

Recognition of Indonesian sign language using deep learning: convolutional neural network-based approach

Olivia Kembuan^{1,2}, Haryanto^{1,3}, Mochamad Bruri Triyono¹

¹Doctoral Program in Technology and Vocational Education, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia

²Informatics Engineering Study Program, Faculty of Engineering, University of Manado, Manado, Indonesia

³Educational Research and Evaluation, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia

Article Info

Article history:

Received Nov 28, 2024

Revised Sep 6, 2025

Accepted Oct 16, 2025

Keywords:

Convolutional neural network

Deep learning

Image recognition

Neural network

Sign language

ABSTRACT

This study focuses on developing an automatic Indonesian sign language (SIBI) recognition system using a convolutional neural network (CNN). Sign language is essential for communication among deaf and hard-of-hearing individuals, and automatic recognition helps improve accessibility and inclusivity. CNNs are chosen for their ability to learn image features automatically, eliminating manual extraction and improving classification accuracy. The SIBI dataset used contains 5,280 images of 26 letters, divided into training and validation sets. In early training, the model achieved low accuracy (3.63% training, 3.33% validation), but after five epochs, it significantly improved to 97.58% for training and 100% for validation.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Olivia Kembuan

Doctoral Program in Technology and Vocational Education, Universitas Negeri Yogyakarta
Yogyakarta, Indonesia

Email: oliviakembuan.2023@student.uny.ac.id

1. INTRODUCTION

Image recognition refers to the process of identifying and categorizing objects within an image. This technology supports a wide range of applications, including facial recognition, autonomous vehicles, medical diagnostics, and retail analytics [1]–[6]. The field has advanced significantly due to increasing computational power, the availability of extensive datasets, and breakthroughs in machine learning. This technique, used in computer vision and image processing, has evolved from traditional machine learning methods to sophisticated deep learning approaches. Various methods and approaches have been developed for image classification, ranging from traditional machine learning techniques to advanced deep learning models. Convolutional neural networks (CNNs) are the cornerstone of modern image recognition systems.

A CNN is a type of deep learning model specifically designed for analyzing structured grid data such as images [7]–[9]. A CNN is a mathematical construct that generally consists of three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two layers, the convolution and pooling layers, perform feature extraction, while the third layer, the fully connected layer, maps the extracted features into the result, such as classification. A convolution layer plays an important part in CNN, which is constructed of a stack of mathematical operations, such as convolution, a specialized sort of linear operation [7]. CNNs are particularly effective for tasks like image classification, object detection, and image segmentation due to their ability to learn spatial hierarchies of features automatically and adaptively [10]–[12]. The motivation for using CNNs in image classification stems from their ability to automatically

and adaptively learn spatial hierarchies of features through backpropagation. This reduces the need for manual feature engineering and significantly improves classification accuracy.

In parallel with advancements in image recognition, sign language recognition (SLR) has emerged as a critical application of deep learning, aiming to bridge communication gaps for the deaf and hard-of-hearing communities. In Indonesia, Indonesian sign language system (SIBI) serves as the formal sign language used in educational and governmental contexts. Despite its standardized status, research on SIBI recognition remains limited, especially in terms of publicly available datasets and deep learning models tailored to its unique linguistic characteristics.

Recent studies have begun to explore the application of deep learning techniques, such as CNNs and hybrid models for recognizing both static and dynamic SIBI signs. However, these efforts are relatively modest in scale and scope. In contrast, sign languages such as American sign language (ASL) [13], Indian sign language (ISL), British sign language (BSL), and *bahasa isyarat* Indonesia (BISINDO) [14], [15] have been studied more extensively. ASL has received significant attention, supported by large-scale datasets and the adoption of advanced architectures including CNNs, long short-term memory (LSTM) networks, and Transformer-based models for both isolated and continuous sign recognition. To contextualize the current study, Table 1 compares recent efforts across various sign languages, summarizing the key contributions and highlighting the novelty of this work in advancing SIBI-based recognition.

Table 1. Comparison of related studies in SLR

Reference	Model/technique	Language/ dataset	Task type	Dataset properties	Accuracy	Notes	
[14]	CNN+LSTM	BISINDO, Video	Recognition (static)	10 BISINDO signs (2 letters + 8 words), 720 p video used for testing	CNN: accuracy/18% loss LSTM: accuracy/41% loss CNN+LSTM: accuracy/17% loss	96% 86% 96%	Metrics: accuracy, loss, word error rate (WER), character error rate (CER)
[15]	Hidden Markov model (HMM) with Gaussian densities	BISINDO	Data acquisition using Microsoft Kinect Xbox (skeleton tracking)	25 BISINDO root words	Accuracy ranges from 56% to 76%	Data labeled per frame - Training/testing split using K-Fold (K=10)	
[16]	3D-CNN, bidirectional recurrent neural network (Bi- RNN), GRU, SoftMax, CTC loss	SIBI	Sequence-to- sequence recognition task.	3,006 original videos of 30 sentences in SIBI	Average WER across models: 88.79%	Combined 3D-CNN (for spatial- temporal feature extraction) and Bi- RNN (for sequence modeling)	
[17]	CNN	ASL	Static sign language alphabet recognition	-1. Public Dataset 1: 52,000 images 2. Public Dataset 2: 62,400 images 3. Custom ASLA Dataset: 104,000 images	Dataset 1 [18]: accuracy =99.41%, loss =0.0204 Dataset 2 [19]: accuracy =99.48%, loss =0.0210 ASLA (custom dataset): accuracy= 99.38%, loss =0.0250	Captured with laptop/smartphone cameras	
This work	Custom CNN	SIBI, 26 letters (static)	StaticSign recognition	5,280 images of 26 letters	97.58% for training and 100% for validation.	New dataset; high accuracy after training	

A significant portion of SLR research has utilized computer vision while the majority of SLR research employs vision-based methods using RGB images or videos, recent advancements have also introduced sensor-based approaches. These leverage tools such as the leap motion controller (LMC) or wearable gloves to capture fine-grained, three-dimensional motion data. Such systems offer benefits including high temporal resolution, depth sensing, and real-time feedback, making them well-suited for dynamic gesture recognition and embedded deployments. Some studies, for instance, applied extreme learning machines (ELM) and meta-learning techniques to recognize two-handed Turkish sign language (TSL) gestures using leap motion [20], [21]. Others have focused on optimizing SLR systems for low-power edge devices, demonstrating the potential for portable and efficient sensor-driven SLR [22]. Despite their

strengths, sensor-based systems often rely on specialized hardware, limiting accessibility in educational or resource-constrained settings. By contrast, the present study proposes a vision-based CNN model that operates solely on RGB images, eliminating the need for external sensors. This approach offers high classification accuracy while remaining cost-effective and scalable making it particularly suitable for deployment in schools, public institutions, and inclusive communication environments across Indonesia.

The purpose of this research is to develop the Indonesia sign language SIBI image recognition system by using CNN architecture. Sign language is a vital communication method for the deaf and hard-of-hearing community [18], [23], [24]. Automatic SLR systems can facilitate seamless communication, enhancing accessibility and inclusivity [19], [25], [26]. The study introduces a CNN architecture specifically optimized for recognizing SIBI. Unlike generic CNN models, the proposed architecture is fine-tuned to handle the unique characteristics of SIBI signs, ensuring higher recognition accuracy and robustness. The primary contributions of this work include the design of an efficient CNN architecture tailored for SLR systems and the creation of a robust dataset for training and evaluating the model.

2. METHOD

This section will be described the method to collect, preprocess, and process data that we used. This system purposed is using CNN that consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The diagram of this research methodology is shown in Figure 1. All experiments were conducted locally on a MacBook Air (2020) equipped with a 1.1 GHz Quad-Core Intel Core i5 processor, 8 GB LPDDR4X RAM, and Intel Iris Plus integrated graphics. The development environment included Python 3.9 and TensorFlow 2.10, along with supporting libraries such as Keras, NumPy, OpenCV, and Matplotlib. Model training and evaluation were performed in a Jupyter Notebook environment without GPU acceleration. As such, computational time varied based on background processes and system load, and precise benchmarking was not the focus of this study.

The convolution layer, a crucial component of the CNN, applies a convolution operation to the output of the preceding layer. This process forms the core mechanism of the CNN, enabling it to learn and extract essential features from input data. Convolution, in this context, involves repeatedly applying a set of learnable filters to capture spatial patterns, such as edges, shapes, and textures, which are critical for classification and recognition tasks. This iterative process is illustrated in Figure 2.

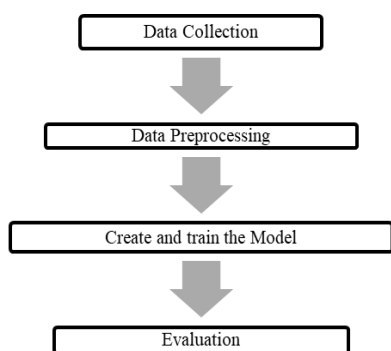


Figure 1. Diagram for research methodology

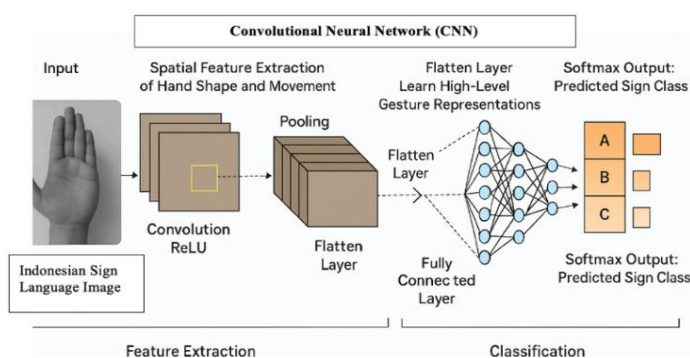


Figure 2. Convolutional neural network structure

2.1. Data collection

Data collection was carried out to obtain the information needed to achieve the objectives of the study. This phase begins by downloading the SIBI dataset from Kaggle and storing it in the local directory. This research utilizes the SIBI dataset of Indonesian sign language. SIBI is used because nearly all formal educational institutions that implement sign language utilize this form of sign language [27]. The SIBI dataset contains 5,280 images of static pose Indonesian sign language across twenty-six (26) categories of alphabets. Example images from the dataset are shown in Figure 3. The dataset, which is available on Kaggle, has a size of 2.7 GB. There are approximately 102 to 104 images per alphabet character in standard RGB format. A white paper is placed behind the hand sign as a background. Among the 5,280 images, we used 3,696 images (70%) for the training set, and 1,584 images (30%) for the validation set. The training

dataset is used to train the model while the validation dataset is used to monitor the working of the model which is not used during the training data. The validation dataset also helps to check whether the model is overfitting or not.

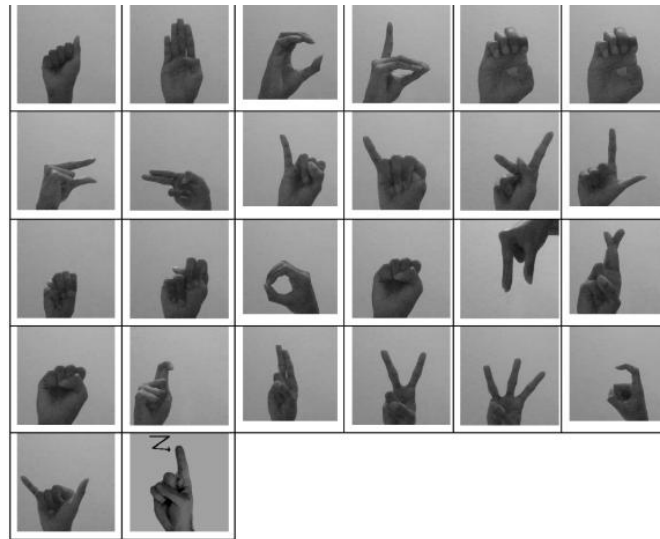


Figure 3. Example of dataset SIBI sign language

2.2. Data preprocessing

Image processing is a method that converts an image into a computer algorithm computation, this image processing technique is used for computers to find out the information in an image that has been done feature extraction process. The steps of the image processing technique are as follows [28]:

- Input images that have been taken through the scanner process or through the photo process directly.
- After inserting the image, the image analysis and manipulation process will be carried out, which consists of improving image quality, compressing image data, and designing image patterns.
- After the image pre-processing process, the image data will be converted back into an image that will be used in the classification process.

Data preprocessing is a crucial step in preparing your dataset for training a CNN. Proper preprocessing helps improve the quality and performance of your model. This phase begin by downloading the SIBI dataset from Kaggle and store it in the local directory. After extracting the contents, we assign variables with the file path for training dataset and validation data set. We assign variables with the file path for the training dataset and validation data set after extracting the contents. After establishing a 70% training dataset and a 30% validation dataset ratio, we save the photographs in various folders. The CNN model is trained using the train dataset. After making a set of predictions, the model was evaluated using the validation dataset. The next stage is to use the ImageDataGenerator class that tf.keras provides to decode the contents of these sign language images and transform them into floating point tensors. Data that has been divided into training data and validation data is then preprocessed such as rescale, rotation, and flip.

The purpose of this flipping process is to make padding easier when it is running in each process. At the next preprocessing stage, the rotation process is carried out where the face image will be rotated from the left or from the right. The scaling process stage will also be applied in the training set, with the aim that later neural networks can learn the features of the original scale.

2.3. Create and train the model

The structure of CNN used in this research shown in Figure 4. Pooling layers are responsible for reducing the dimensionality of feature maps, specifically the height and width, preserving the depth [29]. Max pooling outputs the maximum value of the elements in the portion of the image covered by the filter. Max pooling is better at extracting dominant features and therefore, considered more performant [30]. During the training process, the CNN is trained on a dataset and then evaluated on a separate validation dataset to monitor its performance. At this stage, a convolution operation is performed between the input matrix and the filter matrix. These filters will be shifted repeatedly over the entire image area, producing a feature map matrix as output. This feature map matrix can be calculated using the (1).

$$n_{out} = (\frac{n_{in}-k+2p}{s} + 1) \quad (1)$$

Where n_{out} is feature map size; n_{in} is matrix input size; p is padding's size; and s is stride.

In (2) is the convulsion operation formula of CNN:

$$FM[i]_{j,k} = (\sum_m \sum_n N_{[j-m,k-n]} f_{[m,n]} + bF) \quad (2)$$

Where $FM[i]$ is feature map matrix I ; N is input image matrix; F is convolution filter matrix; bF is bias value in filter; j, k is pixel position in the input image matrix; and m, n is pixel position in the convolution filter matrix.

After the convolution process is complete, the next step is to apply an activation function using the rectified linear unit (ReLU). The ReLU layer, or rectified linear unit layer, can be thought of as a thresholding process or activation function in artificial neural networks [31], [32]. Each pixel in the feature map will be input to the ReLU function, where pixels with values less than 0 will be converted to 0. The formula used for ReLU is $f(x) = \max(0, x)$.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 256)	295168
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	0
dropout (Dropout)	(None, 7, 7, 256)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 1024)	12846080
dense_1 (Dense)	(None, 26)	26650

Figure 4. Structure of conv CNN

2.4. Evaluation

This phase is to evaluate the accuracy of a CNN on both the training and validation datasets using TensorFlow/Keras and PyTorch. This process involves training the model, evaluating it on both datasets, and optionally plotting the accuracy values to visualize the model's performance over time. When evaluating a CNN, accuracy is a key metric that indicates the proportion of correct predictions made by the model out of all predictions.

3. RESULTS AND DISCUSSION

This section describes the results obtained from the implementation and discussion of the image recognition of sign language using the CNN. The results shown are the results of preprocessing, data modeling and training, and evaluation. The results are divided into accuracy testing and data loss testing.

3.1. Training and validation accuracy

The model is composed of four convolution blocks, as summarized in Figure 4, each of which has a max pool layer and is triggered by a ReLU activation function. The model begins with the first convolutional layer (conv2d), which employs 32 filters to extract initial features from the input images. This layer produces feature maps with dimensions of 148×148 and 32 channels. Next, the model includes a second convolutional layer (conv2d_1) with 64 filters, which extracts more complex features. The spatial dimensions are slightly

reduced to 72×72 . A second max pooling operation is applied through the `max_pooling2d_1` layer, further reducing the dimensions to 36×36 while maintaining 64 channels.

The architecture then incorporates a third convolutional layer (`conv2d_2`) with 128 filters, producing feature maps with dimensions of 34×34 and a greater depth. This step is followed by a third max pooling operation, which reduces the spatial dimensions to 17×17 . The final convolutional layer (`conv2d_3`) utilizes 256 filters to extract highly detailed features. The spatial dimensions are reduced to 15×15 , followed by a final pooling operation (`max_pooling2d_3`), which outputs feature maps with dimensions of 7×7 and a depth of 256.

To prevent overfitting, a dropout layer is applied, which randomly deactivates some neurons during training. Dropout is a CNN regularization technique that resolves neuronal interdependency. Overfitting of the data is a result of this interdependency. Poor predictions in a dataset can be caused by overfitting [33]. Afterward, the three-dimensional feature maps are flattened through the `flatten` layer, converting them into a one-dimensional vector of 12,544 features.

The model then connects to the first dense layer, comprising 1,024 neurons, which serves as a bridge between the extracted features and the final output. Lastly, the second dense layer acts as the output layer with 26 neurons, representing the 26 alphabet classes for classification. In total, the model has 13,259,234 trainable parameters, with the majority concentrated in the dense layers. This architecture is designed to efficiently classify alphabets with high accuracy by combining spatial feature extraction with deep learning. The function of pooling layers is to reduce the dimensionality of feature maps, meaning that the depth is preserved while the height and breadth are reduced [31]. Max pooling produces the maximum value within each region of the image encompassed by the filter. Max pooling is thought to be more efficient since it is more effective at extracting dominating features [34].

The final feature mappings are converted into a single 1D vector using the model's "Flatten" layer. After certain convolutional/maxpool layers, the flattening step is required in order to employ fully linked layers [20]. We used the softmax activation function in the final layer. SoftMax activation or SoftMax classifier is another form of logistic regression algorithm that we can use to classify more than two classes. Each class's output in SoftMax is constrained to a value between 0 and 1. This indicates the likelihood that an input is a member of a specific class. Using a batch size of 10 and five epochs of data, the CNN was trained for 100 steps per epoch. When the whole dataset runs through the training dataset, it is called an epoch.

The model is evaluated on the test dataset to check the accuracy. The training and validation accuracy and loss are plotted for visualization. The model trained for the dataset had initial training set and validation set accuracy of 3.63% and 3.33% in epoch-1. The validation accuracy ended up after 5 epochs with 97.58% accuracy for training set, and 100% accuracy for validation set. The accuracy for training set and validation set can be shown in Figure 5.

3.2. Training and validation loss

The model trained for the dataset had initial training set and validation set loss of 33.78% and 32.47% in epoch-1. The validation accuracy ended up after 5 epochs with 8.78% loss for training set, and 0.62% loss for validation set. The loss for training set and validation set can be shown in Figure 6.

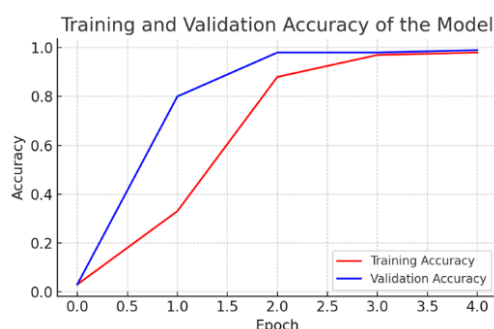


Figure 5. Training and validation accuracy graphic

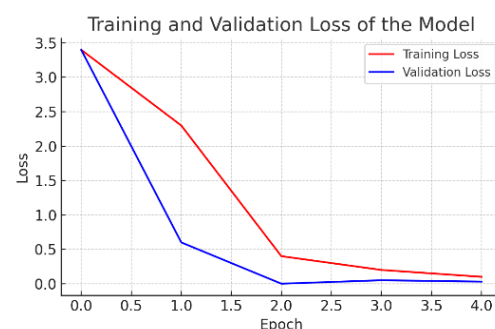


Figure 6. Training and validation loss

4. CONCLUSION

The initial training set and validation set accuracy for the model trained on the dataset was 3.63% and 3.33% in epoch-1. After five epochs, the validation accuracy was 97.58% for the training set and 100%

for the validation set. The model trained for the dataset had initial training set and validation set loss of 33.78% and 32.47% in epoch-1. The validation accuracy ended up after 5 epochs with 8.78% loss for training set, and 0.62% loss for validation set. The model performs better in testing when there is a greater supply of training data. Selecting the training data's batch size and epoch count is a crucial step in this study. This work presents a predictive model that is trained exclusively to recognize SIBI. The model can be improved in the future and still can be trained to recognize more characters and even for another language. The dataset as an input for this model with a lot of variations and can be effectively used to train the proposed model in order to increase its efficiency and accuracy as well. Some types of SIBI sign language characters require movement, therefore for further development a system that is able to recognize not only images but also videos is needed. In future work, computational time can be benchmarked more rigorously on a standardized GPU setup, enabling fairer comparisons across different models and datasets. However, the current results confirm the model's potential for efficient deployment in assistive technologies, particularly in educational and communication tools for the deaf and hard-of-hearing community.

ACKNOWLEDGMENTS

The Indonesia Endowment Funds for Education (LPDP) of the Republic of Indonesia, the Center for Educational Financial Services (PUSLAPDIK), and the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) are acknowledged by the authors for providing Indonesian Education Scholarships (BPI).

FUNDING INFORMATION

This research was funded by the Indonesia Endowment Funds for Education (LPDP) of the Republic of Indonesia under the Indonesian Education Scholarship (BPI) program [Grant No. 202329113295].

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Olivia Kembuan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Haryanto	✓	✓				✓				✓		✓		
Mochamad Bruri Triyono	✓			✓			✓			✓		✓	✓	✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**editing

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

DATA AVAILABILITY

The data that support the findings of this study are partly available from Kaggle at <https://www.kaggle.com/datasets/alvinbintang/sibi-dataset?resource=download>. Restrictions apply to the availability of these data, which were used under license for this study. In addition, this study also employed customized data generated by the authors. These customized data are available from the authors upon reasonable request, with the permission of Kaggle.




REFERENCES

- [1] E. Albalawi *et al.*, "Integrated approach of federated learning with transfer learning for classification and diagnosis of brain tumor," *BMC Medical Imaging*, vol. 24, no. 1, pp. 1–15, 2024, doi: 10.1186/s12880-024-01261-0.
- [2] H. Aouani and Y. B. Ayed, "Deep facial expression detection using Viola and Jones algorithm, CNN-MLP and CNN-SVM," *Social Network Analysis and Mining*, vol. 14, no. 1, 2024, doi: 10.1007/s13278-024-01231-y.
- [3] L. Wang, X. Kang, F. Ding, S. Nakagawa, and F. Ren, "A joint local spatial and global temporal CNN-Transformer for dynamic facial expression recognition," *Applied Soft Computing*, vol. 161, 2024, doi: 10.1016/j.asoc.2024.111680.
- [4] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision-ECCV 2014*, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.
- [5] C. Qingzheng, T. Qing, Z. Muchao, and M. Luyao, "CNN-based gesture recognition using raw numerical gray-scale images of surface electromyography," *Biomedical Signal Processing and Control*, vol. 101, 2025, doi: 10.1016/j.bspc.2024.107176.
- [6] N. Thakur, P. Kumar, and A. Kumar, "Multilevel semantic segmentation and optimal feature selection based convolution neural network (Op-CNN) for breast cancer identification and classification using mammogram images," *Biomedical Signal Processing and Control*, vol. 103, 2025, doi: 10.1016/j.bspc.2024.107374.
- [7] A. Patil and M. Rane, "Convolutional neural networks: An overview and its applications in pattern recognition," in *Smart Innovation, Systems and Technologies*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.
- [8] D. Gertsvolf, M. Horvat, D. Aslam, A. Khademi, and U. Berardi, "A U-net convolutional neural network deep learning model application for identification of energy loss in infrared thermographic images," *Applied Energy*, vol. 360, 2024, doi: 10.1016/j.apenergy.2024.122696.
- [9] S. Mohapatra, P. S. Jeji, G. K. Pati, M. Mishra, and T. Swarnkar, "Comparative exploration of deep convolutional neural networks using real-time endoscopy images," *Biomedical Technology*, vol. 8, pp. 1–16, 2024, doi: 10.1016/j.bmt.2024.09.003.
- [10] G. Xie, L. Wang, R. A. Williams, Y. Li, P. Zhang, and S. Gu, "Segmentation of wood CT images for internal defects detection based on CNN: A comparative study," *Computers and Electronics in Agriculture*, vol. 224, 2024, doi: 10.1016/j.compag.2024.109244.
- [11] L. Schneider, A. Krasowski, V. Pitchika, L. Bombeck, F. Schwendicke, and M. Büttner, "Assessment of CNNs, transformers, and hybrid architectures in dental image segmentation," *Journal of Dentistry*, vol. 156, 2025, doi: 10.1016/j.jdent.2025.105668.
- [12] S. Li and C. Huang, "Using convolutional neural networks for image semantic segmentation and object detection," *Systems and Soft Computing*, vol. 6, 2024, doi: 10.1016/j.sasc.2024.200172.
- [13] K. Bantupalli and Y. Xie, "American sign language recognition using deep learning and computer vision," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, doi: 10.1109/BigData.2018.8622141.
- [14] A. Aljabar and Suhartito, "BISINDO (Bahasa Isyarat Indonesia) sign language recognition using CNN and LSTM," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 282–287, 2020, doi: 10.25046/aj050535.
- [15] 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 2nd International Conference on Statistics, Mathematics, Teaching*, vol. 1028, no. 1, 2018, doi: 10.1088/1742-6596/1028/1/012173.
- [16] M. C. Ariesta, F. Wiryana, Suhartito, and A. Zahra, "Sentence level Indonesian sign language recognition using 3D convolutional neural network and bidirectional recurrent neural network," in *Proceedings of the 1st Indonesian Association for Pattern Recognition International Conference (INAPR)*, Jul. 2018, pp. 16–22, doi: 10.1109/INAPR.2018.8627016.
- [17] A. Kasapbaşı, A. E. A. Elbushra, O. Al-Hardane, and A. Yilmaz, "DeepASLR: A CNN-based human–computer interface for American sign language recognition for hearing-impaired individuals," *Computer Methods and Programs in Biomedicine Update*, vol. 2, Jan. 2022, doi: 10.1016/j.cmpbup.2021.100048.
- [18] M. S. Marcolino *et al.*, "Sign language recognition system for deaf patients: protocol for a systematic review," *JMIR Research Protocols*, vol. 14, 2025, doi: 10.2196/55427.
- [19] M. Alsulaiman *et al.*, "Facilitating the communication with deaf people: building a largest Saudi sign language dataset," *Journal of King Saud University – Computer and Information Sciences*, vol. 35, no. 8, 2023, doi: 10.1016/j.jksuci.2023.101642.
- [20] Z. Katılmış and C. Karakuzu, "ELM-based two-handed dynamic Turkish sign language (TSL) word recognition," *Expert Systems with Applications*, vol. 182, 2021, doi: 10.1016/j.eswa.2021.115213.
- [21] Z. Katılmış and C. Karakuzu, "Double-handed dynamic Turkish sign language recognition using Leap Motion with meta learning approach," *Expert Systems with Applications*, vol. 228, 2023, doi: 10.1016/j.eswa.2023.120453.
- [22] S. Siddique, S. Islam, E. Neon, T. Sabbir, I. Naheen, and R. Khan, "Deep learning-based Bangla sign language detection with an edge device," *Intelligent Systems with Applications*, vol. 18, Mar. 2023, doi: 10.1016/j.iswa.2023.200224.
- [23] E. S. Elfar, D. M. A. Kishk, A. M. Ibrahim, and S. E. Abdelraouf, "Silent lifesavers: Breaking barriers with a sign language health education video for students with deafness on school first aid," *International Journal of Africa Nursing Sciences*, vol. 20, 2024, doi: 10.1016/j.ijans.2024.100725.
- [24] M. Sanaullah *et al.*, "Sign language to sentence formation: a real time solution for deaf people," *Computers, Materials and Continua*, vol. 72, no. 2, pp. 2501–2519, 2022, doi: 10.32604/cmc.2022.021990.
- [25] B. Garg *et al.*, "Sign language detection dataset: a resource for AI-based recognition systems," *Data in Brief*, vol. 61, 2025, doi: 10.1016/j.dib.2025.111703.
- [26] E. P.-Montiel *et al.*, "Automatic sign language recognition based on accelerometry and surface electromyography signals: a study for Colombian sign language," *Biomedical Signal Processing and Control*, vol. 71, 2022, doi: 10.1016/j.bspc.2021.103201.
- [27] I. G. B. H. Widhinugraha and E. Rakun, "Indonesian language sign system (SIBI) recognition using threshold conditional random fields," *ICCPR '19: Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition*, 2019, pp. 380–384, doi: 10.1145/3373509.3373591.
- [28] A. K. S. and M. Gokilavani, "A study of medical image processing and segmentation methods," *International Journal of Innovative Research in Advanced Engineering*, vol. 10, no. 10, pp. 609–615, 2019, doi: 10.26562/IJRAE.2019.OCAE10082.
- [29] A. Derat, "Applied deep learning – part 4: convolutional neural networks," *Towards Data Science*, 2017. [Online]. Available: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.
- [30] S. Saha, "A comprehensive guide to convolutional neural networks – the ELI5 way," *Medium*, 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [31] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," in *Recent Trends in Signal and Image Processing*, vol. 172, 2019, pp. 519–528, doi: 10.1007/978-3-030-32644-9_36.
- [32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.




- [33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [34] M. Ibrahim, A. Shaawat, and M. Torki, "Covariance pooling layer for text classification," *Procedia Computer Science*, vol. 189, pp. 61–66, 2021, doi: 10.1016/j.procs.2021.05.070.

BIOGRAPHIES OF AUTHORS






Olivia Kembuan    received the B.Eng. degree in Informatics Engineering from the Satya Wacana Christin University, Central Java, Indonesia, in 2010, and the master's degree in Informatics Engineering from the Gadjah Mada University, Yogyakarta, Indonesia, in 2012. She is currently a lecturer of the Department of Informatics Engineering, University of Manado, Indonesia. Her current research interests include machine learning, computer networking, and augmented reality. She can be contacted at email: oliviakembuan@unima.ac.id.



Haryanto    is a lecturer in the Graduate School at Yogyakarta State University (UNY), Indonesia. His research interests centered on artificial intelligent control, educational research and evaluation, and technical and vocational education and training. He can be contacted at email: haryanto@uny.ac.id.



Mochamad Bruri Triyono    is a Professor in the Graduate School, Yogyakarta State University (UNY), Indonesia. His research expertise encompasses vocational education and training (VET), curriculum innovation, learning innovation, vocational teacher professional development, and partnerships between vocational schools and industry. He can be contacted at email: bruritriyono@uny.ac.id.