# Convolutional Neural Network Implementation in BISINDO Alphabet Sign Language Recognition System Using Flask

Aning Kinanti[1], Donny Maulana[2], Edora[3]
[1,2,3] Department of Informatics Engineering, Pelita Bangsa University, Cikarang - Indonesia
[1] kinantianing@mhs.pelitabangsa.ac.id, [2] donny.maulana@pelitabangsa.ac.id, [3] edora@pelitabangsa.ac.id

Accepted 30 May 2024
Approved 18 June 2024

***Abstract*—This research develops a system for recognizing finger spelling gestures in Indonesian Sign Language (BISINDO) using Convolutional Neural Network (CNN). The objective of this research is to apply the Convolutional Neural Network (CNN) method to the BISINDO finger spelling gesture recognition system to improve its accuracy. The method employed is Convolutional Neural Network (CNN), an effective method for processing image data for pattern recognition. Based on the test results, the system demonstrates that the developed CNN model is capable of recognizing BISINDO finger spelling gestures with an accuracy of 97.5%. This indicates that the BISINDO finger spelling gesture recognition system performs well in pattern recognition. The implementation of the system for real-time prediction via a web interface using Flask also enhances its accessibility. However, there is still room for improvement, particularly in recognizing one of the 26 letters that has not been predicted accurately. For further development, it is recommended to consider collecting a larger dataset and incorporating more complex gesture variations to improve recognition accuracy.**

***Keywords*— Accuracy; Dataset; Flask; Convolutional Neural Network; Gesture Recognition; Indonesian Sign Language***

## I. INTRODUCTION

Sign language is a method of communication used by individuals who were deaf and mute [1]. It involves bodily or physical movements to convey messages [2]. For these individuals, sign language has become the primary means of communication in their daily lives [3]. According to data from the Disability Management Information System of the Ministry of Health of the Republic of Indonesia in March 2022, there were 212,240 individuals with disabilities in Indonesia. Of these, approximately 9.14% or 19,392 individuals were deaf-mute [4]. In this particular setting, sign language holds dual significance: it serves not only as a means of communication for those with hearing impairments but also as a vital component of their cultural heritage and identity.

Indonesian Sign Language (BISINDO) is one of the two sign languages used in Indonesia, along with the Indonesian Sign System (SIBI), which was officially recognized at the sixth National Congress of Gerkatin in Bali in 2002 to preserve the authenticity of natural sign language [4]. Despite the existence of both sign languages, the disabled community in Indonesia more frequently uses BISINDO. This preference is due to the fact that BISINDO is derived from the Indonesian language, which is their native language and is used daily [3].

One of the critical elements in sign language used by the deaf-mute community is finger spelling or manual alphabet, which serves to spell out words in spoken language by showing each letter individually using hand fingers [5]. Consequently, finger spelling forms the foundation of communication in sign language. This also makes finger spelling a key element in building an effective communication system for the deaf-mute. However, as known, not all individuals can understand sign language [1]. Based on research conducted by Suryati in 2019, a survey was carried out involving deaf individuals, with responses from two deaf participants, assisted by their closest companions, and ten members of the general public who had interacted with the deaf in Bogor Regency. The results concluded that both groups of respondents agreed there were difficulties in communication between the deaf and the general public [6]. This highlights the importance of developing a hand gesture recognition system to aid in understanding and interpreting sign language.

Several systems have been developed to assist in translating sign language into alphabetical text. One such example is the research conducted by Dimas Permana and Joko Sutopo in 2023, where they developed an alphabet recognition application for the Indonesian Sign System (SIBI) using the YOLOv5 algorithm, achieving an accuracy of 77% [7]. Although an accuracy of 77% is considered adequate, there were still some shortcomings that need to be addressed. Firstly, an accuracy rate of 77% is not yet optimal for daily use, particularly in communication contexts that require a high level of accuracy. Secondly, there were still some letters that the system fails to recognize accurately, indicating that there is room for improvement in sign recognition. These findings suggest that while advancements in sign language recognition technology have been made, further research is needed to enhance letter recognition accuracy. To improve accuracy in image recognition systems, it is important to consider the use of Convolutional Neural Networks (CNN), a method effective in processing image data for pattern recognition [8].

Convolutional Neural Network (CNN) or ConvNet is a notable algorithm in deep learning [9]. Deep Learning itself is a branch of machine learning, derived from Artificial Neural Networks (ANN), or it can be considered an evolution of ANN [10]. Deep Learning employs deeper network architectures (with many layers) compared to conventional ANN, enabling more complex modeling of larger and more intricate data. CNNs were specifically used for classification tasks [11]. Since 2012, CNNs have become a crucial model in object recognition [12]. CNNs have become one of the most dominant methods in image data processing and pattern recognition, particularly due to their ability to handle complex problems such as object recognition in images and face recognition. CNNs have proven to be effective and were widely used with satisfactory results [13].

The excellent performance of CNN can be observed from related research conducted by Fauzan Akbar, Aris Triwiyanto, and Achmad Hidayanto, titled "Design of Hand Gesture Recognition Program using Convolutional Neural Network (CNN)," which achieved an accuracy of 92% [10]. This result confirms that CNN is an effective method for hand gesture recognition, with a high level of accuracy. Both studies demonstrate that CNN has great potential in hand gesture recognition, particularly in the context of Indonesian Sign Language. This success provides hope that the development of hand gesture recognition systems using CNN can be continuously improved to make a more significant contribution to the field of technology.

Flask, as a minimalist and easy-to-understand web framework, has advantages in developing web applications that require integration with webcams, including in the context of hand gesture recognition. Flask's advantage lies in its flexibility to facilitate developers in integrating webcam features into the web application they were developing.

From the outlined issues, there is a need to develop a hand gesture recognition system for the Indonesian Sign Language (BISINDO) alphabet that can recognize hand gestures with a high level of accuracy in real-time. Therefore, in this research, the Convolutional Neural Network (CNN) method is chosen for its excellence in image and object recognition. The development of this system will be integrated with the Flask framework to allow access through the web. Flask will be used to create a user interface that allows users to perform real-time hand gesture recognition using a webcam. After the hand gesture is identified via the webcam, Flask will send it to the CNN model for processing. The model will identify the hand gesture in real-time images and return the result to Flask. Flask will then display the identification result to the user through the user interface.

By using Flask, developers can build a BISINDO hand gesture recognition system that is easily accessible and usable via the web with real-time recognition, without the need to re-run the model, which would require more time when the system is to be used. This also allows more people to access and utilize it, as well as integration with various other platforms and applications for further development.

## II. LITERATURE REVIEW

### *1. Sign Language*

Sign language is a communication system that utilizes hand movements, facial expressions, and body postures to create symbols representing letters or words [7]. It is employed by individuals with hearing and speech impairments to communicate [14]. Communication in sign language is a form of non-verbal communication that does not involve any auditory components [15]. Instead, it involves the use of hand gestures, body movements, and eye contact [16]. Sign language is considered unique as it varies from country to country [17]. In Indonesia, two main types of sign language were commonly used: the Indonesian Sign System (SIBI) and Indonesian Sign Language (BISINDO) [18]. Based on the aforementioned theory, it can be concluded that sign language is a non-verbal communication system that relies on hand movements, facial expressions, and body postures to form symbols representing letters or words. Sign language is primarily used by individuals with hearing or speech impairments to communicate. In Indonesia, sign language is divided into two types: the Indonesian Sign System (SIBI) and Indonesian Sign Language (BISINDO). According to the European Union of the Deaf, as explained on their website, there is no universal use of sign language in the world [19]. Each country has its own unique and distinct form of sign language. Communication in sign language does not involve sound elements but rather uses hand gestures, body movements, and eye contact symbols.

### *2. Indonesian Sign Language (BISINDO)*

Indonesian Sign Language (BISINDO) is the sign language used by deaf individuals based on their understanding of their surroundings [17]. BISINDO is often used by the deaf community from childhood and can be considered their mother tongue, as it is easily understood and used in daily life [20]. BISINDO is one of the two sign languages used in Indonesia, alongside the Indonesian Sign System (SIBI) [17]. BISINDO is a type of sign language used by deaf individuals and is considered older than SIBI [4]. The alphabet in BISINDO consists of 26 characters, from A to Z. These letters can be formed with one hand, such as C, E, I, J, L, O, R, U, V, and Z, as well as letters that can be formed with two hands, such as A, B, D, F, G, H, K, M, N, P, Q, S, T, W, X, and Y [4]. This can be seen in Figure 8.

### *3. Deep Learning*

Deep learning is a learning method that employs artificial neural networks with multiple layers [11]. It uses metadata as input [9] and consists of hidden layers that train a set of metadata based on the output from previous neurons [9]. With deep learning, computers can directly learn to classify images or sounds [21]. The deep learning approach can automatically extract features from input images without manual intervention [2]. This machine learning technique mimics the way human neurons work, which were fundamental components of the brain [8]. Deep learning has achieved remarkable results, largely due to more powerful computation, larger datasets, and better neural network training techniques [22]. Deep learning is a subset of machine learning that uses Artificial Neural Networks (ANN) as its foundation or can be considered an evolution of ANN [10]. By utilizing multi-layered neural networks, deep learning enables computers to learn complex patterns from raw data, such as images or sounds, without the need for manual feature extraction.

### *4. Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) or ConvNet is a method commonly used for image analysis [23]. CNN utilizes convolutional features to extract features from images [24]. CNN is a form of Multilayer Perceptron (MLP) that works on two-dimensional data and resembles the human neural network [25]. The main advantage of CNN is its ability to effectively extract important features from images and classify them with high accuracy [8]. This method can independently learn to recognize objects, extract object features, classify, and can be used on images [23]. Broadly speaking, CNN has two layers in the pattern recognition model, namely the feature extraction layer and the classification layer [8]. CNN also has two methods, namely classification using feedforward and learning stages using backpropagation [26]. CNN can be used to process input images with complex backgrounds and various hand gesture postures [2]. CNN has a significant impact on image recognition because it mimics the human visual cortex's image recognition system, thus able to process image information effectively [22]. Although CNN achieves the best results on large datasets, it requires a lot of data and computational resources to train [27]. The architecture of Convolutional Neural Network (CNN) can be seen in Figure 7.

1) Convolutional Layer

The Convolutional Layer is the first layer in the architecture of a Convolutional Neural Network (CNN) [1]. It is responsible for calculating the output of neurons connected to a local region of the input image. Each neuron in this layer uses a filter that is repeatedly shifted to multiply a small region connected to the input image, perform convolution operations, and produce the output of the layer. The filters in the convolutional layer were used to capture specific patterns from the image, such as edges, corners, and textures [11]. This process helps in extracting important features from the image [1]. Initially, the input image is enlarged by adding zero pixels around it. Each shift of the filter produces a 2-dimensional matrix as the output of the convolution process [9].

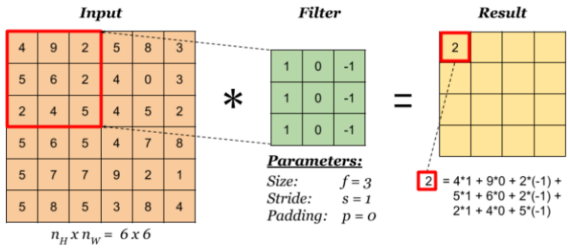


**Fig. 1. Illustration of the Convolution Calculation Matrix [9]**

2) Pooling Layer

Pooling or subsampling is the process of reducing the size of a matrix. The pooling layer in Convolutional Neural Network (CNN) aims to reduce the dimensions of an image to make it easier to process by the next convolutional layer. There were two commonly used pooling techniques, which were average pooling and max pooling [1]. Max Pooling is a technique to represent an area by using the largest value within that area. On the other hand, Average Pooling is a technique that uses the average value within an area to represent that area [18].

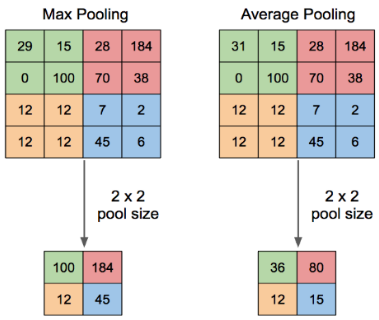


**Fig. 2. Illustration of the Max Pooling and Average Pooling Processes [18]**

3) Flatten Layer

Flatten is the stage in which the matrix produced from feature learning (through convolution and pooling) is transformed into a vector. This vector is then used as input for classification with a fully connected layer structure [13]. It converts the 2-dimensional matrix resulting from feature learning into a 1-dimensional vector [9].

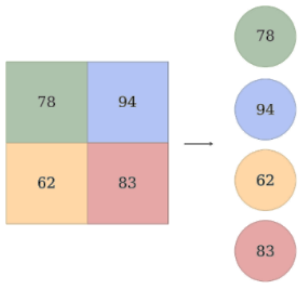


**Fig. 3. Ilustation of the Flatten Layer [9]**

4) Dropout

Dropout is used to randomly remove neurons temporarily, which were considered as noise, aiming to prevent overfitting [9].

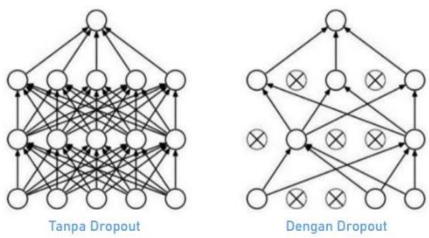


**Fig. 4. Illustration of Without Dropout and With Dropout [9]**

5) Full Connected Layer

Fully Connected layer is a crucial component in a neural network consisting of a number of nodes that were fully connected to the previous layer. Its main function is to combine the features that have been previously extracted into the corresponding classes. The process begins with the random initialization of weights and biases, which were then used to multiply with the input. If the result is still far from expected, the weights and biases were updated using the appropriate learning algorithm, and this process is repeated until the optimal weights and biases were obtained [11]. The Fully Connected layer is the result of the convolution process that receives input from the previous layers and is responsible for extracting the features that were most correlated with a particular class. Each activation neuron from the previous layer is connected to every neuron in the Fully Connected layer, allowing for comprehensive information processing. This layer typically consists of multiple hidden layers, activation functions, output layers, and loss functions used for classification [1]. The process of combining all neurons into one-dimensional data that has gone through the feature learning process is generally done with flatten, transforming it into a vector [9].

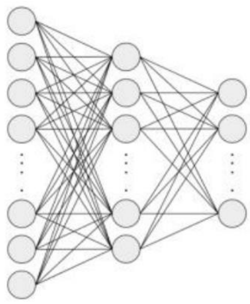

**Fig. 5. Illustrates the Fully Connected layer [9]**

*5. Flask*

Flask is a web framework that provides you with tools, libraries, and technologies to build web applications. These web applications can range from web pages, blogs, and wikis to web-based calendar applications or commercial websites. Flask is a lightweight web application framework written in Python and based on the WSGI toolkit and Jinja2 template engine. Flask falls into the category of microframeworks, which typically have few or no dependencies on external libraries [28].

*6. Python*

Python is an interpreted programming language designed for general purposes, with a focus on code clarity. It aims to be a language that combines functionality and capability with very clear syntax. Additionally, Python comes with a comprehensive standard library of functions [21]. Python supports various programming paradigms, including object-oriented, imperative, and functional programming. Python is also known as a dynamic programming language with automatic memory management. Although often used as a scripting language, Python can be used for various software development needs and can run on various operating system platforms [18].

## III. METHODOLOGY

In this study, Convolutional Neural Network (CNN) is selected as the primary method. The methodology includes data preparation, CNN model construction, and model training. The workflow of the CNN method used in this research is illustrated in Figure 6.

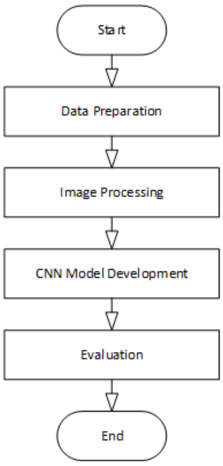


**Fig. 6. Research Flow of Method**

Convolutional Neural Network (CNN) or ConvNet is a method commonly used for image analysis [23]. The CNN method can extract features in an unsupervised manner [11]. CNN has been proven to have excellent accuracy and can handle large volumes of data. In 2012, CNN achieved image recognition accuracy levels that rival human performance on certain datasets [13]. Overall, Convolutional Neural Network (CNN) or ConvNet is a highly effective method in image analysis. CNN can extract important features from images with high accuracy, similar to the way the human neural network operates. With its ability to recognize objects, extract features, and classify images, CNN has had a significant impact in image recognition and is one of the important algorithms in deep learning.

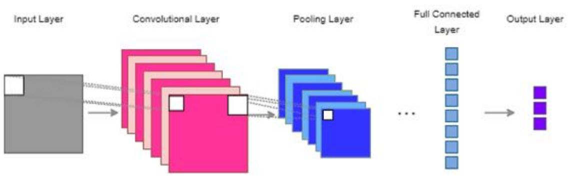


**Fig. 7. Convolutional Neural Network Architecture**

### 1. *Data*

The data used in this study is a dataset consisting of images of Indonesian Sign Language (BISINDO) alphabet hand signs, downloaded from Kaggle OpenDataset. This dataset was collected by a Kaggle user named Agung Ma'ruf from various sources for the purpose of analysis and development of the BISINDO alphabet hand sign recognition model [29]. The dataset consists of a number of images showing hand signs for each letter of the BISINDO alphabet, from the letter A to Z. The dataset can be visualized as follows:

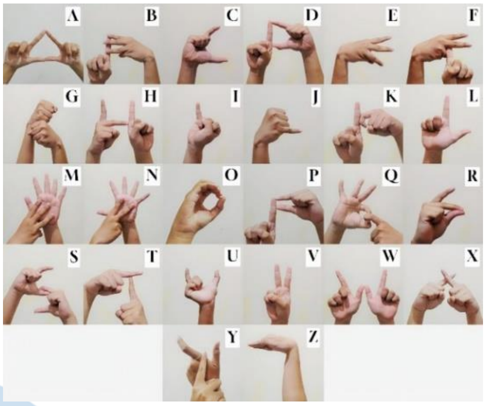


**Fig. 8. Indonesian Sign Language (BISINDO) Alphabet Hand Sign Dataset [29]**

This dataset consists of a number of photos taken to represent each BISINDO alphabet hand sign. Each photo is labeled according to the letter it represents, which will be used to train and test the BISINDO alphabet hand sign recognition model.

### 2. *Data Preparation*

Data preparation in the development of the Indonesian Sign Language (BISINDO) alphabet hand sign recognition system using Convolutional Neural Network (CNN) is an important stage. This stage includes collecting hand sign data, preprocessing data, encoding labels, data augmentation, and data splitting. The process in this stage is as follows:

1) Data Collection

The Indonesian Sign Language (BISINDO) alphabet hand sign data was collected from Kaggle's OpenDataset, with hand sign images that have been properly labeled. This dataset includes hand sign images for each letter of the BISINDO alphabet, with variations in position, angle, and lighting. Data collection was done by ensuring a balanced representation of each hand sign class, so that the model can learn well from the variations present in the dataset. After the data was collected, a validation process was conducted to ensure the accuracy and reliability of the data before using it in model training. This dataset consists of 7131 images with a white background, has 26 classes, and a total dataset size of 876 MB. An example of the hand sign image for the letter "A" can be seen in Figure 9.

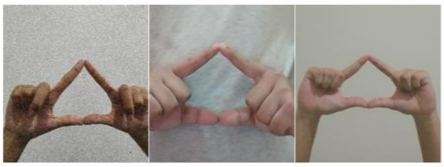

Fig. 9. Sample of BISINDO Alphabet Hand Sign "A" [29]

2) Data Preprocessing

During the data preprocessing phase, the dataset consisting of 7131 images categorized into 26 classes was uploaded to Google Colab for conversion into pixel representations. After reading the BISINDO alphabet hand sign images, these images were converted into pixel matrices. Each image, originally represented as a 2D matrix, was resized into a 28x28 pixel matrix. This step was taken to transform the visual representation of the images into a numerical representation that can be processed by the Convolutional Neural Network (CNN) model. For example, the hand sign image of the letter 'A' would be converted into a 28x28 pixel matrix, where each matrix element represents the color intensity value at one pixel. After that, the separation between features (pixel matrices) and labels (alphabet letter classes) was performed. Once the pixel matrices were formed from the images, the next step was to separate the features (pixel matrices) and labels (alphabet letter classes) from the dataset. The 'label' column containing information about the alphabet letter in each image was removed from the feature dataset. For instance, if there was a hand sign image of the letter 'A' with the label '0', that image was removed from the feature dataset, and the label '0' was stored separately for use in model training. The data preprocessing flowchart can be seen in Figure

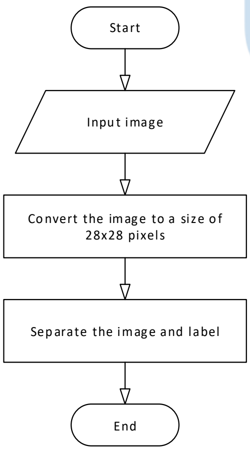


Fig. 10 Data Collection Flowchart

This separation is important so that the CNN model can correctly learn the relationship between the images (features) and the labels. By separating the features and labels, the model can learn to recognize patterns in the images that correspond to the correct labels.

3) Label Encoding

The next step is to encode the labels in the dataset. Encoding is done so that the labels can be processed more efficiently by the model. In the case of BISINDO alphabet hand sign recognition, the labels were generally represented in the form of numbers, for example, from 0 to 25 (for letters A to Z). The label encoding flowchart can be seen in Figure 11.

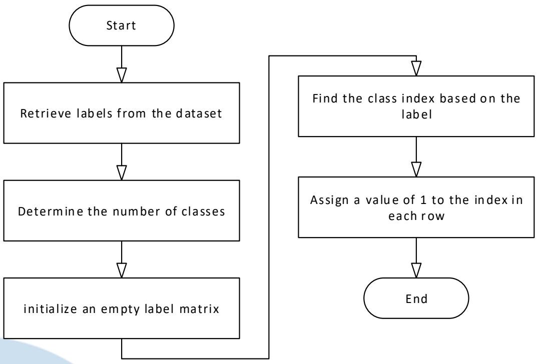


Fig. 11. Label Encoding Flowchart

One of the encoding techniques that can be used is one-hot encoding, where each label is represented as a binary vector with a length equal to the number of classes. For example, if there were 26 classes (A to Z), then each label will be represented as a vector of length 26, where only the index corresponding to the label class has a value of 1, while the other indices have a value of 0.

4) Data Augmentation

In this stage, data augmentation is performed on the dataset of BISINDO alphabet hand sign images. Data augmentation is a technique used to create variations in the training dataset by manipulating the original images. Data augmentation aims to decrease overfitting and enhance the model's capacity to generalize to unseen data. The data augmentation process is carried out using the ImageDataGenerator object from TensorFlow. The data augmentation is performed with the following parameters:

- rotation_range = 10,
- width_shift_range=0.1,
- height_shift_range=0.1,
- shear_range=0.1,
- zoom_range=0.1,
- horizontal_flip=True,
- fill_mode='nearest'

After configuring the data augmentation, the augmentation process is applied to the training data (`x_train`) using the `fit` method of the ImageDataGenerator object.

5) Data Splitting

Following the completion of the data augmentation stage, the next step involves splitting the dataset into two portions: the training data and the test data. The purpose is to test the model's performance on data it has never seen before, providing a better indication of how well the model can generalize. The data is split using the parameter test_size=0.2, which means that out of the total 7131 data points, the test data will consist of 20% of the total data, which is 1426 test data points, while the training data will consist of 80% of the total data, which is 5704 training data points. The splitting is done randomly using the parameter random_state=101 to ensure consistent results each time the code is executed. After the data splitting is completed, four sets of data will be formed: x_train (training feature data), y_train (training label data), x_test (test feature data), and y_test (test label data). The training data will be utilized to train the model, while the test data will be employed to assess the model's performance post-training.

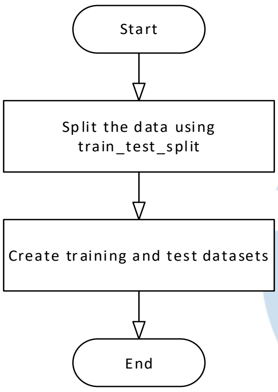


Fig. 12. Data Splitting Flowchart

*3. CNN Model Development*

To develop the Convolutional Neural Network (CNN) model for recognizing BISINDO alphabet hand signs, the Python programming language is used along with the TensorFlow/Keras library. The CNN model flowchart can be seen in Figure 13.

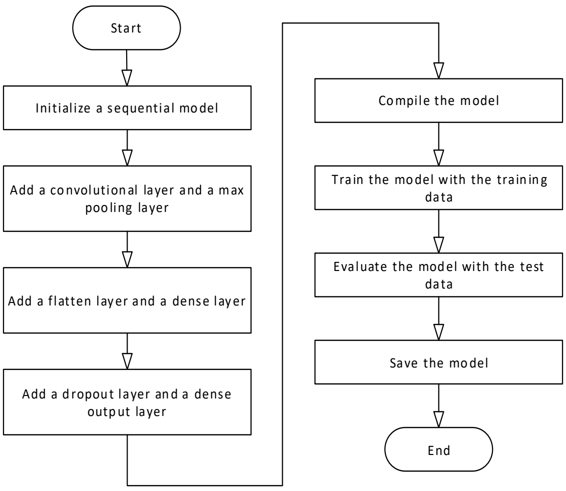


**Fig. 13. CNN Model Development Flowchart**

The sequential model used can be seen in Figure 14.

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 26, 26, 64) | 640 |
| max_pooling2d (MaxPooling2D) | (None, 13, 13, 64) | 0 |
| conv2d_1 (Conv2D) | (None, 11, 11, 128) | 73,856 |
| max_pooling2d_1 (MaxPooling2D) | (None, 5, 5, 128) | 0 |
| conv2d_2 (Conv2D) | (None, 3, 3, 256) | 295,168 |
| max_pooling2d_2 (MaxPooling2D) | (None, 1, 1, 256) | 0 |
| flatten (Flatten) | (None, 256) | 0 |
| dense (Dense) | (None, 512) | 131,584 |
| dropout (Dropout) | (None, 512) | 0 |
| dense_1 (Dense) | (None, 26) | 13,338 |

Total params: 514,586 (1.96 MB)

Trainable params: 514,586 (1.96 MB)

Non-trainable params: 0 (0.00 B)

**Fig. 14. Sequential Model**

After training is completed, the model is evaluated using the test data to measure the accuracy and loss of the model. Finally, the model is saved for use in predicting hand signs.

*4. Training*

In training the Convolutional Neural Network (CNN) model for Indonesian Sign Language (BISINDO) alphabet hand sign recognition, the `fit` method is used on the model object. The training process utilizes the training data (x_train, y_train) as input, with a batch size of 128 and 30 epochs. The verbose=1 parameter is used to monitor the training progress at each epoch, while the evaluation metric employed is accuracy to gauge the model's performance in correctly classifying data. The iteration process runs for 30 iterations, where each iteration experiences fluctuations in accuracy on the validation data, showing both increases and decreases. This can be observed in Figure 15:

```
Epoch 1/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 49s 708ms/step - accuracy: 0.0466 - loss: 3.2529 - val_accuracy: 0.1781 - val_loss: 3.0227
Epoch 2/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 34s 595ms/step - accuracy: 0.2848 - loss: 2.5026 - val_accuracy: 0.7980 - val_loss: 0.9708
Epoch 3/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 40s 562ms/step - accuracy: 0.7024 - loss: 1.0082 - val_accuracy: 0.9011 - val_loss: 0.4885
Epoch 4/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 46s 666ms/step - accuracy: 0.8471 - loss: 0.5791 - val_accuracy: 0.9264 - val_loss: 0.3123
Epoch 5/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 40s 629ms/step - accuracy: 0.9098 - loss: 0.3556 - val_accuracy: 0.9306 - val_loss: 0.2842
Epoch 6/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 55s 945ms/step - accuracy: 0.9127 - loss: 0.3346 - val_accuracy: 0.9397 - val_loss: 0.2056
Epoch 7/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 68s 615ms/step - accuracy: 0.9389 - loss: 0.2285 - val_accuracy: 0.9453 - val_loss: 0.2028
Epoch 8/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 42s 625ms/step - accuracy: 0.9390 - loss: 0.2225 - val_accuracy: 0.9481 - val_loss: 0.1582
Epoch 9/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 39s 583ms/step - accuracy: 0.9455 - loss: 0.1891 - val_accuracy: 0.9439 - val_loss: 0.1586
Epoch 10/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 48s 738ms/step - accuracy: 0.9531 - loss: 0.1636 - val_accuracy: 0.9558 - val_loss: 0.1405
Epoch 11/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 37s 640ms/step - accuracy: 0.9613 - loss: 0.1435 - val_accuracy: 0.9593 - val_loss: 0.1183
Epoch 12/30
45/45 ━━━━━━━━━━━━━━━━━━━━ 41s 632ms/step - accuracy: 0.9655 - loss: 0.1207 - val_accuracy: 0.9572 - val_loss: 0.1241
Epoch 13/30
```

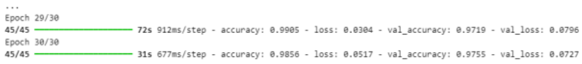


**Fig. 15. Iteration Process**

The training results, including accuracy and loss on the training and validation data, will be stored in a variable named `history` for further analysis. Visualization of training metrics, such as accuracy and loss graphs, will be used to understand the model's performance during the training process.

*5. Training*

In the evaluation stage, the Convolutional Neural Network (CNN) model is evaluated using separate test data (x_test, y_test) to measure the model's performance on unseen data. Evaluation is done using the evaluate method on the model object, which provides the loss and accuracy values on the test data. Additionally, the sklearn.metrics library is used to calculate the accuracy score of the model on the test data. The accuracy score provides information about how well the model can classify BISINDO alphabet hand signs on unseen data. This testing process is important to validate the model's performance before it is used in real-world applications. Finally, the training results of the model were visualized to understand the performance trends of the model during the training process.

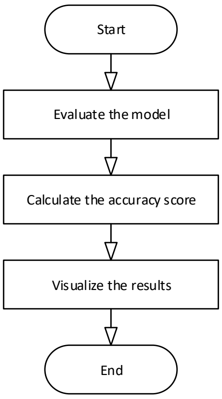


**Fig. 16. Testing Flowchart**

This process helps evaluate whether the model experiences overfitting or underfitting and determines if adjustments were needed in the architecture or parameters of the model. The results of this testing serve as the primary evaluation basis to determine how well the CNN model that has been built can be used in the BISINDO alphabet hand sign recognition application.

## IV. RESULT AND DISCUSSION

In this section, the results of testing the BISINDO alphabet hand sign recognition system using Convolutional Neural Network (CNN) were presented to evaluate the system's performance in recognizing hand signs in the prepared dataset. The following were the detailed test results obtained:

*1. Evaluation Metrics*

Evaluation metrics were used to measure the performance of the Convolutional Neural Network (CNN) model that has been built. Evaluation is done using four main metrics, namely accuracy, recall, F1 score, and precision. The details of each metric were as follows:

TABLE I
EVALUATION METRICS

| Accuracy | Recall | F1 Score | Precision |
|---|---|---|---|
| 97,5% | 97,5% | 97,5% | 97,6% |

*2. Training and Validation Data*

In the training stage of the Convolutional Neural Network (CNN) model for BISINDO alphabet hand sign recognition, the data is divided into two main parts: training data and validation data, which were taken from the test data. The training data is used to train the model, while the validation data is used to measure the model's performance during the training process.

1) Training and Validation Data Accuracy
In the final epoch, the model training is stopped after the 30th iteration, as specified by the batch size of 128. At this point, the evaluation accuracy reaches 97.5%. The model also achieves an accuracy of 97.5% for the training dataset, demonstrating the model's ability to recognize hand signs with a high level of accuracy. The accuracy graph can be seen in Figure 17.

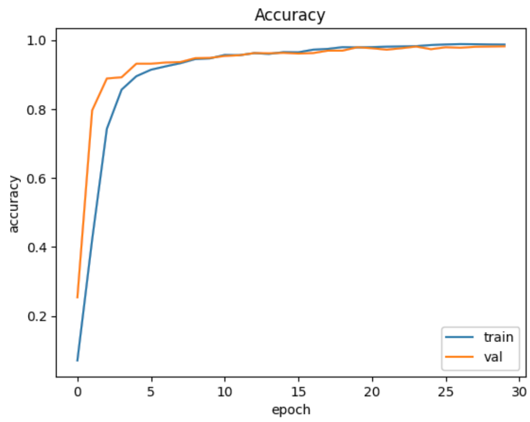


**Fig. 17. Graph od Training and Validation Data Accuracy**

2) Training and Validation Data Loss
In the final epoch, the model training is stopped after the 30th iteration, as specified by the batch size of 128. At this point, the evaluation loss reaches 0.076, indicating that the model has a low error rate in making predictions. The detailed loss for the training dataset

also reaches 0.076, indicating that the model can recognize hand signs with a high level of accuracy. The loss graph can be seen in Figure 18.

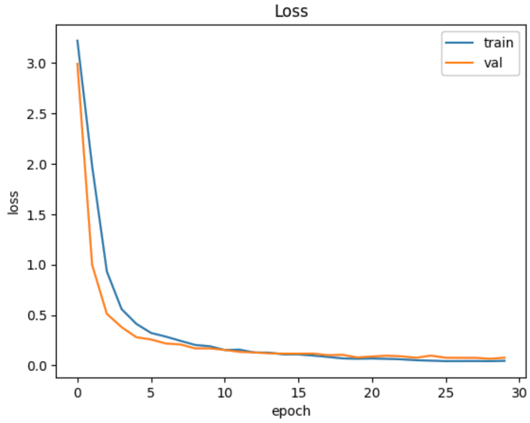


**Fig. 18. Graph of Training and Validation Data Loss**

Previous training was conducted until a satisfactory accuracy value was achieved, resulting in an optimal model. Here is a comparison of the Training Accuracy, Training Loss, Validation Accuracy, and Validation Loss values in Table 2.

TABLE II
COMPARISON OF ACCURACY AND LOSS VALUES PER EPOCH

| Epoch | Train Accuracy | Train Loss | Val Accuracy | Val Loss |
|---|---|---|---|---|
| 5 | 0.9098 | 0.3556 | 0.9306 | 0.2842 |
| 10 | 0.9531 | 0.1636 | 0.9558 | 0.1405 |
| 15 | 0.9728 | 0.0903 | 0.9621 | 0.1111 |
| 20 | 0.9858 | 0.0533 | 0.9712 | 0.0913 |
| 25 | 0.9925 | 0.0331 | 0.9804 | 0.0599 |
| 30 | 0.9856 | 0.0517 | 0.9755 | 0.0727 |

### 3. *Prediction Test*

The prediction test is conducted to evaluate the performance of the Indonesian Sign Language (BISINDO) alphabet hand sign recognition system that has been developed. Prediction is done using Flask to run the previously designed CNN model. The prediction test is conducted on a prediction dashboard via a web interface that displays a webcam prediction feature to predict alphabet hand signs in real-time. Below were sample prediction results that can be seen in the images below:

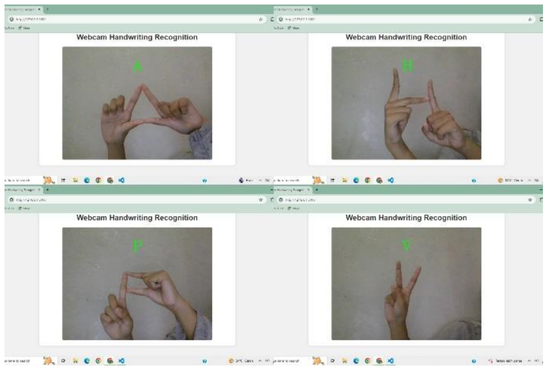


**Fig. 19. Sample Prediction Result**

After the prediction process, a prediction table is created to present the results of predicting the letters of the alphabet hand sign. The table contains correctly and incorrectly predicted letters based on the test data used in the evaluation. An example of a prediction table can be seen in Table 3 below:

TABLE III
PREDICTED LETTER RESULT

| No. | Letter | Prediction Result |
|---|---|---|
| 1. | A | Correct |
| 2. | B | Correct |
| 3. | C | Correct |
| 4. | D | Correct |
| 5. | E | Correct |
| 6. | F | Correct |
| 7. | G | Incorrect |
| 8. | H | Correct |
| 9. | I | Correct |
| 10. | J | Correct |
| 11. | K | Correct |
| 12. | L | Correct |
| 13. | M | Correct |
| 14. | N | Correct |
| 15. | O | Correct |
| 16. | P | Correct |
| 17. | Q | Correct |
| 18. | R | Correct |
| 19. | S | Correct |
| 20. | T | Correct |
| 21. | U | Correct |
| 22. | V | Correct |
| 23. | W | Correct |
| 24. | X | Correct |
| 25. | Y | Correct |
| 26. | Z | Correct |

### 4. *Comparison with Previous Research*

In this study, input images of size 28x28 pixels were used, and training was conducted for 30 epochs. The results show that the proposed algorithm achieved an accuracy of 97.54%. The comparison of the accuracy of this research with previous research can be seen in Table 4:

TABLE IV
COMPARISON OF ACCURACY WITH PREVIOUS RESEARCH

| No. | Method | Data | Input Size | Accuracy (%) |
|---|---|---|---|---|
| 1. | *Convolutional Long Short Term Memory* (LTSM) [3] | Alphabet Sign Language BISINDO | 100 x 100 | 68% |
| 2. | *Local Directional Pattern &* Klasifikasi *K-Nearest Neighbour* (KNN) [14] | *Turkey Ankara Ayrancı Anadolu High School's Sign Language Digits* | 100 x 100 | 88,45% |
| 3. | *Convolutional Neural Network* (CNN) [9] | Alphabet Sign Language SIBI | 200 x 200 | 90,05% |
| 4. | *Convolutional Neural Network* (CNN) | Alphabet Sign Language BISINDO | 28 x 28 | 97,54% |

From the table above, it can be seen that the developed method has achieved a higher level of accuracy compared to previous research. This difference indicates significant progress in the development of a better model.

*5. Prediction Test*

In testing the Indonesian Sign Language (BISINDO) alphabet sign recognition system using Convolutional Neural Network (CNN), evaluation metrics such as accuracy, recall, precision, and F1 score provided satisfactory results. An accuracy of 97.5% indicates a high level of accuracy in recognizing sign language. The same recall value indicates that the model is able to recognize most of the sign language that should be positive, while a precision of 97.6% indicates that most of the sign language classified as positive by the model is indeed positive. An F1 score of 97.5% indicates that the model has a good balance between precision and recall, making it reliable in recognizing BISINDO alphabet sign language.

During the training and validation stages, the model successfully achieved a high level of accuracy on their respective datasets. This indicates that the model did not experience overfitting and was able to generalize hand sign patterns well on new, unseen data. Thus, the developed Convolutional Neural Network (CNN) model has a strong capability in recognizing Indonesian Sign Language (BISINDO) alphabet signs.

The results of the prediction test demonstrate the model's ability to recognize sign language in real-time with high accuracy. By using Flask to run the previously designed CNN model, the model can provide predictions for BISINDO alphabet sign language directly through a web interface. The webcam prediction feature allows for direct sign language prediction through the webcam.

During the testing process, the system successfully predicted 25 out of 26 alphabet letters as predicted by the sign language recognition system. However, the prediction results indicate that there is still room for improvement in detecting one letter that was not predicted well out of the total of 26 Indonesian Sign Language (BISINDO) alphabet letters. With further evaluation and necessary adjustments, it is hoped that the system can improve its ability to recognize all alphabet letters with high accuracy.

## V. CONCLUSION

Based on the research findings, analysis, method implementation, and discussion, this study successfully developed a hand gesture recognition system for the Indonesian Sign Language alphabet (BISINDO) using Convolutional Neural Networks (CNN). With an accuracy rate of 97.5% and a good balance between recall, precision, and F1 score, this system demonstrates excellent performance in recognizing hand gesture patterns/images of the BISINDO alphabet. The implementation of the system for real-time prediction through a web interface using Flask also enhances its accessibility. However, there is room for improvement, particularly in recognizing a specific letter that has not been predicted accurately. Therefore, the continuous development of this system is necessary to further enhance accuracy and proficiency in recognizing hand gestures more comprehensively.

## REFERENCES


[1] O. D. Nurhayati, D. Eridani, and M. H. Tsalavin, “Sistem Isyarat Bahasa Indonesia (SIBI) Metode Convolutional Neural Network Sequential secara Real Time,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 4, pp. 819–828, 2022, doi: 10.25126/jtiik.2022944787.

[2] I. J. Thira, D. Riana, A. N. Ilhami, B. R. S. Dwinanda, and H. Choerunisya, “Pengenalan Alfabet Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Convolutional Neural Network,” *J. Algoritm.*, vol. 20, no. 2, pp. 421–432, 2023, doi: 10.33364/algoritma/v.20-2.1480.

[3] A. Dwi Baitur Rizky, M. Aulia Faqihuddin, F. Fatha Romadhan, and I. Agustien Siradjuddin, “Identifikasi Alfabet Bahasa Isyarat Indonesia dengan Menggunakan Convolutional LSTM,” *Pros. SENIATI*, vol. 7, no. 2, pp. 183–190, 2023, doi: 10.36040/seniati.v7i2.7925.

[4] L. Arisandi and B. Satya, “Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network,” *J. Sist. Cerdas*, vol. 5, no. 3, pp. 135–146, 2022, doi: 10.37396/jsc.v5i3.262.

[5] A. R. Syulistyo, D. S. Hormansyah, and P. Y. Saputra, “SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN),” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 732, no. 1, 2020, doi: 10.1088/1757-

899X/732/1/012082.

[6] Suryati, “Bab 1 pendahuluan,” *Pelayanan Kesehat.*, no. 2015, pp. 3–13, 2019.

[7] D. Permana and J. Sutopo, “APLIKASI PENGENALAN ABJAD SISTEM ISYARAT BAHASA INDONESIA (SIBI) DENGAN ALGORITMA YOLOv5 MOBILE APPLICATION ALPHABET RECOGNITION OF INDONESIAN LANGUAGE SIGN SYSTEM (SIBI) USING YOLOv5 ALGORITHM,” *J. SimanteC*, vol. 11, no. 2, pp. 231–240, 2023.

[8] S. N. Amartama, A. N. Hidayah, P. K. Sari, and R. A. Ramadhani, “Implementasi Convolutional Neural Network (CNN) dalam Pengenalan Pola Penulisan Tangan,” *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 133–138, 2024, doi: 10.29407/stains.v3i1.4155.

[9] M. Sholawati, K. Auliasari, and F. Ariwibisono, “Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn),” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 134–144, 2022, doi: 10.36040/jati.v6i1.4507.

[10] F. Akbar, A. Hidayatno, and A. Triwiyatno, “Perancangan Program Pengenalan Isyarat Tangan Dengan Metode Convolutional Neural Network (Cnn),” *Transient J. Ilm. Tek. Elektro*, vol. 9, no. 1, pp. 26–36, 2020, doi: 10.14710/transient.v9i1.26-36.

[11] M. E. Al Rivan and S. Hartoyo, “Klasifikasi Isyarat Bahasa Indonesia Menggunakan Metode Convolutional Neural Network,” *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 2, pp. 364–373, 2022, doi: 10.28932/jutisi.v8i2.4863.

[12] R. Setiawan, Y. Yunita, F. F. Rahman, and H. Fahmi, “BISINDO ( Bahasa Isyarat Indonesia ) Sign Language Recognition Using Deep Learning,” vol. 09, no. 01.

[13] N. Kasim and G. S. Nugraha, “Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode Convolution Neural Network,” *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA )*, vol. 3, no. 1, pp. 85–95, 2021, doi: 10.29303/jtika.v3i1.136.

[14] N. Ilmi and H. Suryoprayogo, “Pengenalan Angka Bahasa Isyarat dengan Menggunakan Local Directional Pattern dan Klasifikasi K-Nearest Neighbour,” *J. Informatics Commun. Technol.*, vol. 4, no. 1, pp. 48–55, 2022, doi: 10.52661/j_ict.v4i1.103.

[15] S. Khotijah, J. Juliana, and D. Driyani, “Perancangan Media Pembelajaran Interaktif Bahasa Isyarat Bisindo Untuk Penyandang Disabilitas Tuna Rungu Berbasis Android,” *J. Ilm. Multidisiplin*, vol. 2, no. 1, pp. 142–149, 2023, doi: 10.59000/jim.v2i1.101.

[16] A. Nur and H. Nugroho, “Deteksi Fitur Huruf Sistem Isyarat Bahasa Indonesia menggunakan Metode Chain Code,” *POSITIF J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, pp. 36–40, 2022, doi: 10.31961/positif.v8i1.1133.

[17] 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,” *J. Holistika*, vol. 5, no. 1, p. 28, 2023, doi: 10.24853/holistika.5.1.28-33.

[18] F. M. Abdurrahman, *Pengenalan Bahasa Isyarat Indonesia dengan Algoritma Convolutional Neural Network (CNN) menggunakan Kinect 2.0*. 2019.

[19] A. Alvin, N. H. Shabrina, A. Ryo, and E. Christian, “Hand Gesture Detection for Sign Language using Neural Network with Mediapipe,” *Ultim. Comput. J. Sist. Komput.*, vol. 13, no. 2, pp. 57–62, 2021, doi: 10.31937/sk.v13i2.2109.

[20] B. R. Fajri, A. D. Samala, and F. Ranuharja, “Media Interaktif Pengenalan Bahasa Isyarat Bisindo,” *J. Teknol. Inf. dan Pendidik.*, vol. 13, no. 1, pp. 35–44, 2020, doi: 10.24036/tip.v13i1.293.

[21] G. E. Ripera, M. Hikmatyar, and R. Hartono, “NEURAL NETWORK PADA PENGENALAN AKSARA,” vol. 12, no. 1, 2024.

[22] R. Jannah, M. Walid, and H. Hoiriyah, “Sistem Pengenalan Citra Dokumen Tanda Tangan Menggunakan Metode CNN (Convolutional Neural Network),” *Energy - J. Ilm. Ilmu-Ilmu Tek.*, vol. 12, no. 2, pp. 1–8, 2022, doi: 10.51747/energy.v12i2.1116.

[23] Y. Brianorman and R. Munir, “Perbandingan Pre-Trained CNN: Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah,” *J. Sist. Info. Bisnis*, vol. 13, no. 1, pp. 52–59, 2023, doi: 10.21456/vol13iss1pp52-59.

[24] H. N. Al Falah and K. K. Purnamasari, “Implementasi Convolutional Neural Network Pada Pengenalan Tulisan Tangan,” *Elibrary.Unikom.Ac.Id*, no. 112, 2019.

[25] A. Mulyanto, E. Susanti, F. Rossi, W. Wajiran, and R. I. Borman, “Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR),” *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 1, p. 52, 2021, doi: 10.26418/jp.v7i1.44133.

[26] 127-133. Arciniegas, 2006. Inte- ligencia emocional en estudiantes de la Universidad Autónoma de Los Andes. Revista Conrado, 17(78), “No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title,” no. 2, p. 6, 2021.

[27] D. M. Wonohadidjojo, “Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih,” *Ultim. J. Tek. Inform.*, vol. 13, no. 1, pp. 51–57, 2021, doi: 10.31937/ti.v13i1.2040.

[28] V. Rama Vyshnavi and A. Malik, “Efficient Way of Web Development Using Python and Flask,” *Int. J. Recent Res. Asp.*, vol. 6, no. 2, pp. 16–19, 2019.

[29] A. Ma’ruf, “Indonesian Sign Language - BISINDO,” *kaggle*, 2023. https://www.kaggle.com/datasets/agungmrf/indonesian-sign-language-bisindo.