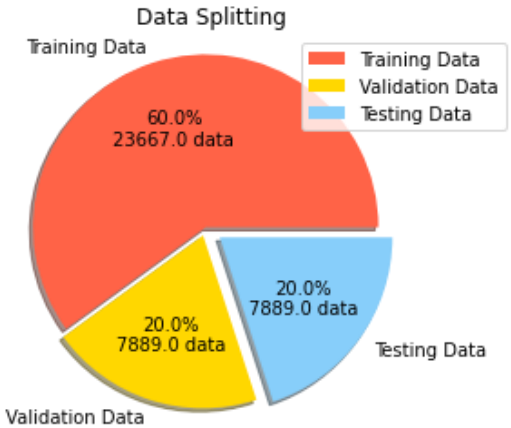


Fig. 9. Data Splitting.

D. Training Process

The training process was conducted using the CNN algorithm. The training parameters for the three models are listed in Table I.

TABLE I. PARAMETERS OF TRAINING

Parameter	Value
Image size	60 x 60
Optimizer	Adam
Epoch	100
Learning Rate	0.001

The loss and accuracy of the training results using models A, B, and C are shown in Fig. 10, 11, and 12, respectively.

As shown in Fig. 10, model A exhibited training and validation losses of 0.011 and 0.096, respectively. Further, the training and validation accuracies were 0.997 and 0.984, respectively. As shown in the loss graph, the model tends to fluctuate, indicating instability. Nevertheless, the model can learn the patterns as shown by the loss values, which tend to zero in each epoch, and the accuracy is improved. In contrast, model B has a high loss value and low accuracy, as shown in Fig. 11. This implies that the model cannot learn the patterns given by hand gestures because the loss values are high. Fig. 12 shows that model C has training and validation losses of 0.020 and 0.079, respectively. In addition, the training and validation accuracies were 0.995 and 0.984, respectively. Thus, model C can learn the hand gestures given because the loss value goes to zero and the accuracy increases. A comparison of these models is shown in Table II.

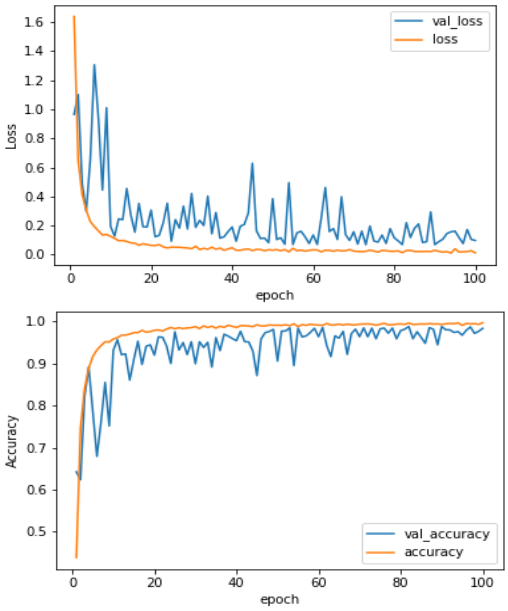


Fig. 10. Loss Value and Accuracy of Model A.

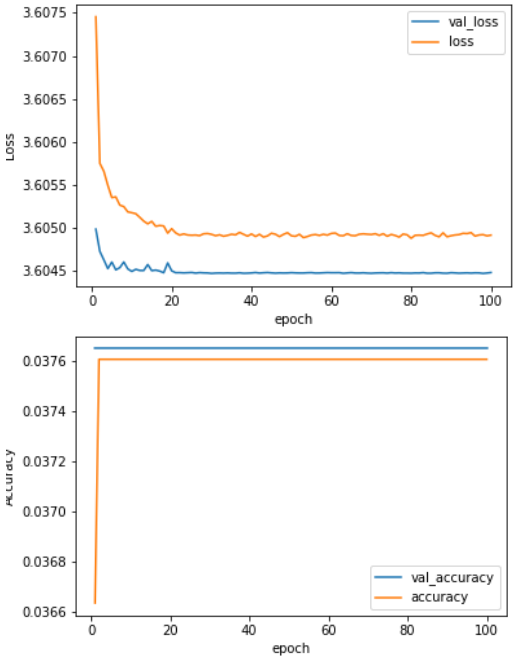


Fig. 11. Loss Value and Accuracy of Model B.

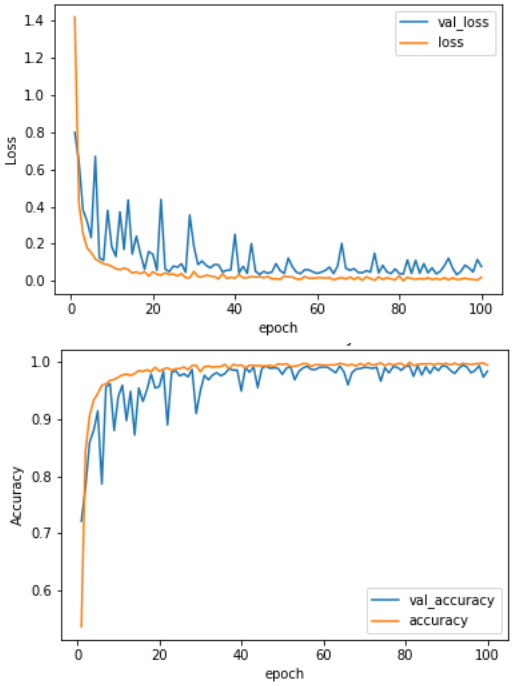


Fig. 12. Loss Value and Accuracy of Model C.

TABLE II. COMPARISON OF TRAINING IN MODEL A, B, AND C

Parameter	Model A	Model B	Model C
Training Loss	0.0113	3.6049	0.0201
Validation Loss	0.0967	3.6045	0.0785
Training Accuracy	0.9972	0.0376	0.9948
Validation Accuracy	0.9839	0.0376	0.9839
Total Parameter	1,432,261	2,140,405	177,373

As shown in Table II, Models A and C have low loss values and high accuracy compared to Model B. Overall, Model A has the lowest training loss value, and high training and validation accuracy. Model C has the lowest validation loss, and high training and validation accuracy. In addition, the total number of parameters used in Model C was 177,383 while Model A had 1,432,261 parameters. Therefore, the computation time in Model C was the smallest compared to the other models. In addition, although Model C still exhibited a fluctuation in validation loss and validation accuracy (Fig. 12), it is lesser than that of Model A (Fig. 10). Thus, Model C has more stable validation loss. Based on these results, Model C exhibited the best performance compared to the other models. Consequently, these models were used to test whether the model is optimal and can generalize the testing data.

E. Testing

Testing was performed after training to determine the ability of the model to predict the class of hand gestures. The test results are shown in Table III.

TABLE III. EVALUATION OF TESTING DATA

Model	Total Param.	Average prediction time per data (second)	Acc.	F1 Score	Precision	Recall
Model A	1,432,261	0.0002	0.986	0.996	0.987	0.987
Model B	2,140,405	0.0001	0.038	0.002	0.001	0.027
Model C	177,373	0.0001	0.983	0.993	0.983	0.984

As shown in Table III, model A has an accuracy of 0.986, F1 score of 0.996, precision of 0.987, and recall of 0.987. The results of testing using Model C are very similar to model A, with an accuracy of 0.983, F1 score of 0.993, precision of 0.983, and recall of 0.984. Since model B failed to learn, its test results were very low. Thus, Models A and C obtained the best results. However, Model C has fewer parameters, thereby requiring less time to predict the data compared to Model A. The average prediction time per data for Model C was half the time for Model A: 0.0001 s for Model C and 0.0002 s for Model A. Therefore, Model C is twice as efficient as Model A while achieving near-equivalent performance levels.

1) Test results by lighting: This study used two lighting conditions: bright and dim. The performances for both conditions are shown in Table IV.

As shown in Table IV, both Models A and C could recognize the testing data in the two different lighting conditions, and they both had high performance. Meanwhile, Model B performed poorly in recognizing the signs.

2) Test results by perspective: This study used the first- and second-person perspectives. The position of the camera was considered to be from the direction of the object considered (first-person perspective) and from the directions of others who observe the hand gesture (second-person perspective). The performances for both conditions are shown in Table V.

Table V shows that Model A and C can recognize the signs in both the first and second-person perspectives with high performance levels. There was a slight improvement with the second-person perspective.

F. Hand Gesture Prediction Results

The performance of the proposed model for predicting hand gestures was evaluated as well. Each class of hand gestures was performed, and the results obtained are shown in Fig. 13. The proposed model can recognize new data. Further, the hand gesture in the dim condition yielded a higher accuracy than in the light condition for the first-person perspective. In contrast, the second-person perspective exhibited the same performance under both dim and bright conditions. Certain samples of hand gesture recognition are listed in Table VI.

TABLE IV. TESTING RESULTS FOR DIFFERENT LIGHTING CONDITIONS

MODEL	Bright				Dim			
	Accuracy	*F1 Score*	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*Precision*	*Recall*
A	0.985	0.985	0.986	0.984	0.987	0.987	0.988	0.987
B	0.038	0.002	0.001	0.027	0.038	0.002	0.001	0.027
C	0.979	0.980	0.981	0.981	0.987	0.987	0.987	0.988

TABLE V. TESTING RESULTS FOR DIFFERENT PERSPECTIVES

MODEL	First-person perspective				Second-person perspective			
	Accuracy	*F1 Score*	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*Precision*	*Recall*
A	0.984	0.984	0.985	0.984	0.987	0.987	0.988	0.987
B	0.031	0.002	0.001	0.027	0.043	0.002	0.001	0.027
C	0.978	0.979	0.980	0.980	0.987	0.987	0.987	0.988

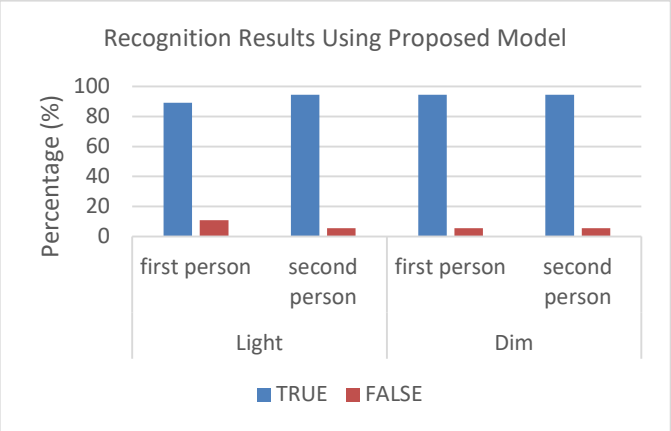
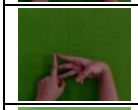
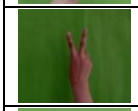
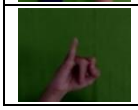


Fig. 13. Recognition Results using Proposed Model C.

TABLE VI. SAMPLE OF HAND GESTURES PREDICTION RESULTS

Data Test	Lighting Condition	Perspective	Actual Class	Result of Prediction
	Bright	First-person	4	4 (True)
	Dim	Second-person	H	H (True)
	Bright	Second-person	S	S (True)
	Dim	First-person	8	8 (True)
	Dim	First-person	B	B (True)
	Bright	Second-person	3	3 (True)
	Dim	First-person	2	V (False)
	Bright	First-person	M	M (False)
	Dim	First-person	N	M (False)
	Dim	Second-person	J	I (False)

As shown in Table VI, the proposed CNN model C works well in predicting hand gestures that were not included in the training data. This implies that the CNN can be implemented to recognize hand gestures. However, certain prediction errors occurred in certain classes, such as 2, M, N, V, and J. The occurrence of prediction errors due to hand gestures from these classes is almost the same or similar to other classes. The numbers 2 and V have the same hand gesture, thus, an error occurred in the CNN while predicting the class. From the first-person perspective, no difference was observed between the letter M and N hand gestures, thereby resulting in an error in the prediction. Further, the hand gesture for the letter J is not static, thus a prediction error occurred wherein the letter I was predicted because the initial movement of the signal letter J resembles that for the letter I.

VI. CONCLUSION

The results of this study demonstrated that our new simplified CNN model exhibited good performance in recognizing BISINDO hand gestures. The CNN architecture used was a simple architecture consisting of convolutional layer 1, max pooling 1, convolutional layer 2, convolutional layer 3, max pooling 2, convolutional layer 4, max pooling 3, flattened layer, and 3 fully connected layers. The parameters used were the Adam Optimizer, an iteration parameter of 100 epochs, and a learning rate of 0.001. During the training process, the last epoch resulted in a training loss value of 0.0201, validation loss value of 0.0785, and training accuracy value of 0.9948 with a validation accuracy value of 0.9839. The results of hand signal recognition testing using the CNN model on test data obtained performance results of 98.3%. Thus, this new simplified CNN model can recognize the BISINDO hand gestures well under dim and bright lighting and from the first- and the second-person perspective.

In the future, we will improve Model C to address those performance factors. We also expect to conduct the process of data retrieval with different backgrounds and do further research on real-time implementations of BISINDO hand gestures.

ACKNOWLEDGMENT

The research/publication of this article was funded by the DIPA of the Public Service Agency of Universitas Sriwijaya 2021. SP DIPA-023.17.2.677515/2021. In accordance with the rector's decree number, 0010/UN9/SK.LP2M.PT/2021 on April 28, 2021.

REFERENCES

[1] American Speech-Language-Hearing Association, "Definitions of communication disorders and variations." 1993. [Online]. Available: https://www.asha.org/policy/RP1993-00208/ [Accessed: August, 2021].

[2] M. R. Islam, U. K. Mitu, R. A. Bhuiyan, and J. Shin, "Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language," 2018 4th Int. Conf. Front. Signal Process. ICFSP 2018, pp. 115–119, 2018, doi: 10.1109/ICFSP.2018.8552044.

[3] S. Hayani, M. Benaddy, O. El Meslouhi, and M. Kardouchi, "Arab Sign language Recognition with Convolutional Neural Networks," Proc. 2019 Int. Conf. Comput. Sci. Renew. Energies, ICCSRE 2019, pp. 1–4, 2019, doi: 10.1109/ICCSRE.2019.8807586.

[4] M. A. Hossen, A. Govindaiah, S. Sultana, and A. Bhuiyan, "Bengali sign language recognition using deep convolutional neural network," 2018 Jt. 7th Int. Conf. Informatics, Electron. Vis. 2nd Int. Conf. Imaging, Vis. Pattern Recognition, ICIEV-IVPR 2018, pp. 369–373, 2019, doi: 10.1109/ICIEV.2018.86409622.

[5] B. Berru-Novoa, R. Gonzalez-Valenzuela, and P. Shiguihara-Juarez, "Peruvian sign language recognition using low resolution cameras,"

Proc. 2018 IEEE 25th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2018, 2018, doi: 10.1109/INTERCON.2018.8526408.

[6] S. Yuan, Y. Wang, X. Wang, H. Deng, S. Sun, H. Wang, and G. Li., "Chinese Sign Language Alphabet Recognition Based on Random Forest Algorithm," In 2020 IEEE Int. Workshop Metrology Industry 4.0 & IoT, pp. 340-344, June 2020, IEEE.

[7] M. J. Cheok, Z. Omar, M.H. Jaward, "A review of hand gesture and sign language recognition techniques," Int. J. Mach. Learn. Cybern. 10(1), 131-153, 2019.

[8] A. Anwar, A. Basuki, R. Sigit, A. Rahagiyanto, and M. Zikky, "Feature extraction for indonesian sign language (SIBI) using leap motion controller," In 2017 21st Int. Comput. Sci. Eng. Conf. (ICSEC) (pp. 1-5). November 2017. IEEE.

[9] A. Rahagiyanto, A. Basuki, and R. Sigit, "Moment invariant features extraction for hand gesture recognition of sign language based on SIBI," EMITTER Int. J. Eng. Technol. 5(1), 119-138, 2019.

[10] F. M. Humairah, Supria, D. Herumurti, and K. Widarsono, "Real Time SIBI Sign Language Recognition Based on K-Nearest Neighbor," In 2018 5th Int. Conf. on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 669-673). 2018.

[11] W. N. Khotimah, N. Suciati, Y.E. Nugyasa, and R. Wijaya,"Dynamic Indonesian sign language recognition by using weighted K-Nearest Neighbor," In 2017 11th Int. Conf. Inform. Commun. Technol. Syst. (ICTS) (pp. 269-274), Oct 2017, IEEE.

[12] Rosalina, L. Yusnita, N. Hadisukmana, R. B. Wahyu, R. Roestam, and Y. Wahyu, "Implementation of real-time static hand gesture recognition using artificial neural network," Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/CAIPT.2017.8320692.

[13] E. Rakun, M. I. Fanany, I. W.W. Wisesa, and A. Tjandra. "A heuristic Hidden Markov Model to recognize inflectional words in sign system for Indonesian language known as SIBI (Sistem Isyarat Bahasa Indonesia)." In 2015 Int. Conf. Technol. Inform. Manag. Eng. Environ. (TIME-E), pp. 53-58. IEEE, 2015.

[14] Ridwang, Syafaruddin, A. A. Ilham, and I. Nurtanio, "Indonesian Sign Language Letter Interpreter Application Using Leap Motion Control based on Naïve Bayes Classifier," IOP Conf. Ser. Mater. Sci. Eng., vol. 676, no. 1, 2019, doi: 10.1088/1757-899X/676/1/012012.

[15] T. Handhika, D.P. Lestari, I. Sari, and R.I.M. Zen, "The generalized learning vector quantization model to recognize Indonesian sign language (BISINDO)," In 2018 Third Int. Conf. Inform. Comput. (ICIC) (pp. 1-6). Oct 2018. IEEE.

[16] I. Mahfudi, M. Sarosa, R.A. Asmara, and M.A. Gustalika, "Indonesian Sign Language Number Recognition using SIFT Algorithm," In IOP Conf. Series: Mater. Sci. Eng. (Vol. 336, No. 1, p. 012010), April 2018. IOP Publishing.

[17] M. Iqbal, E. Supriyati, and T. Listiyorini, "SIBI Blue: Developing Indonesian Sign Language Recognition System Based On The Mobile Communication Platform," Int. J. Inform. Technol. Comput. Sci. Open Source, 1(1), 2017.

[18] Q. Liu, N. Zhang, W. Yang, S. Wang, Z. Cui, X. Chen, and L.Chen, " A review of image recognition with deep convolutional neural network,". In International conference on intelligent computing (pp. 69-80). Springer, Cham., August 2017.

[19] M. C. Stöppler, "Medical Definition of Sign Language," MedicineNet, 2021. [Online]. Availabe: www.medicinenet.com/sign_language/definition.htm. [Accessed: Feb. 02, 2021].

[20] Gerakan untuk Kesejahteraan Tunarungu Indonesia (Gerkatin) Solo, Bahasa Isyarat Alfabet BISINDO, [Alphabets in Indonesia Sign Language (BISINDO)] (in Indonesian) GERKATIN Solo, 2013. [Online]. Availabe: http://gerkatinsolo.or.id/ [Accessed: Feb. 02, 2021].

[21] Noviani, Bahasa Isyarat Angka BISINDO, [Number in Indonesia Sign Language (BISINDO] (in Indonesian) Penulis Cilik, 2019. [Online]. Available: https://www.penuliscilik.com/bahasa-isyarat-angka/ [Accessed: Feb. 02, 2021].

[22] S. Dwijayanti, R.R. Abdillah, H. Hikmarika, Z. Husin, and B.Y. Suprapto, "Facial Expression Recognition and Face Recognition Using a Convolutional Neural Network," In 2020 3rd Int. Seminar Res. Inform. Technol. Intell. Syst. (ISRITI) (pp. 621-626). Dec 2020, IEEE.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2012.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.