Research Article

SIBI Alphabet Detection System Based on Convolutional Neural Network (CNN) Method as Learning Media

Michael Arthur Limantara[1]*, Didik Tristianto [2]

[1] Department of Informatic Engineering, Universitas Narotama Surabaya, Indonesia

* Corresponding Author: star4michael@gmail.com

Abstract: Sign language is a form of communication that relies on body movements and facial expressions to interact, especially for deaf and hard-of-hearing people. The Indonesian Sign Language System (SIBI) is the official sign language in Indonesia. Until now, there is still a communication gap between deaf and hard-of-hearing people and normal people. The Computer Vision approach is expected to overcome the problem by developing a sign language recognition system. This research focuses on applying Deep Learning with the Convolutional Neural Network (CNN) method to detect hand gestures in SIBI alphabetic sign language and translate them. Hopefully, the results of this research can be the foundation for developing sign language recognition applications optimized specifically for SIBI. They can help people with disabilities and the general public communicate more effectively.

Keywords: Sign Language; Indonesian Sign Language System; Computer Vision; Deep Learning; Convolutional Neural Network; Hand Gesture Detection.

Citation: M.A,Limantara & D. Tristianto," SIBI Alphabet Detection System Based on Convolutional Neural Network (CNN) Method as Learning Media". *Iota,* **2024**, ISSN 2774-4353, Vol.04, 01. https://doi.org/10.31763/iota.v4i1.716

Academic Editor : Adi, P.D.P

Received : January, 15 2024

Accepted : January, 22 2024

Published : February, 27 2024

1. Introduction

Language has an essential role as a tool for communication in human social interaction. For those with disabilities, such as deafness or speech impairment, communication is often challenging due to difficulties in understanding and mastering spoken language or acquiring conventional speaking skills. One form of communication commonly used by individuals with deafness or speech impairment is sign language. However, not everyone understands sign language due to the infrequent use of sign language among people without disabilities. While using human interpreters can be effective, it is sometimes avoided due to cost considerations. This can limit communication between individuals with disabilities and the general public.

Sign language is a communication tool for individuals with special needs, such as those who are deaf and hard of hearing and who use body and hand movements as an alternative to communicating without sound [1]. This language relies on physical movements and hands to convey messages [2]. In Indonesia, there are two commonly used forms of sign language, namely Indonesian Sign Language (BISINDO) and Indonesian Sign Language System (SIBI) [3]. Indonesian Sign Language (BISINDO) is encouraged by the Indonesian Deaf Welfare Movement (GERKATIN) and developed by the deaf community, while the Indonesian Sign Language System (SIBI) is officially used in Sekolah Luar Biasa (SLB) under the Ministry of Education and Culture [1].

SIBI (Indonesian Sign Language System) is a communication medium for deaf people that combines spoken language, gestures, mimics, and other movements [4]. SIBI has 26 finger spellings that indicate 26 alphabets, with 24 signs in static movements and 2 signs in dynamic movements (J and Z). The SIBI alphabet is composed of a combination of finger and hand shapes [5]. SIBI was deliberately created and formalized by the Indonesian government to present Indonesian spoken grammar into artificial signs. Despite being an official standard, perceptual differences between sign language users and normal individuals are often a challenge in communication.

The application of Deep Learning in recognizing SIBI symbols opens up opportunities to strengthen the communication of deaf people in Indonesia. Deep Learning is a machine learning method with excellent Computer Vision capabilities. Computer Vision is a computer science that works by imitating human visual abilities [6]. Deep Learning, especially in Computer Vision, has demonstrated significant visual recognition capabilities, a key component in Sign Language Recognition (SLR) technology. With a significant deaf population in 2018 [7], [8], the development of SLR is essential in facilitating communication through sign language and gestures.

Image identification is an essential step in hand gesture recognition. Hand gesture recognition requires image processing and feature extraction to be performed to recognize and classify hand gestures with the maximum possible accuracy, where Convolutional Neural Network (CNN) comes as the main solution. CNN is used in this research to train the system to recognize SIBI letters. Hopefully, the application of CNN can improve recognition accuracy.

Convolutional Neural Network (CNN) is one of the machine learning methods that can be used for object image classification. CNN is a Deep Learning model that can mimic image recognition capabilities in the human visual cortex [9]. CNN's ability to recognize and classify objects is the main focus in hand gesture recognition, as found in SIBI sign language [10]. One of the advantages of CNN is that it is considered the best model for solving problems related to object detection and object recognition because it does not require large computations in the process. Despite being one of the best methods in object detection and recognition, CNN also has disadvantages related to the time-consuming duration of model training [11].

Based on these problems, in this research, a system is made to understand sign language by applying Deep Learning technology using the Convolutional Neural Network (CNN) method. This system makes Learning and understanding the Indonesian Sign Language (SIBI) alphabet easier. Some of the challenges faced include the data training process, object positioning, pose variations, lighting, and object background differences in the context of sign language recognition [12], [13]. With the SIBI sign language detection system, it is hoped that the gap in communication between deaf people and those without disabilities can be minimized.

2. Theory

2.1 SIBI

SIBI is the abbreviation of *'Sistem Isyarat Bahasa Indonesia.'* Sign language is an essential means of communication for deaf individuals to interact and convey information. This form of communication occurs through manual movements, including hand, lip, and body movements [16]. In the Kamus Besar Bahasa Indonesia, sign language is described as a system of communication that does not rely on human speech or writing but uses hand, head, and body movements, often used for the deaf and speech-impaired communities. Sign language focuses on visual communication and body expression, providing tools for the deaf community to communicate and obtain information. In sign language, hand and body movements convey essential concepts such as synonyms and antonyms. For example, words with similar meanings are transmitted through similar gestures, while antonym concepts can be expressed with different directions of movement.

System Sign Language Indonesia (SIBI) is one form of sign language in Indonesia. SIBI adapts gestures from American Sign Language and has 26 gestures that represent the 26 letters of the alphabet, including 24 passive gestures and two active gestures (J and Z) that use one hand. SIBI is used in formal communication, especially in special education (SLB), following established regulations. SIBI is recognized as an official language in Indonesia with structured sentences that pay attention to the use of subjects, objects, and predicates. SIBI continues to develop with the addition of local signs according to the needs of the deaf community in Indonesia. Figure 1 is the Indonesian Sign Language System (SIBI).

Figure 1. Indonesian Sign Language System (SIBI)

2.2 Deep Learning

Deep Learning is one of the techniques in the field of machine learning that has experienced rapid development in recent years. It utilizes complex neural networks to deeply process data and produce layered non-linear transformations [18]. In Deep Learning, input data undergoes a series of transformations through hidden layers designed to produce a more abstract representation of the data. In the context of Deep Learning, neural networks are characterized by the presence of hidden layers consisting of a large number of neurons. This deep neural network consists of a hierarchy of simple layers that are tasked with processing and transforming input data into increasingly abstract representations [19]. Each layer in this network plays a role in extracting increasingly complex features from the input data, thus allowing Deep Learning to understand highly complex and abstract data representations.

In addition, Deep Learning is known for its ability to automate feature extraction, allowing the system to determine relevant features from input data without requiring intensive human intervention to define those features. With this capability, Deep Learning continues to evolve and can tackle increasingly complex problems as the amount of data grows. Deep Learning has contributed significantly to technological advancements in various fields, including facial recognition, natural language processing, autonomous vehicles, and other applications. With the ability to understand increasingly complex data representations, Deep Learning opens up new opportunities for more sophisticated data analysis and understanding.

2.3 Convolutional Neural Network (CNN)

Nowadays, Convolutional Neural Network (CNN) technology is one of the significant approaches in two-dimensional data processing, especially in the context of image processing. CNN is part of the Deep Neural Network class, which focuses on analyzing and extracting information from digital images. The algorithms underlying CNNs are designed to recognize and understand the features present in digital images. The CNN architectural structure consists of layers of neurons with specific attributes such as bias, weight, and activation function [18], as shown in Figure 2.

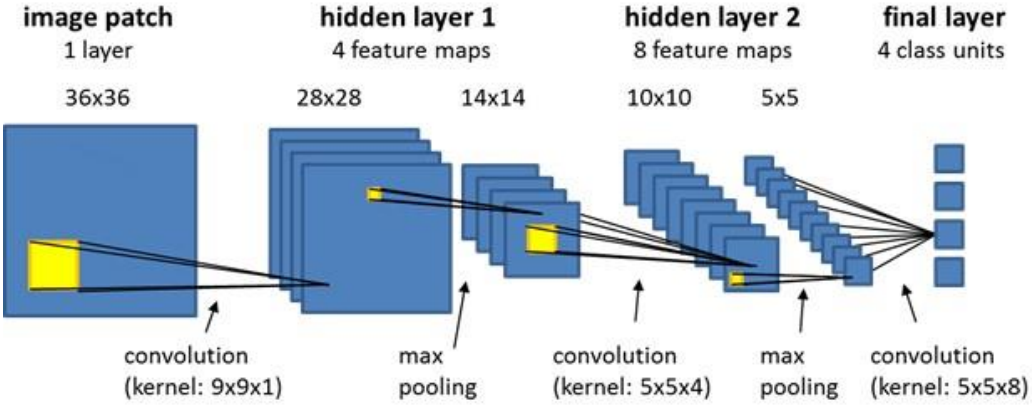

Figure 2. Convolutional Neural Network (CNN)

In Figure 2, there is a detailed illustration of the structure of the CNN method, which is divided into Input, feature extraction, and classification. The feature extraction layer contains several types of layers:

2.3.1 Convolutional Layer

The convolution layer is the initial layer that receives image input. Its operation consists of a convolution process that uses filters to extract information. By shifting the filter over the image, a matrix multiplication between the image and filter parts results in a 2-dimensional matrix.

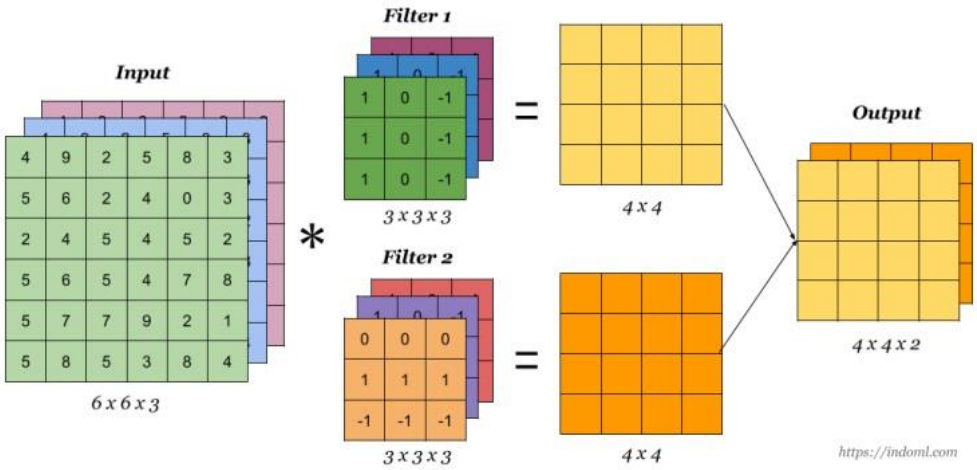

Figure 3. Illustration of Convolution Calculation

By shifting (convolving) the filter at every possible filter position on the image, an output is produced, commonly referred to as an activation map or feature map. The dimensions of the feature map can be calculated using a special equation 1.

$$Output\ f.maps = \frac{W-N+2P}{S} + 1 \quad (1)$$

Description: W= Input size, P = Padding, N= Filter Size, & S= Stride. Moreover, the feature map results in the convolution process will be normalized using the ReLu activation function with the formula in Equation 2.

$$\text{ReLu}\ (x) = \max\ (0, x) \quad (2)$$

Description: X = Input to the neurons in the neural network

2.3.2 Pooling Layer

The merging layer reduces the feature map's size by using various statistical operations based on the nearest pixel values. Regularly adding this layer after multiple convolution layers minimizes the number of parameters and controls overfitting. Overfitting is when the model tries to learn all the details, including noise in the data and tries to fit all the data points into the line. Using a 2x2 filter with step 2 and applying it to each input slice reduces the size of the feature map to 75% of its original size.

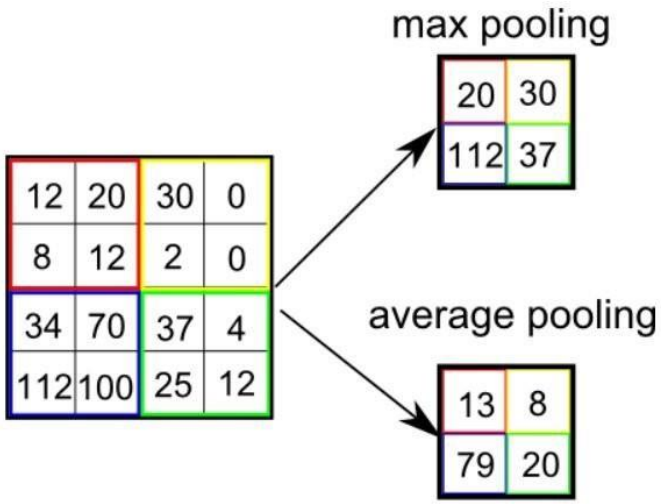

Figure 4. Pooling Illustration

2.3.3 Flatten Illustration

In the feature extraction stage, the result of the feature map is still in the form of a multidimensional array. To be integrated into the next stage, this array needs to be converted into one dimension, often called "flatten." This process is the first step before entering the Fully Connected Layer. This process converts the feature extraction matrix into a 1-dimensional matrix.

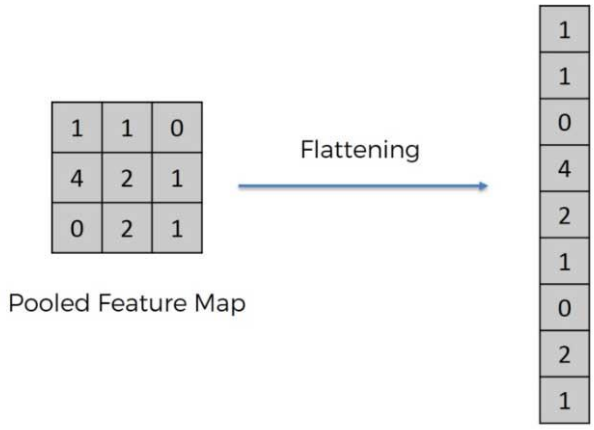

Figure 5. Flatten Illustration

2.3.4 Dropout

This step randomly removes neurons, which is considered noise, to prevent overfitting [18].

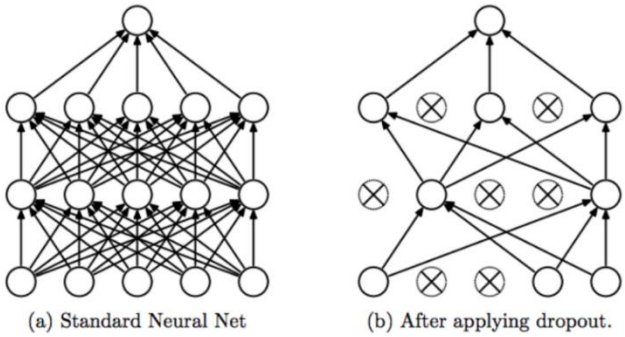

Figure 6. Dropout Illustration

2.3.5 Fully Connected Layer

The fully connected layer, also known as the fully connected layer, has characteristics similar to that of the multi-layer perceptron (MLP) and involves hyper-aligned parameters. These parameters include the hidden layer, activation function, output layer, and loss function. This stage involves two main processes, namely forward propagation and backpropagation [20].

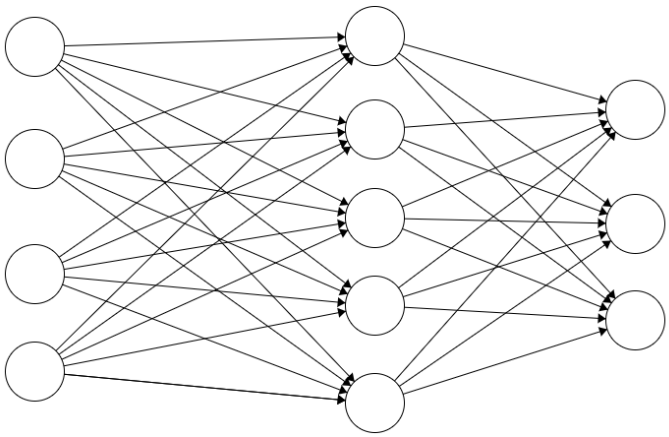

Figure 7. Fully Connected Illustration

2.3.6 Softmax Function

This function calculates the probability of the target class out of all target classes. This probability ranges from 0 to 1 [18]. The mathematical formulation of the softmax function can be found in Equation 3.

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (3)$$

Where x = Vektor input yang akan diubah menjadi probabilitas, i = Indeks dari elemen vektor x, e = Bilangan konstan Euler (2.71828…), and n = Jumlah elemen dalam vektor x.

2.4 Object or Motion Detection

Object detection is an essential aspect in the context of image processing and Computer Vision. Object detection is defined as a recognition process that can distinguish objects from the background and separate objects that may be suspicious [21]. This approach allows the system to recognize and separate particular objects more efficiently from their surroundings. Object detection involves identifying object instances in an image and determining the recognized object by the bounding box marks around it [22]. This approach provides a foundation for developing systems that identify and track objects in various visual situations.

2.5 Comparison with Previous Research

This research on CNN analysis of Sign Language is not new. Previous research shows significant progress in the application of methods to classify data. Table 1 shows the Comparison and Results of Previous CNN Sign Language Research.

Table 1. Comparison and Results of Previous CNN Sign Language Research

Previous Research	Results	Research Similarities/Differences
In 2018, Arfian adopted CNN to identify traditional transportation types	Produces an accuracy rate of 75%.	Different types of identification, i.e., sign language.
In 2018, Candra Kusuma Dewa et al. applied MLP and CNN methods to classify Javanese script handwriting.	CNN models have better accuracy compared to MLP models.	Applying the CNN Method to this research.
In 2019, Mochammad Bagus Setiyo Bakti et al. classified numbers in Indonesian sign language using the CNN method.	Achieved an accuracy of 98.89%.	Differences in the use of datasets that are SIBI alphabets.
In 2020, Devina Yolanda et al. classified alphabetic sign language by applying CNN and RNN methods.	Achieved an accuracy of 60.58%.	Does not combine CNN and RNN methods.
In 2022, Putri et al. integrated Mediapipe and LSTM to assess BISINDO sign language in real-time detection.	Achieved the highest accuracy of 92% for real-time detection.	Different methods for real-time detection of SIBI alphabetic sign language.

3. Method

3.1 Type of Research

This research is an in-depth exploration of the utilization of high-level technology, especially by utilizing the Convolutional Neural Network (CNN) method, in the context of developing a sign language detection system. In terms of methodology, this research can be identified as an experimental study with a quantitative approach. This approach allows researchers to collect empirical data that can be measured numerically, following the research objectives, to analyze the performance of CNN in detecting sign language. Experimental research tends to focus on empirical data collection and hypothesis testing. The application of variable control and manipulation of certain variables in this research is intended to understand the impact on experimental results systematically. With this approach, the research leads to technology development and a deeper scientific understanding of the quantitative aspects of applying high-level technology to sign language detection systems.

3.2 Research Design

The research design that can be used is a research strategy with a One-Group Pretest-Posttest approach. In the pretest stage, baseline data is collected to provide an initial picture of sign language comprehension before any intervention (change). After that, the sign language detection model using CNN will be implemented. After the model implementation phase, the research variables are re-measured at the post-test stage. The data collected from the posttest will allow researchers to evaluate the impact of the effectiveness of the sign language detection model in improving sign language comprehension.

3.3 Research Variables

3.3.1 Independent Variables

In this research, the main focus is on the utilization of a convolutional neural network (CNN) approach as a major element in the development of artificial intelligence systems. Specifically, CNN has been adopted as a deep learning technique to analyze and process sign language-related data. This approach creates an efficient and accurate structure for managing the sign information.

3.3.2 Bound Variables

The center of attention in this research lies in evaluating the system's ability to identify and interpret sign language. This includes the accuracy in recognizing gesture patterns and expressions, the ability of the system to give meaning to the cues, and the system's responsiveness in responding appropriately. Therefore, this dependent variable reflects how CNN integration can improve the system's performance in effectively understanding and responding to sign language.

3.4 Research Procedures

The research method applied to this report involves several systematically designed steps. First, the research began with a problem identification and literature review to understand the theoretical framework and related literature. This step was done to build a strong foundation for the formulation of research questions and research objectives. After the literature review stage, the research continued with needs analysis and data collection. Researchers identified system requirements at this stage, developed a framework, and collected relevant data. This needs analysis forms the basis for developing the model and software that will be further evaluated.

The design stage involves drawing up a detailed plan for the system implementation. In the context of image classification, the research used the Convolutional Neural Network (CNN) approach because of its ability to detect unique features without the need for pre-processing steps and recognize features automatically without supervision. The system implementation involves data training and model building using CNN. This

process is essential to ensure the model can provide accurate and reliable results. The success of this stage will affect the overall quality of the research.

Finally, the research reached the system testing stage to evaluate the performance of the developed model. This process involves trials, evaluation, and validation of the results. In software development, the research adopted the waterfall method approach to ensure that each stage was carried out sequentially and well-documented. The research flowchart, as seen in Figure 8, visually depicts the series of research steps.

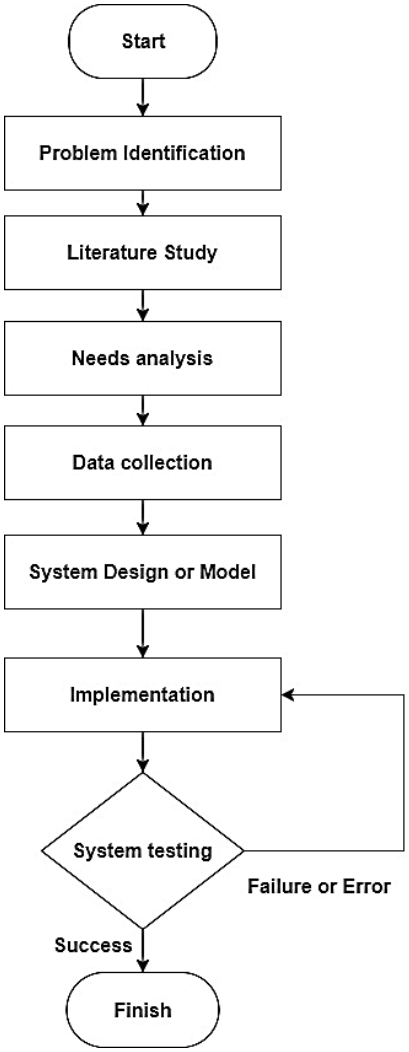

Figure 8. Research Methods

3.5 Problem Identification

Problem identification involves an in-depth understanding of the constraints faced in sign language detection using conventional methods. It involves careful consideration of the rationale for using Deep Learning, especially CNNs, and its contribution to solving the problem. Aspects such as detection accuracy, processing speed, and adaptation to sign language variations become focal points in the problem identification stage. It is essential to form a solid foundation for the research by explaining in detail the relevance of the identified problem and the potential positive impact that can be achieved. Along with that, it is necessary to emphasize the boundaries of the research to ensure proper focus and provide the necessary context to formulate the research questions properly.

3.6 Literature Study

At this stage, the researcher conducted a literature review to detail the understanding of the basic concepts of Deep Learning, especially in the context of the Convolutional Neural Network (CNN) method, which is the main focus of this research. First, researchers underwent a literature exploration to gain a deep understanding of Deep Learning as a subset of Machine Learning that focuses on using deep neural networks to recognize patterns and make decisions. A thorough understanding of this concept was essential so that the researcher could detail the application of Deep Learning in the context of sign language detection. Furthermore, the researchers focused their literature review on the Convolutional Neural Network (CNN) method, a neural network designed to process spatial data, such as images. This entire process formed a strong foundation for moving on to the following research stage: designing.

3.7 Needs Analysis

The needs analysis in this research includes two main aspects: data needs analysis, and device needs analysis.

3.7.1 Data Requirement Analysis

a) Input Data

The input data in this research comes from the image taken by the webcam device. The user opens the application, and the image from the webcam will be captured in real time. The system then tracks the user's hand position, processes it, and translates the sign language.

b) Process Overview

The process starts with the user opening the application and activating the webcam, and the image from the webcam will be captured in real time. There will be a box in the system that will be used to detect the user's hand, and if it is detected that the user is using sign language, the application will directly translate the sign language in real time.

c) Overview of Data Pre-processing

Data pre-processing is the data preparation stage before the classification process is carried out. The stages involve:

1) Input data from the webcam, where the system detects the hand in real time.
2) The hand pattern detection process is based on SIBI sign language, where the system tracks the patterns of the fingers for classification.
3) Classification using the Convolutional Neural Network (CNN) method. The system will classify the hand pattern and provide results according to the SIBI sign language form pattern.

3.7.2 Device Requirement Analysis

a) Hardware Requirements: (1) Laptop with 12th Gen Intel(R) Core (TM) i5-12500 H Processor, (2) Intel(R) Iris(R) Xe Graphics GPU, (3) NVIDIA GeForce GTX 1650 GPU, (4) 16GB RAM, (4) 1TB SSD, and (5) 720p Webcam camera.

b) Software Requirements: (1) Windows 11 64 Bit, (2) Visual Studio Code, (3) Python 3.11, (4) OpenCV, (5) Keras, and (6) TKinter

These hardware and software specifications are designed to support developing and implementing Convolutional Neural Network models in sign language detection systems.

3.8 Data Collection

The data acquisition process in this research framework begins with collecting datasets that match the research objectives. Dataset selection is essential because the dataset's quality greatly affects the model's performance in recognizing sign language cues with varying characteristics. In this context, the dataset used consists of sign language images captured using a webcam. These images include the SIBI (Indonesian Sign Language System) alphabet.

The acquired dataset consists of 2,600 digital images of sign language demonstrations of the alphabets A to Z in ".jpg" file format. There are 26 dataset classes representing the A-Z alphabet with 100 images, according to the SIBI gestures registered on the SIBI Dictionary website managed by the Ministry of Education and Culture of the Republic of Indonesia. The reference to the SIBI Dictionary website as an index source shows an accurate and thorough approach to determining the classes of sign language gestures that are the focus of the research. Overall, this careful and representative dataset selection provides a solid foundation, ensuring that the implemented Convolutional Neural Network (CNN) model can recognize sign language signs with varied characteristics, including gesture variations covering the A-Z alphabet in SIBI sign language.

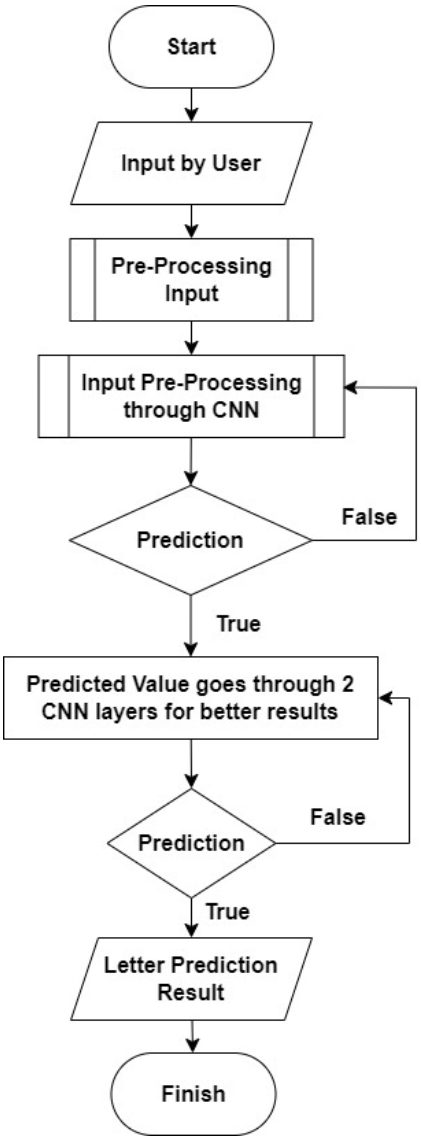

Figure 9. SIBI Alphabet Classification Flowchart

3.9 System Design

In the system design stage of this research, researchers developed a Convolutional Neural Network (CNN) architecture that was customized specifically to meet the needs of sign language detection. This process involves determining the optimal model parameters, including the number of convolution layers, kernel size, and activation function, to improve detection accuracy. Next, an SIBI alphabet classification flowchart is applied to understand the implementation of sign language detection. This diagram explains in detail the process of recognition and classification of SIBI alphabets using data obtained from webcam captures when users express SIBI sign language. This series of processes is illustrated in Figure 9. Data processing involves interaction between the user and the system, which aims to strengthen the model's understanding of specific sign language.

3.10 Implementation

After the model design, the next step involves implementing the model. Dataset preparation is a critical stage in this implementation. Once the dataset is prepared, the next step is to engage the CNN model in the training process. The training process involves iteratively adjusting the model parameters to achieve optimal sign language detection results. The training stage is key to ensuring that the model can deliver accurate and reliable results in real-world scenarios.

3.11 System Testing

In addition to parameter adjustments, the training results are assessed to measure the model's performance. First, the detection results are evaluated using essential metrics such as accuracy, precision, and recall. These metrics provide an overall view of the model's ability to identify sign language with optimal accuracy and efficiency. The model validation process is essential to ensure that the model is generalizable and reliable in various situations. Validation is done through testing the model on diverse datasets, guaranteeing the model's ability to recognize sign language consistently and accurately in various contexts. This validation also helps in identifying potential overfitting (where the model tries to learn all the details, including noise in the data, and tries to fit all data points into the line) or underfitting (where the Machine Learning model cannot learn the relationship between variables in the data and predict or classify new data points) that may affect the performance of the model.

3.12 Data Sources

The data collected may include variations of hand and finger movements and other visual elements of sign language, such as the position and direction of hand movements, thus allowing for a comprehensive analysis without sacrificing their uniqueness.

3.13 Data Collection Technique

A practical data collection approach in sign language research involves visual recording technology. Using webcams or optical sensors allows the detailed recording of hand and finger movements in sign language, resulting in a dataset rich in visual information for model development. The next step involves a careful data labeling process, essential for training the Convolutional Neural Network (CNN) model to recognize and interpret sign language more accurately. The data labeling process is done carefully to ensure the accuracy and relevance of the data used in training the model. This approach can improve the reliability and validity of sign language recognition models.

3.14 Research Instruments

This research uses a combination of webcams or visual sensors and Deep Learning software to analyze sign language gestures and expressions. The instrument is designed to gain a deep understanding of sign language interaction by utilizing advanced technology.

3.14.1 Webcam or Visual Sensor

This research instrument utilizes a webcam or visual sensor to record sign language gestures and expressions accurately. This technology allows for more detailed and real-time data capture, creating a solid foundation for in-depth analysis.

3.14.2 Deep Learning Software

Deep Learning software implements and trains the Convolutional Neural Network (CNN) model. This model is designed to recognize hand and finger movement patterns associated with sign language. Deepening it through training can improve interpretation accuracy, assist in complex gesture recognition, and provide a basis for data-driven analysis.

3.14.3 Human Validation

Human validation is used to ensure the accuracy of the interpretation; the results of the automated system will be validated through the participation of humans who have expertise in sign language. This aims to confirm the accuracy and sustainability of the analysis results obtained from the Deep Learning model.

3.15 Data Analysis

The data analysis of this study combined two essential aspects, namely the application of statistical methods to parse the data from the pre-and post-implementation measurements, as well as the assessment of the model's performance. In the statistical analysis, various techniques were applied to investigate patterns in the data before and after the measures were taken. Using appropriate statistical methods helps identify significant changes in the observed variables, enabling solid conclusions based on the observations. Meanwhile, the performance evaluation of the model, specifically the Convolutional Neural Network (CNN), provides deep insight into its ability to recognize sign language. Through model performance evaluation, this research provides qualitative findings on the effectiveness of the CNN model and presents an empirical basis for further recommendations or improvements to the implemented approach.

4. Result and Analysis

4.1 Testing and analysis of the SIBI sign language detection system

The results of testing the SIBI sign language detection system can be seen in Figures 10, 11, 12, and 13. In Figure 10, the user interface of the SIBI sign language detection system is shown. A dark-colored box displays the user's image from the camera, while a light-colored box displays the user's hand gestures in black and white. On the right side is an illustration of the SIBI alphabet sign language gesture as a guide. At the bottom, there are three main elements: "Letter" shows the alphabetic character being demonstrated, "Word" depicts the word formed from the letters, and "Sentence" shows the sentence formed from the words created by the user.

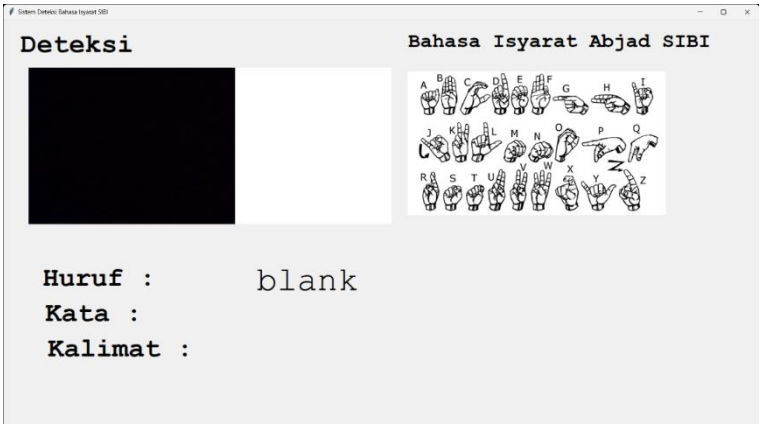

Figure 10. SIBI Alphabetic Sign Language Detection System Interface Display

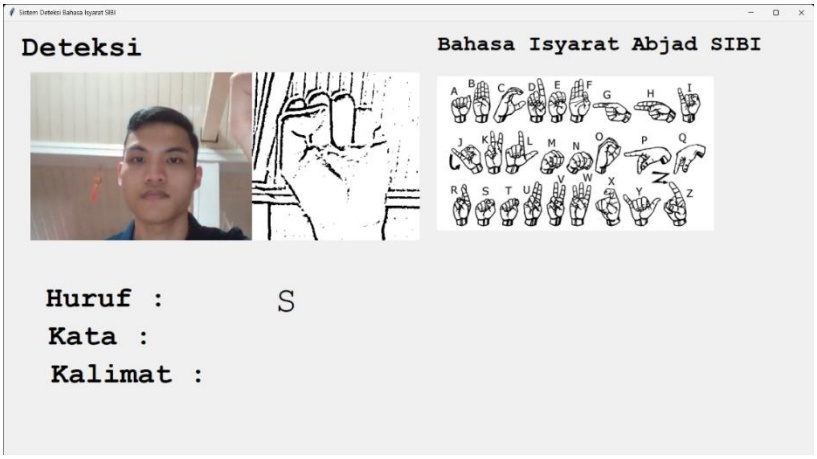

Figure 11. SIBI Alphabetic Sign Language Detection System Letter S

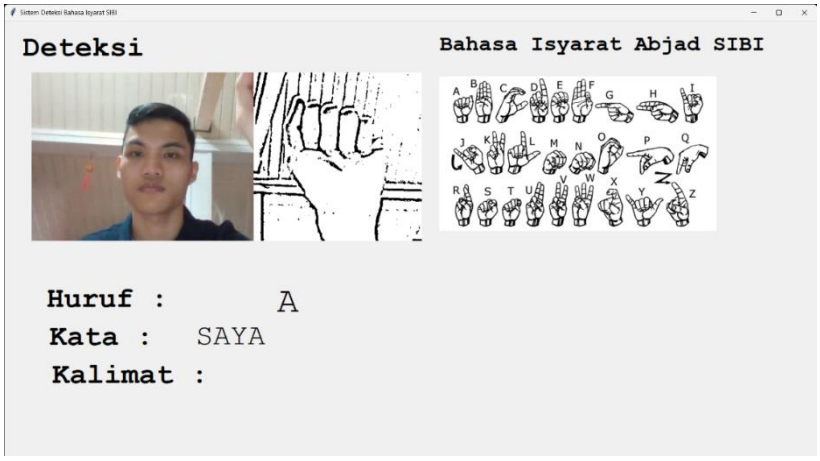

Figure 12. SIBI Alphabetic Sign Language Detection System Letter A

In Figure 12, the system recognizes the alphabetic sign language A according to the user's hand movements. The process involves the user performing SIBI sign language gestures, which are then recorded in black and white. The system can then predict the alphabet, which is demonstrated by using a previously prepared dataset. The result of this character prediction will be displayed in front of "Letter" in real time.

Moreover, Figure 13 shows the system composing the word "ME". In the initial 50 frames, the predicted text is stored in the backend, and the character with the highest prediction is displayed in front of "Word." The prediction results for each frame are retrieved and stored, forming the text according to the sign language shown. When the system detects a change to a blank screen, indicating that the word prediction is complete, the program considers the word complete and moves on to the next character.

Figure 13. System Displays Sentences from Word Arrays

Once all characters are predicted, the complete word is displayed in front of the characters, and the system looks for a blank screen to determine that the word is whole. If the screen remains blank, the program considers that the word is complete and moved to the front of the sentence. In this way, the program composes text from sign language on the front, uses a model to predict text, and has a mechanism to detect when a word or sentence is complete to display the result appropriately on the screen.

4.2 Dataset creation process

This research uses a new dataset that uses the OpenCV library, consisting of images taken using a webcam. This dataset includes images of SIBI sign language demonstrations ranging from letters A to Z in ".jpg" file format. A total of 2,600 images were collected, with 26 classes representing the letters of the alphabet. Each class consists of 100 images. The dataset creation process can be seen in Figure 14.

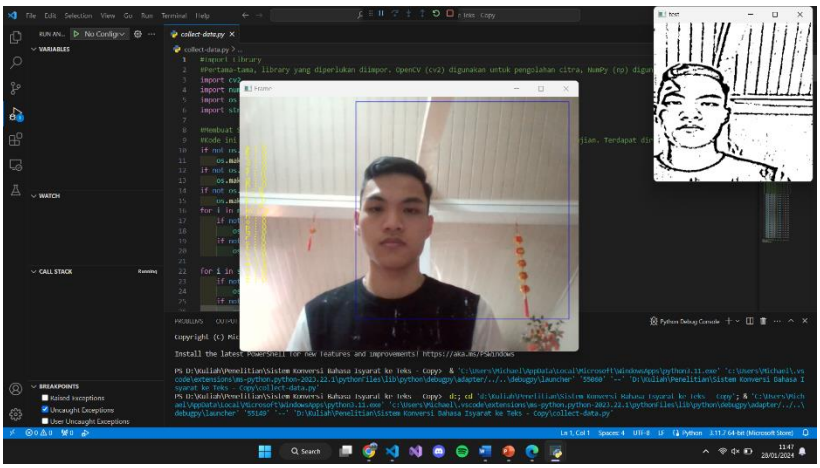

Figure 14. The dataset creation process

The process starts by capturing each frame displayed by the webcam. Each frame defines the region of interest (ROI) with a blue bounding box. From the full image, the ROI is extracted in the form of RGB values and converted to a grayscale image. Finally, a Gaussian blur filter is applied to the image to help extract various image features. The following is an example of an A-alphabet SIBI sign language dataset that has been pre-processed, as shown in Figure 15.

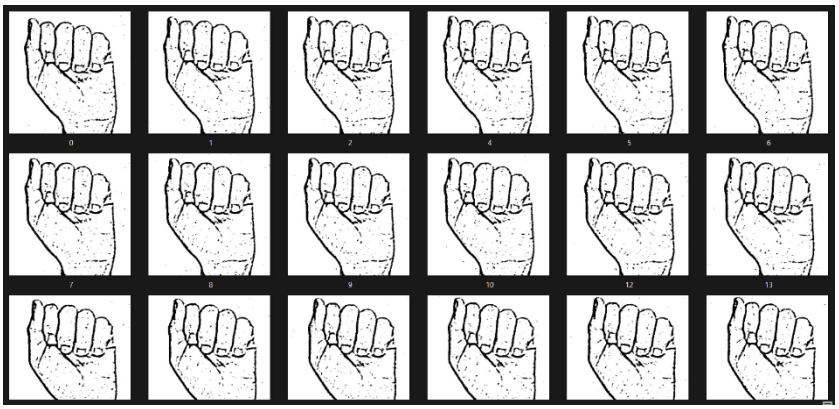

Figure 15. Pre-Processed Dataset

The next step involves training the data to form the model to be tested. Before starting the training, the dataset was split into two parts: training data and testing data. Of the 2,600 data, 70% was used as training data, while 30% served as testing data. The training data involved batch size 10 epoch 5, with steps per epoch of 12,841. Batch size is the number of data samples that pass simultaneously through the neural network. An epoch is a hyperparameter that determines how often the Deep Learning algorithm works through the entire forward and backward dataset.

Each epoch produces learning variables that are reflected in accuracy and loss values. The loss value acts as an evaluation parameter to assess the extent to which the system learning results can be considered good or bad; the smaller the loss value, the more consistent the model learning. The training results show an accuracy rate of about 99%, with a loss value of about 3.7%, as shown in Figures 16 and 17.

```
Epoch 1/5
1285/1285 [==============================] - 190s 148ms/step - loss: 0.0691 - accuracy: 0.9811 - val_loss: 0.0046 - val_accuracy: 0.9986
Epoch 2/5
1285/1285 [==============================] - 167s 130ms/step - loss: 0.0539 - accuracy: 0.9853 - val_loss: 0.0069 - val_accuracy: 0.9977
Epoch 3/5
1285/1285 [==============================] - 164s 128ms/step - loss: 0.0433 - accuracy: 0.9889 - val_loss: 0.0207 - val_accuracy: 0.9946
Epoch 4/5
1285/1285 [==============================] - 164s 127ms/step - loss: 0.0503 - accuracy: 0.9863 - val_loss: 0.0059 - val_accuracy: 0.9988
Epoch 5/5
1285/1285 [==============================] - 168s 131ms/step - loss: 0.0370 - accuracy: 0.9900 - val_loss: 0.0018 - val_accuracy: 0.9993
```

Figure 16. Accuracy and Loss Value

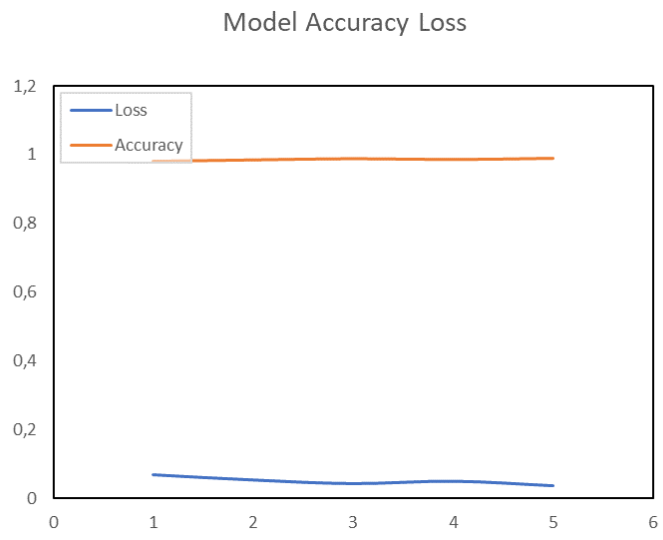

Figure 17. Accuracy and Loss Value Curves

4.3 Gesture Classification

1) Convolutional Layer 1: The Input is an image with a resolution of 128x128 pixels. The image is processed in the first convolutional layer in an initial step using 32 filter weights of 3x3 pixels each. The result is an image of 126x126 pixels, one for each filter weight.
2) Pooling Layer 1: The images are down-sampled using the 2x2 max pooling technique, where only the highest value in each 2x2 box of the matrix is retained. As a result, the image is down-sampled to 63x63 pixels.
3) Convolutional Layer 2: The 63x63 pixel image from the output of the first pooling layer is taken as Input to the second convolutional layer. Here, the image is processed using 32 filter weights of 3x3 pixels, resulting in an image of 60x60 pixels.
4) Pooling Layer 2: The resulting images are downsampled again using the 2x2 Max-Pooling technique, and the resolution is reduced to 30x30 pixels.
5) Fully Connected Layer 1: These images become the Input for the fully connected layer with 128 neurons. The output of the second convolutional layer is reshaped into an array of size 30x30x32 = 28800 values. This layer uses Dropout with a value of 0.5 to avoid overfitting.
6) Fully Connected Layer 2: The output of the first fully connected layer becomes the Input for the fully connected layer containing 96 neurons.
7) Final Layer: The output of the second fully connected layer is used as Input for the final layer, which has the number of neurons corresponding to the number of classified classes.

4.4 Training and Testing

The input image (RGB) is converted to grayscale format, and the Gaussian blur technique is applied to remove unnecessary noise. Next, Adaptive Threshold extracts the hand area from the background and adjusts the image size to 128 x 128. After going through the pre-processing process, the input image is provided to the model for training and testing after going through the mentioned operations. The prediction layer evaluates how likely the image belongs to one of the classes. The output is normalized between 0 and 1, ensuring that the total value in each class is equal to 1, using the softmax function.

Moreover, to improve the output of the prediction layer, the network was trained using the labeled data. Cross-entropy is used as a performance indicator in classification. This continuous function takes a positive value at points different from the labeled data and has a zero value when it corresponds to the labeled data. Therefore, cross-entropy is optimized by minimizing it close to zero. In the network layer, the weights of the neural network are adjusted. After finding the cross-entropy function, its value is optimized using the gradient descent method, especially with Adam Optimizer, to obtain optimal results.

In this system testing phase, the accuracy of the CNN implementation will be evaluated. The test is conducted by determining the distance between the hand and the webcam to detect and predict finger gestures. Researchers conducted trials by pointing the webcam at the test subject, namely the researcher himself, and then making sign language gestures that match the class in each image data. The accuracy of the results during the data testing process can be seen in Table II.

Table II. Test Data Prediction Results

No	Room Condition	Number of Tests	Accuracy Level
1	Low Light (Dim)	5	94,37 %
2	Bright	5	96,29 %

Testing was conducted five times on 26 SIBI alphabet sign language classes in the same room with varying lighting conditions. The data in Table II shows that in a room with minimal or dim light conditions, the accuracy reaches 94.37%. This is influenced by the number of letters that cannot be predicted correctly due to the lack of incoming light, resulting in the webcam not being able to capture the movement properly. Meanwhile, in a room with sufficient or bright light conditions, the accuracy rate increased to 96.29%. This result indicates that improved lighting improves the quality of the captured image and the prediction process's performance.

5. Conclusions and Suggestions

Based on the results of the implementation and testing of the SIBI alphabet sign language detection system with the Convolutional Neural Network (CNN) method, it can be concluded that the real-time SIBI alphabet sign language detection system using the Convolutional Neural Network method has been successfully implemented. The main focus of this development is the recognition of the SIBI alphabet. The system can recognize alphabet classes based on user demonstration. In the dataset training process using 2,600 images with epoch 5 and batch size 10, a training accuracy of 99% was obtained, indicating the optimality of SIBI alphabet sign language detection. Improved prediction occurs after the application of two layers of algorithms, where verification and prediction are performed on symbols that are similar to each other. The system can detect almost all symbols, if demonstrated correctly, with minimal or no background noise and adequate lighting.

The system that has been built still has some shortcomings. Therefore, the author provides suggestions for improving this research in the future. The suggestions and Input given by the author for further research development are as follows:

1) Increase the number and variety of training datasets so that the model can perform detection more accurately and quickly. Some aspects of datasets that need to be considered involve variations in subjects as dataset models, variations in backgrounds and objects at the time of data collection, and variations in camera types with different resolutions.
2) The sign language detection system using the CNN method can be further developed so that it can not only be used to detect alphabets but also detect a wider sign language vocabulary.
3) Developing the system as a two-way communication tool between deaf and speech-impaired people and normal people.
4) Develop existing features and add new features to encourage users to learn sign language.

Acknowledgments: We would like to thank all my research supervisors, examiners, and lecturers at the Department of Informatic Engineering, Narotama University Surabaya. Hopefully, this research will be helpful to many people.

Author contributions: All authors are responsible for building Conceptualization, Methodology, analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision of project administration, funding acquisition, and have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. A. B, Yunanda,; F, Mandita,; A. P, Armin. Pengenalan bahasa isyarat indonesia (bisindo) untuk karakter huruf dengan menggunakan microsoft kinect. *Fountain Of Informatics Journal*, **2018**, 3, 2, 41–45. [Crossref]
2. D. Yolanda,; K. Gunadi,; E. Setyati. Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network, *Jurnal Infra*, **2020**, 8, 1, 203–208. [Crossref]
3. N. Nuryazid and A. Mulwinda. Pengembangan Aplikasi Kamus Bahasa Isyarat Indonesia (Bisindo) dengan Mengintegrasikan Cloud Video Berbasis Android," *Edu Komputika Journal*, **2017**, 4, 1, 34. [Crossref]
4. A. S, Nugraheni,; A. P. Husain,; and H. Unayah. Optimalisasi penggunaan bahasa isyarat dengan sibi dan bisindo pada mahasiswa difabel tunarungu di prodi pgmi uin sunan kalijaga. *Holistika: Jurnal Ilmiah PGSD* **2023**, 5, 1, 28–33. [Crossref]
5. T. Y, Pajar,; D. Purwanto,; H. Kusuma. Pengenalan Bahasa Isyarat Tangan Menggunakan Depth Image. *Jurnal Teknik ITS 2018*, 7, 1, A104–A109. [Crossref]
6. T. A, Dompeipen,; S. R. U. A. Sompie,; and M. E. I, Najoan. Computer Vision Implementation for Detection and Counting the Number of Humans," *Jurnal Teknik Informatika,* **2021**, 16, 1, 65–76, 2021. [Crossref]
7. S. Subburaj and S. Murugavalli. Survey on sign language recognition in context of vision-based and deep Learning. *Measurement: Sensors,* **2022**, 23, 100-385. [Crossref]
8. E. R. Kasim,; A. Fransiska,; M. Lusli, and S. Okta. Analisis situasi penyandang disabilitas di Indonesia: Sebuah desk-review. *Pusat Kajian Disabilitas, Fakultas Ilmu-Ilmu Sosial dan Politik Universitas Indonesia, 2010*.
9. G. Ciaburro and B. Venkateswaran. Neural Networks with R: Smart models using CNN, RNN, deep Learning, and artificial intelligence principles. *Packt Publishing Ltd.*
10. V. Bheda dan D. Radpour, "Using deep convolutional networks for gesture recognition in american sign language," arXiv preprint arXiv:1710.06836, 2017. [Crossref]
11. W. S. E. Putra, "Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101," Jurnal Teknik ITS, vol. 5, no. 1, 2016. [Crossref]
12. S. Sharma dan S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," Expert Syst Appl, vol. 182, hlm. 115657, 2021. [Crossref]
13. N. Thiracitta dan H. Gunawan, "SIBI Sign Language Recognition Using Convolutional Neural Network Combined with Transfer Learning and non-trainable Parameters," *Procedia Comput Sci* **2021**, 179, page: 72–80. [Crossref]
14. R. B. F. Hakim, Implementasi convolutional neural network terhadap transportasi tradisional menggunakan keras," **2018**. *Final Project of Statistics Study Program, Faculty of Mathematics and Natural Sciences UII Yogyakarta*.
15. C. K. Dewa, A. L. Fadhilah, dan A. Afiahayati, "Convolutional neural networks for handwritten Javanese character recognition," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems),* **2018**, 12, 1, 83–94. [Crossref]
16. M. B. S, Bakti and Y. M. Pranoto. Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network," dalam Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi), 2019, hlm. 11–16.
17. H. M. Putri, F. Fadlisyah, dan W. Fuadi, "Pendeteksian Bahasa Isyarat Indonesia Secara Real-Time Menggunakan Long Short-Term Memory (LSTM)," *Jurnal Teknologi Terapan and Sains* **2022**, *4.0,* 3, 1, 663–675. [Crossref]
18. D. Darmatasia. Pengenalan Sistem Isyarat Bahasa Indonesia (Sibi) Menggunakan Gradient-Convolutional Neural Network," Jurnal INSTEK (*Informatika Sains dan Teknologi*), **2021**, 6, 1, 56–65. [Crossref]
19. S. Satwikayana,; S. A. Wibowo,; and N. Vendyansyah. Sistem Presensi Mahasiswa Otomatis Pada Zoom Meeting Menggunakan Face Recognition Dengan Metode Convulitional Neural Network Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)* **2021**, 5, 2, 785–793. [Crossref]
20. A. R. Syulistyo,; D. S. Hormansyah,; P. Y. Saputra. SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN)," dalam IOP Conference Series: Materials Science and Engineering, *IOP Publishing*, **2020**, 012082. [Crossref]
21. S. Gong dkk.Visual Object Recognition. *Advanced Image and Video Processing Using MATLAB*, **2019**, 351–387.
22. U. Michelucci. Advanced applied deep Learning: convolutional neural networks and object detection. *Springer,* **2019**.