Deep Learning Approach For Sign Language Recognition

Bambang Krismono Triwijoyo, Lalu Yuda Rahmani Karnaen, Ahmat Adil
Bumigora University, Jl. Ismail Marzuki No.22, Mataram 83127, Indonesia

ARTICLE INFO

Article history:

Received October 16, 2022
Revised January 15, 2023
Accepted January 19, 2023

Keywords:

Deep Learning;
Sign Language;
CNN

ABSTRACT

Sign language is a method of communication that uses hand gestures between people with hearing loss. Each hand sign represents one meaning, but several terms don't have sign language, so they have to be spelled alphabetically. Problems occur when communicating between normal people with hearing loss, because not everyone understands sign language, so a model is needed to recognize sign language as well as a learning tool for beginners who want to learn sign language, especially alphabetic sign language. This study aims to create a hand sign language recognition model for alphabetic letters using a deep learning approach. The main contribution of this research is to produce a real-time hand sign language image acquisition, and hand sign language recognition model for Alphabet. The model used is a seven-layer Convolutional Neural Network (CNN). This model is trained using the ASL alphabet database which consists of 27 categories, where each category consists of 3000 images or a total of 87,000 hand gesture images measuring 200×200 pixels. First, the background correction process is carried out and the input image size is changed to 32×32 pixels using the bicubic interpolation method. Next, separate the dataset for training and validation respectively 75% and 25%. Finally the process of testing the model using data input of hand sign language images from a web camera. The test results show that the proposed model has good performance with an accuracy value of 99%. The experimental results show that image preprocessing using background correction can improve model performance.

Corresponding Author:

Bambang Krismono Triwijoyo, Universitas Bumigora, Jl. Ismail Marzuki No.22, Mataram 83127, Indonesia
Email: bkrismono@universitasbumigora.ac.id

1. INTRODUCTION

The research background is communication is very important in the process of social interaction. Communication leads to better understanding among the community, including the deaf [1]. Hand gesture recognition serves as the key to overcoming many difficulties and providing convenience for human life, especially for the deaf [2][3]. Sign language is a structured form of hand movement that involves visual movement and the use of various body parts namely fingers, hands, arms, head, body, and facial expressions to convey information in the communication process. For the deaf and speech-impaired community, sign language serves as a useful tool for everyday interactions [4]. However, sign language is not common among normal people, and only a few people understand sign language. This creates a real problem in communication between the deaf community and other communities, which has not been fully resolved to this day [3]. Not all words have sign language, so special words that do not have sign language must be spelled using a letter sign one by one [5]. Based on the background, this study aims to develop a sign language recognition model for letters of the alphabet using a deep learning approach. The deep learning approach was chosen because deep learning methods are popular in the field of computer science and are proven to produce a good performance for image classification [6][7]. The novelty of this study is the application of resizing and background correction of the input image for training and testing to improve model performance, where the results of testing the model we propose are better than previous similar studies.

The related work from past research is as follows. There have been many studies to recognize sign language, using various methods and varied datasets. Researchers [5] proposed a language recognition system using a 3D motion sensor by applying a k-nearest neighbor and support vector machine (SVM) to classify 26 letters in sign language. The average results of the highest classification levels of 72.78% and 79.83% were achieved by k-nearest neighbors and support vector machines, respectively. Based on previous studies that proposed hand sign language recognition models for alphabets, the results were not optimal, this is due to the complexity of lighting factors and other objects that appear in hand gesture images [5]. While there have been quite many studies on sign language recognition using a deep learning approach, here are several related studies including Study [8] has proposed a recognition system using a convolutional neural network (CNN) that can recognize 20 Italian gestures with high accuracy. Meanwhile, the following researchers introduced a sign language recognition (SLR) model using a deep learning approach. Study [9] implements transfer learning to improve accuracy. While study [10] has proposed the Restricted Boltzmann Machine (RBM) for automatic hand sign language recognition from visual data. The experimental results using four datasets show that the proposed multi-modal model achieves a fairly good accuracy. In the work [11], have proposed a deep learning-based framework for analyzing video features (images and optical flow) and skeletons (body, hands, and faces) using two sign language datasets. The results reveal the advantages of combining frame and video features optimally for SLR tasks.

A continuous deep learning-based sign language recognition model has also been introduced [12] which has proposed a 3D convolution residual network architecture and two-way LSTM, as a grammatical rule-based classification problem. The model has been evaluated on the benchmark of Chinese continuous sign language recognition with better performance. Other deep learning approach models have also been carried out for sign language recognition. A study [13] has proposed a ResNet50 Based Deep Neural Network architecture to classify finger-spelled words. The dataset used is the standard American Sign Language Hand gesture which produces an accuracy of 99.03%. While the study [14] Densely Connected Convolutional Neural Networks (DenseNet) to classify sign language in real-time using a web camera with an accuracy of 90.3%. The following studies [15]-[19] have implemented the CNN model for sign language recognition and tested using the American Sign Language (ASL) dataset, with an accuracy rate of 99.92%, 99.85%, 99.3%, 93%, and 99.91%.

Based on previous related studies, most of the sign language recognition methods use a deep learning approach. This study focuses on the introduction of hand sign language from the letters of the alphabet, which is used as a means of communication with the deaf. This study also uses the CNN model but with a different model architecture from previous studies. The CNN model was chosen because previous studies showed relatively better accuracy for image recognition [20][21][22]. The contributions of this research are: first, produce a real-time hand sign language image acquisition model by capturing each frame using a webcam video. Second, produce a hand sign language recognition model for the Alphabet, using a seven-layer CNN that has been trained using the ASL dataset and by applying resizing and background correction to the input image.

2. METHOD

This research is quantitative experimental research to measure the performance of hand sign language recognition models based on training datasets. Fig. 1 shows the proposed method for hand sign language recognition. In general, the proposed method consists of four stages, each of which is data acquisition, preprocessing, training and testing.

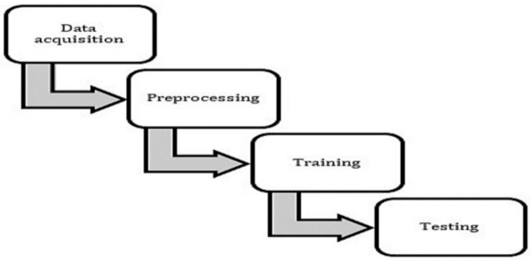


Fig. 1. Proposed Method

Based on the methodology applied in this study, as shown in Fig. 1, the first stage is data acquisition, where the data used in this study is the image. Image acquisition is the action of retrieving an image from an external source for further processing [23]. In this stage, the dataset used as model input is hand sign images, which are divided into 26 classes consisting of 26 sets of alphabets from A to Z. In this study, the dataset used as model input is hand sign images, which are divided into 26 classes consisting of 26 sets of alphabets from

A to Z. The second stage is preprocessing. At this stage, the image size transformation is carried out to reduce the complexity of the model architecture. In this study, the transformation of the image size of the training data images from the initial size of 200×200 pixels was resized to 32×32 pixels. In this study, we apply the bicubic interpolation method for resizing images as proposed by [24]. This resizing process is to reduce the computational time required for model training. To improve the segmentation accuracy of the hand sign language image Fig. 2 for complex lighting conditions, we apply a correction background method with luminance partition correction and adaptive threshold [25][26]. Furthermore, at the training stage, the CNN model architecture and its hyperparameters will be determined first. In this study, we use hyperparameter tuning to control the behavior of the machine learning model to produce optimal results. [27][28], then model training will be carried out using the dataset from the hand sign language image preprocessing. The last stage is model testing. At this stage, the model will be tested with hand sign language images in real-time using a webcam. The research proposed method flowchart is presented in Fig. 2.

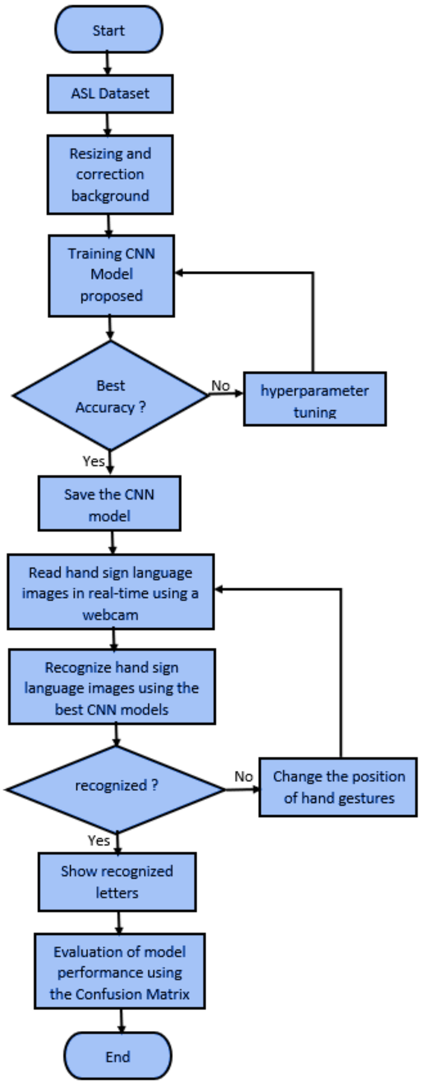

Start
ASL Dataset
Resizing and correction background
Training CNN Model proposed
Best Accuracy ?
No
hyperparameter tuning
Yes
Save the CNN model
Read hand sign language images in real-time using a webcam
Recognize hand sign language images using the best CNN models
recognized ?
No
Change the position of hand gestures
Yes
Show recognized letters
Evaluation of model performance using the Confusion Matrix
End

Fig 2. The Flowchart of Research Methodology

Furthermore, the results will be measured using a confusion matrix to determine the performance of the model [29]. Confusion matrices create result representations such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP means a positive result that is predicted by machine learning correctly. TN means negative outcome predicted by machine learning Correct. While FP means positive results predicted by machine learning are wrong, and FN means negative outcomes predicted by machine learning are wrong. Confusion Matrix Performance evaluation with a confusion matrix results in accuracy, precision, and recall [30][31]. Accuracy is the number of data points that machine learning correctly predicts among all data. Can calculate as (1):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision is the percentage of relevant elements that can indicate the number of times the model can predict correctly and can be calculated as (2).

$$Precission = \frac{TP}{TP + FP} \tag{2}$$

Meanwhile, recall is the percentage of relevant elements that are correctly classified by the model above all relevant elements. Recall calculation can be done using (3).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

3. RESULTS AND DISCUSSION

3.1. Data Acquisition

The data is secondary data downloaded from kaggle.com in the ASL Alphabet repository which contains image datasets. in JPG format containing 29 folders with each containing 3000 hand sign images. The total image data obtained amounted to 87,000 images [32]. Fig. 3 shows examples of hand sign language images sourced from the ASL Alphabet repository. The 87,000 images were distributed for the training process. The data is divided into two 75% for training and 25%. The train-test split is a technique for evaluating the performance of a machine-learning algorithm [33]. In the testing so for each class, there are 2,400 images for training and 600 images for testing, or a total of 69,600 images for training and 17,400 images for testing.

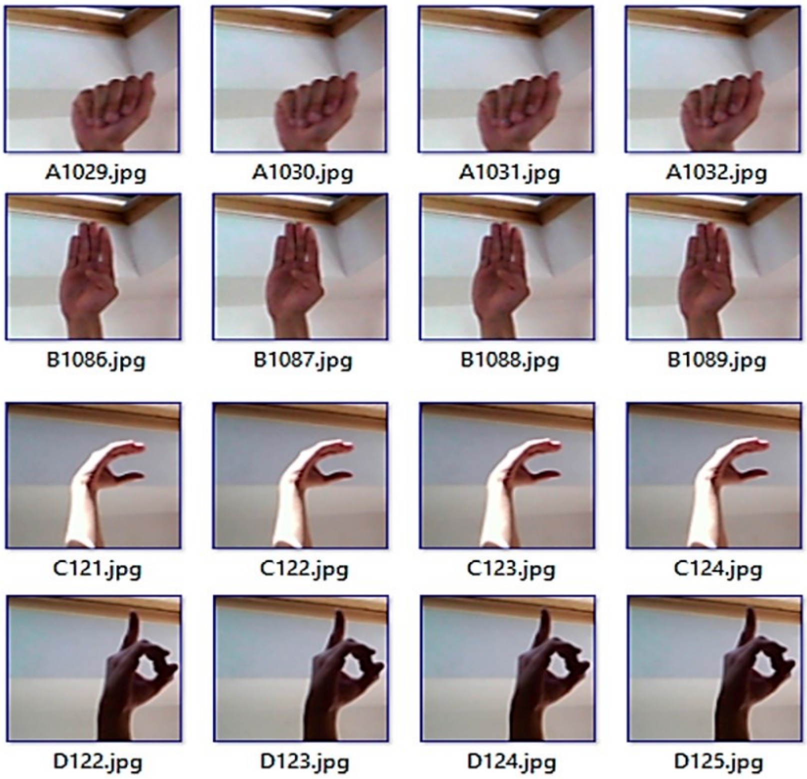


Fig. 3. Examples of hand sign language images from Kaggle

3.2. Preprocessing

At the preprocessing stage, the resizing process is carried out using the bicubic interpolation method [24], the results of this process produce an image size of 32×32 pixels, from the original image size of 200×200 pixels. This step is taken to reduce the time complexity during model training. the next preprocessing step is image background correction, to produce better accuracy using luminance partition correction and adaptive threshold [25]. Fig. 4 shows examples of preprocessing results.

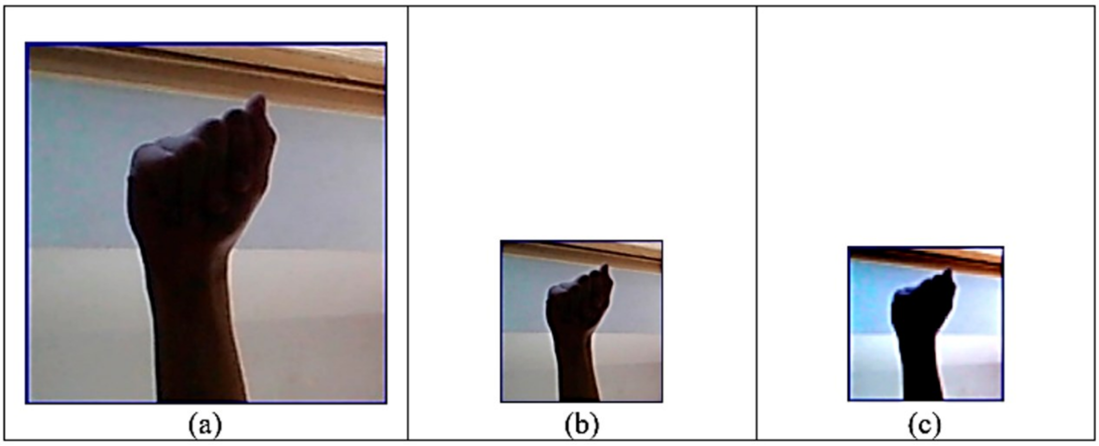


Fig. 4. (a). Original image size 200×200 pixels, (b). The result of resizing to a size of 32×32 pixels, (c). Background correction results.

In Fig. 4 it can be seen that the dimensions of the hand sign language image size for the training dataset are reduced in size as well as the increase in color brightness and contrast resulting from the background correction process.

3.3. Training

At this stage, the CNN model design used in the training process is carried out to produce an appropriate model to classify hand sign language images. The CNN model applied uses hyperparameter values consisting of the learning rate, epoch, loss function, and optimizer. Table 1 is CNN architecture specification used in this study.

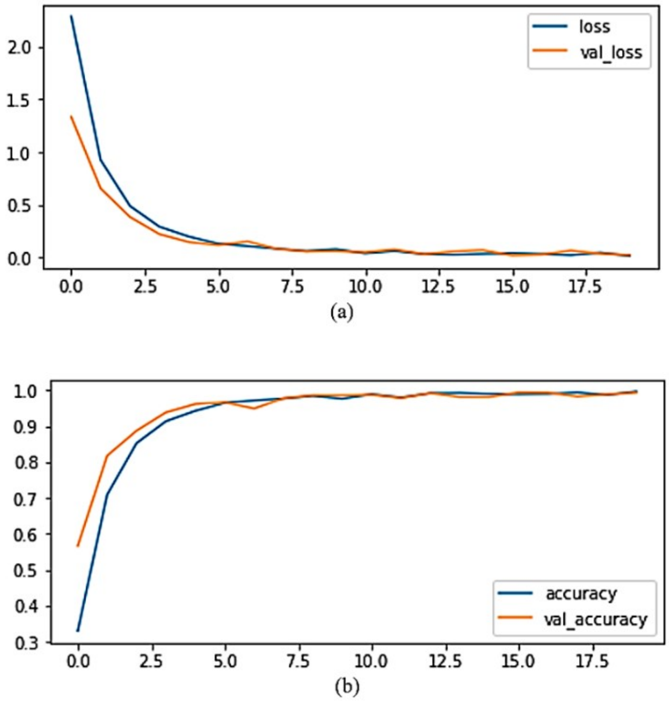
Table 1. CNN Model Architecture

Layer type	Description	Size
Input Layer	Input Image	32×32×3
Conv 1	Convolutional	8 kernel, 3×3
	ReLU	
	MaxPooling	Window size 2×2
Conv 2	Convolutional	16 kernel, 3×3
	ReLU	
	MaxPooling	Window size 2×2
Conv 3	Convolutional	32 kernel, 3×3
	ReLU	
	MaxPooling	Window size 2×2
Flatten	Flatten	512
Fully Connected Layer	Dense	512
	ReLU	
Output Layer	Dense Softmax	29

The CNN architectural design is as shown in Table 1, there are input layers, 3 convolution layers, flatten, fully connected layers, and output layers. The Input Layer requires input with a size of 32×32×3 where 32×32 pixels 3 layers RGB, then in Conv 1 using 8 kernels measuring 3×3 after that using ReLU activation [34], and using Maxpool with a window size of 2×2, as well as Conv 2 and Conv 3, only on Conv 2 uses 16 kernels and Conv 3 uses 32 kernels. Next is the Flatten layer and Fully Connected Layer with 512 nodes, and the last is the output layer containing Dense Softmax with 29 nodes [35], this is adjusted to the number of classes in the hand signal dataset of the alphabet.

The training process will be carried out through a series of iterations whose number of repetitions is determined by the maximum epoch value [36]. One epoch is a process when all training data has been used and passes through all network nodes once. While the hyperparameter used in this study is the Adam Optimizer

[37], with The Batch Size is 32 and Epochs are 20. The model training process is carried out in a hardware and software environment with specifications for Dell Latitude E7440 Laptop, 12 GB DDR4 RAM, Processor Intel® Core™ i5-7200U CPU @ 2.50GHz, Nvidia GeForce 940MX GPU, 256 GB SSD. The operating system used is Windows 10 Professional and the training and testing model algorithms are implemented using python code by utilizing the Tensorflow library. The results of the training model as shown in Fig. 5.



(a)

(b)

Fig. 5. Graphics of the Training Model Process: (a) Loss of Training and Validation, (b) Accuracy of Training and Validation

As seen in Fig. 5, The training process is carried out in 20 epochs, in the first epoch training accuracy is 33.04% and validation accuracy is 56.69%, while training loss is 28.26% and validation loss is 33.30%. in the tenth epoch, the training accuracy is 97.60% and validation accuracy is 98.57%, while the training loss is 8.11% and validation loss is 6.08%. Finally, in the last or twentieth epoch, the training accuracy is 99.60% and the validation accuracy is 99.68%, while the training loss is 1.64% and the validation loss is 2.55%. The main finding of this research is that the use of resizing and background correction methods, as well as setting hyperparameters can improve model accuracy.

3.4. Testing

After the model training process is complete, the next process is to test the model. The testing process is similar to the model training process, only the difference is that when the model testing process is not carried out backward pass or backpropagation iterations it does not change the weight or weight of the model as in the training process [38]. The Testing process is carried out using test data that is different from the training data set, to obtain valid testing results, this is following the recommendations of [39][40] regarding training and testing of the CNN model. Model testing is done by reading the hand signal image that is inputted by taking each frame from the webcam video, then the frame is identified based on the model that has been trained, then produces output in the form of identification results and accuracy values. Then the highest value from the identification results is directly written on the output board. Fig. 6 is a flowchart for the model testing process.

As shown in Fig. 6, the testing process is carried out by entering input in the form of hand signals via the Web camera on the Laptop Computer, where the hand makes a hand gesture in the Region of interest box on the Webcam display. Then the webcam display brings up the alphabetical prediction of the hand signal along with the score on the board display. Testing the model is carried out for each character letter 10 times, so a total of 290 times of testing. Fig. 7 and Fig. 8 Show the Confusion matrix value from the model testing results, for 29 types of hand sign language for each letter of the alphabet. Which consists of accuracy, precision, and recall.

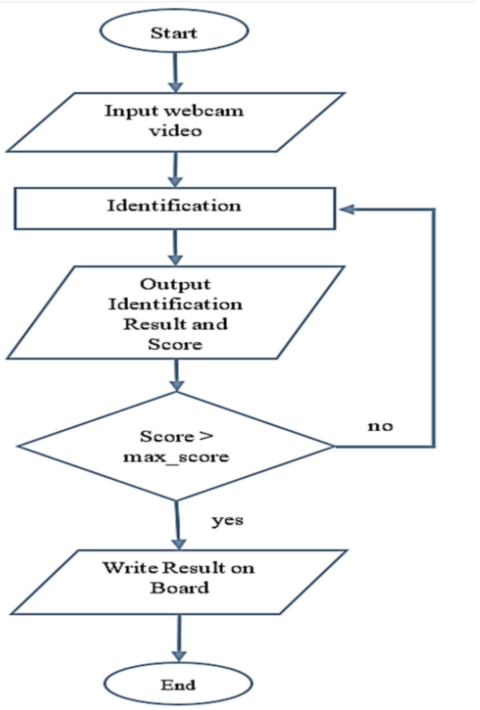


Fig. 6. Flowchart for the testing process

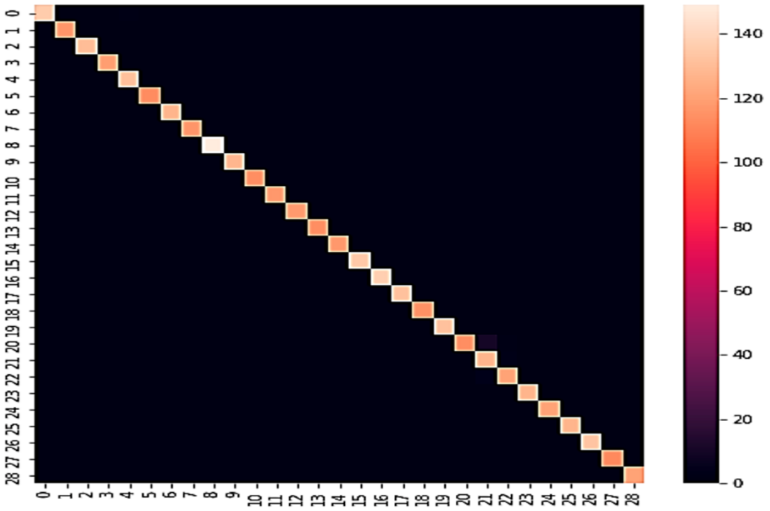
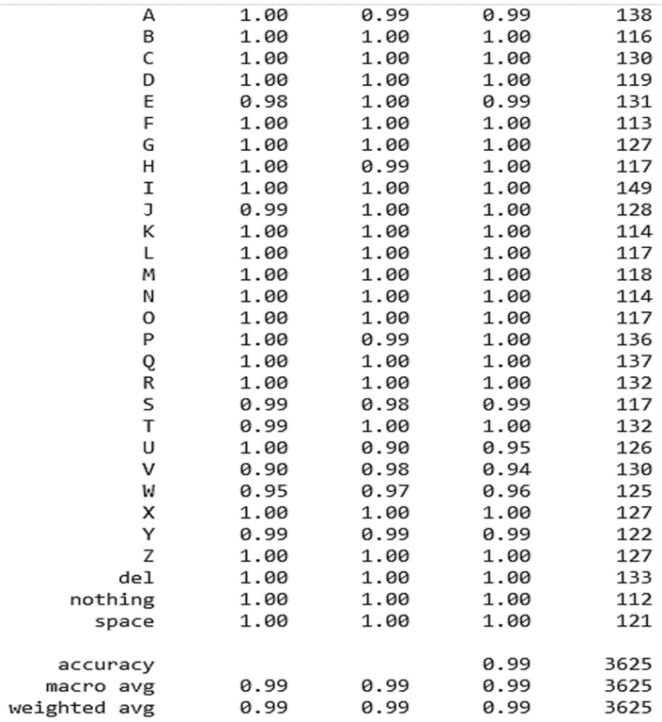


Fig. 7. Confusion Matrix of The Model Testing

Fig. 8 shows that the performance of the testing model achieves the best accuracy of 99%. and the Sensitivity, Recall, and f1-score levels are 99% respectively. This is because we apply resizing and background correction to the training and test images. These results are relatively better than previous similar studies as shown in Table 2.

The findings of this study imply that this hand sign language recognition model can be a hand sign language independent learning tool with a relatively better level of recognition accuracy than previous similar studies. The strength of this study is that the proposed hand sign language recognition model can perform hand sign language recognition from the alphabet in real-time. While the limitation of this model is that the performance of model is strongly influenced by the specifications of the web camera and lighting system.



```
           A       1.00      0.99      0.99       138
           B       1.00      1.00      1.00       116
           C       1.00      1.00      1.00       130
           D       1.00      1.00      1.00       119
           E       0.98      1.00      0.99       131
           F       1.00      1.00      1.00       113
           G       1.00      1.00      1.00       127
           H       1.00      0.99      1.00       117
           I       1.00      1.00      1.00       149
           J       0.99      1.00      1.00       128
           K       1.00      1.00      1.00       114
           L       1.00      1.00      1.00       117
           M       1.00      1.00      1.00       118
           N       1.00      1.00      1.00       114
           O       1.00      1.00      1.00       117
           P       1.00      0.99      1.00       136
           Q       1.00      1.00      1.00       137
           R       1.00      1.00      1.00       132
           S       0.99      0.98      0.99       117
           T       0.99      1.00      1.00       132
           U       1.00      0.90      0.95       126
           V       0.90      0.98      0.94       130
           W       0.95      0.97      0.96       125
           X       1.00      1.00      1.00       127
           Y       0.99      0.99      0.99       122
           Z       1.00      1.00      1.00       127
         del       1.00      1.00      1.00       133
     nothing       1.00      1.00      1.00       112
       space       1.00      1.00      1.00       121

    accuracy                           0.99      3625
   macro avg       0.99      0.99      0.99      3625
weighted avg       0.99      0.99      0.99      3625
```

Fig. 8. Performance of The Testing Model

Table 2. Comparison of Results with previous related studies

Study	Model	Dataset	Accuracy
Chong *et al.*[5]	SVM, DNN	American Sign Language (ASL)	80.30% and 93.81%
Pigou *et al.*[6]	CNN	Italian gestures	91.7%
Rastgoo *et al.*[8]	RBM	Massey University Gesture Dataset 2012	99.31%
Rathi *et al.*[11]	ResNet50 based	American Sign Language Hand gesture	99.03%
Daroya *et al.*[12]	CNN	American Finger Spelling format	90.03%
Abdulhussein *et al.*[15]	CNN	American Sign Language (ASL)	99.3%
Sabeenian *et al.*[16]	CNN	MNIST ASL dataset	93 %
Al-Hammadi *et al.*[17]	CNN	King Saud University Saudi Sign Language (KSU-SSL) dataset	87.69%
Our Method	CNN	ASL Alphabet repository	99%

4. CONCLUSION

In this study, a hand signal recognition model from letters of the alphabet using the CNN model has been successfully created, with significant results compared to previous related studies. Our contribution is the addition of preprocessing to the background correction which results in good accuracy in the proposed model. Our future work is to add to the sign language dataset basic words in addition to the letters of the alphabet, and also to increase the accuracy of the model by adding other hyperparameters.

REFERENCES

[1] L. K. S. Tolentino, R. O. S. Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, “Static Sign Language Recognition Using Deep Learning,” *Int. J. Mach. Learn. Comput.*, vol. 9, no. 6, pp. 821–827, 2019, https://doi.org/10.18178/ijmlc.2019.9.6.879.

[2] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, 2019, https://doi.org/ 10.1007/s13042-017-0705-5.

[3] R. Rastgoo, K. Kiani, and S. Escalera, “Hand sign language recognition using multi-view hand skeleton,” *Expert Syst. Appl.*, vol. 150, p. 113336, 2020, https://doi.org/10.1016/j.eswa.2020.113336.

[4] M. A. Hossen, A. Govindaiah, S. Sultana, and A. Bhuiyan, “Bengali sign language recognition using deep convolutional neural network,” in *2018 Joint 7th International Conference on Informatics, Electronics and Vision*

and 2nd International Conference on Imaging, Vision and Pattern Recognition, ICIEV-IVPR 2018, pp. 369–373, 2019, https://doi.org/10.1109/ICIEV.2018.8640962.

[5] T. W. Chong and B. G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors (Switzerland)*, vol. 18, no. 10, 2018, https://doi.org/10.3390/s18103554.

[6] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Z. Med. Phys.*, vol. 29, no. 2, pp. 86–101, 2019, https://doi.org/10.1016/j.zemedi.2018.12.003.

[7] H. I. K. Fathurrahman, A. Ma'arif, and L.-Y. Chin, "The Development of Real-Time Mobile Garbage Detection Using Deep Learning," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 3, p. 472, 2022, https://doi.org/10.26555/jiteki.v7i3.22295.

[8] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," in *European Conference on Computer Vision*, pp. 572–578, 2014, https://doi.org/ 10.1007/978-3-319-16178-5.

[9] S. Gattupalli, A. Ghaderi, and V. Athitsos, "Evaluation of deep learning based pose estimation for sign language recognition," *ACM Int. Conf. Proceeding Ser.*, pp. 1-7, 2016. https://doi.org/10.1145/2910674.2910716.

[10] R. Rastgoo, K. Kiani, and S. Escalera, "Multi-modal deep hand sign language recognition in still images using Restricted Boltzmann Machine," *Entropy*, vol. 20, no. 11, pp. 1–15, 2018, https://doi.org/10.3390/e20110809.

[11] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning approach for analyzing video and skeletal features in sign language recognition," *IST 2018 - IEEE Int. Conf. Imaging Syst. Tech. Proc.*, pp. 1-6, 2018. https://doi.org/10.1109/IST.2018.8577085.

[12] C. Wei, W. Zhou, J. Pu, and H. Li, "Deep grammatical multi-classifier for continuous sign language recognition," *Proc. - 2019 IEEE 5th Int. Conf. Multimed. Big Data, BigMM 2019*, pp. 435–442, 2019, https://doi.org/10.1109/BigMM.2019.00027.

[13] P. Rathi, R. K. Gupta, S. Agarwal, A. Shukla, and R. Tiwari, "Sign Language Recognition Using ResNet50 Deep Neural Network Architecture Pulkit," *Next Gener. Comput. Technol. 2019 Sign*, pp. 1–7, 2019, http://dx.doi.org/10.2139/ssrn.3545064.

[14] R. Daroya, D. Peralta, and P. Naval, "Alphabet Sign Language Image Classification Using Deep Learning," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 646–650, 2019, https://doi.org/10.1109/TENCON.2018.8650241.

[15] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta, and M. Aktaruzzaman, "A new benchmark on american sign language recognition using convolutional neural network," in *2019 International Conference on Sustainable Technologies for Industry 4.0, STI 2019*, pp. 1-6, 2019, https://doi.org/10.1109/STI47673.2019.9067974.

[16] M. R. M. Bastwesy, N. M. ElShennawy, and M. T. F. Saidahmed, "Deep Learning Sign Language Recognition System Based on Wi-Fi CSI," *Int. J. Intell. Syst. Appl.*, vol. 12, no. 6, pp. 33–45, 2020, https://doi.org/10.5815/ijisa.2020.06.03.

[17] A. Abdulhussein and F. Raheem, "Hand Gesture Recognition of Static Letters American Sign Language (ASL) Using Deep Learning," *Eng. Technol. J.*, vol. 38, no. 6, pp. 926–937, 2020, https://doi.org/10.30684/etj.v38i6a.533.

[18] R. S. Sabeenian, S. Sai Bharathwaj, and M. Mohamed Aadhil, "Sign language recognition using deep learning and computer vision," *J. Adv. Res. Dyn. Control Syst.*, vol. 12, no. 5, pp. 964–968, 2020, https://doi.org/10.5373/JARDCS/V12SP5/20201842.

[19] K. H. Rawf, "Effective Kurdish Sign Language Detection and Classification Using Convolutional Neural Networks," 2022, https://doi.org/10.21203/rs.3.rs-1965056/v1.

[20] M. Al-Hammadi *et al.*, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020. https://doi.org/10.1109/ACCESS.2020.3032140.

[21] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network for Hyperspectral Image Classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022, https://doi.org/10.1109/TIP.2022.3144017.

[22] Y. L. Chang *et al.*, "Consolidated Convolutional Neural Network for Hyperspectral Image Classification," *Remote Sens.*, vol. 14, no. 7, pp. 1–16, 2022, https://doi.org/10.3390/rs14071571.

[23] N. Kanwal, F. Perez-Bueno, A. Schmidt, K. Engan, and R. Molina, "The Devil is in the Details: Whole Slide Image Acquisition and Processing for Artifacts Detection, Color Variation, and Data Augmentation: A Review," *IEEE Access*, vol. 10, pp. 58821–58844, 2022, https://doi.org/10.1109/ACCESS.2022.3176091.

[24] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handb. Med. Image Process. Anal.*, pp. 465–493, 2009, https://doi.org/10.1016/B978-012373904-9.50037-4.

[25] J. Liao, Y. Wang, D. Zhu, Y. Zou, S. Zhang, and H. Zhou, "Automatic Segmentation of Crop/Background Based on Luminance Partition Correction and Adaptive Threshold," *IEEE Access*, vol. 8, pp. 202611–202622, 2020, https://doi.org/10.1109/ACCESS.2020.3036278.

[26] U. K. Acharya and S. Kumar, "Image sub-division and quadruple clipped adaptive histogram equalization (ISQCAHE) for low exposure image enhancement," *Multidimens. Syst. Signal Process.*, 2022. https://doi.org/10.1007/s11045-022-00853-9.

[27] S. S. Mostafa, F. Mendonca, A. G. Ravelo-Garcia, G. Julia-Serda, and F. Morgado-Dias, "Multi-Objective Hyperparameter Optimization of Convolutional Neural Network for Obstructive Sleep Apnea Detection," *IEEE Access*, vol. 8, pp. 129586–129599, 2020, https://doi.org/10.1109/ACCESS.2020.3009149.

[28] K. Krishnakumari, E. Sivasankar, and S. Radhakrishnan, "Hyperparameter tuning in convolutional neural networks

for domain adaptation in sentiment classification (HTCNN-DASC)," *Soft Comput.*, vol. 24, no. 5, pp. 3511–3527, 2020, https://doi.org/10.1007/s00500-019-04117-w.

[29] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, https://doi.org/10.1016/j.patcog.2019.02.023.

[30] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," *IEEE Access*, vol. 8, pp. 90847–90861, 2020, https://doi.org/10.1109/ACCESS.2020.2994222.

[31] M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, and A. Ralescu, "Confusion-matrix-based kernel logistic regression for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1806–1819, 2017, https://doi.org/10.1109/TKDE.2017.2682249.

[32] A. Nagaraj, "ASL Alphabet." 2022, Kaggle. https://doi.org/10.34740/kaggle/dsv/29550 (Accessed July. 8, 2022)

[33] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *J. Anal. Test.*, vol. 2, no. 3, pp. 249–262, 2018, https://doi.org/10.1007/s41664-018-0068-2.

[34] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "RMAF: Relu-Memristor-Like Activation Function for Deep Learning," *IEEE Access*, vol. 8, pp. 72727–72741, 2020, https://doi.org/10.1109/ACCESS.2020.2987829.

[35] M. A. Parwez, M. Abulaish, and Jahiruddin, "Multi-Label Classification of Microblogging Texts Using Convolution Neural Network," *IEEE Access*, vol. 7, pp. 68678–68691, 2019, https://doi.org/10.1109/ACCESS.2019.2919494.

[36] D. Mahapatra and Z. Ge, "Training data independent image registration using generative adversarial networks and domain adaptation," *Pattern Recognit.*, vol. 100, p. 107109, 2020, https://doi.org/10.1016/j.patcog.2019.107109.

[37] Y. Wang, J. Liu, J. Misic, V. B. Misic, S. Lv, and X. Chang, "Assessing Optimizer Impact on DNN Model Sensitivity to Adversarial Examples," *IEEE Access*, vol. 7, pp. 152766–152776, 2019, https://doi.org/10.1109/ACCESS.2019.2948658.

[38] T. R. Gadekallu *et al.*, "Hand gesture classification using a novel CNN-crow search algorithm," *Complex Intell. Syst.*, vol. 7, no. 4, pp. 1855–1868, 2021, https://doi.org/10.1007/s40747-021-00324-x.

[39] H. Chu, X. Liao, P. Dong, Z. Chen, X. Zhao, and J. Zou, "An automatic classification method of well testing plot based on convolutional neural network (CNN)," *Energies*, vol. 12, no. 15, 2019, https://doi.org/10.3390/en12152846.

[40] X. Lei, H. Pan, and X. Huang, "A dilated cnn model for image classification," *IEEE Access*, vol. 7, pp. 124087–124095, 2019, https://doi.org/10.1109/ACCESS.2019.2927169.

BIOGRAPHY OF AUTHORS

Bambang Krismono Triwijoyo obtained his Bachelor of Informatics, STIKI Malang - Indonesia in 1992. He continued his Master in Informatics Engineering from the Sepuluh Nopember Institute of Technology (ITS), Surabaya - Indonesia. He obtained his Master's in Computer Engineering in 2003. He obtained his doctorate in Computer Science doctoral program, at Bina Nusantara University, Jakarta - Indonesia. He is an assistant professor in the computer science study program, at Bumigora University until now. He is a member of the Indonesian Pattern Introduction (INAPR), and the Indonesian Informatics Experts Association (IAII). His research interests are in image processing, pattern recognition, and artificial intelligence. Email: bkrismono@universitasbumigora.ac.id

Lalu Yuda Rahmani Karnaen received a bachelor's degree from the Department of Computer Science, Bumigora University Mataram, Indonesia, in 2022. Currently a junior full-stack developer. The person concerned is pursuing machine learning until now. Email: yuda.karnaen@universitasbumigora.ac.id

Ahmat Adil obtained his Bachelor of Informatics Engineering degree from Palapa Institute of Technology (ITP) Malang - Indonesia in 1995. He continued his Masters's in Computer Science at Bogor Agricultural Institute (IPB), Indonesia. He obtained his Masters's in Computer Science in 2006. He is an assistant professor in the computer science study program, at Bumigora University until now. . Her research interest is in the field of Geographical Information Systems. . Email: ahmat.adil@universitasbumigora.ac.id