IJRCS

International Journal of Robotics and Control Systems
Vol. 5, No. 2, 2025, pp. 813-829
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Hand Keypoint-Based CNN for SIBI Sign Language Recognition

Anik Nur Handayani [a,1,*], Sholikhatul Amaliya [a,2], Muhammad Iqbal Akbar [a,3], Muhammad Zaki Wiryawan [a,4], Yeoh Wen Liang [b,1], Wendy Cahya Kurniawan [b,2]

[a] Department of Electrical Engineering and Informatics, Universitas Negeri Malang, Jl. Semarang 5, Malang 65145, Indonesia
[b] Department of Information Science and Engineering, Saga University, 1 Honjomachi, Saga, 840-8502, Japan
[1] aniknur.ft@um.ac.id; [2] amalia.sholikhatul@gmail.com; [3] iqbal.akbar.ft@um.ac.id;
[4] muhammad.zaki.2305348@students.um.ac.id; [5] wlyeoh@cc.saga-u.ac.jp; [6] 23805192@edu.cc.saga-u.ac.jp
* Corresponding Author

ARTICLE INFO

Article history
Received December 23, 2024
Revised February 12, 2025
Accepted February 22, 2025

Keywords
Convolutional Neural Network (CNN);
Hand Keypoint;
Image Classification;
Sistem Isyarat Bahasa Indonesia (SIBI)

ABSTRACT

SIBI is less widely adopted, and the lack of an efficient recognition system limits its accessibility. SIBI gestures often involve subtle hand movements and complex finger configurations, requiring precise feature extraction and classification techniques. This study addresses these issues using a Hand Keypoint-based Convolutional Neural Network (HK-CNN) for SIBI classification. The research utilizes Kinect 2.0 for precise data collection, enabling accurate hand keypoint detection and preprocessing. The optimal data acquisition distance between 50 and 60 cm from the camera is considered to obtain clear and detailed images. The methodology includes four key stages: data collection, preprocessing (keypoint extraction and image filtering), classification using HK-CNN with ResNet-50, EfficientNet, and InceptionV3, and performance evaluation. Experimental results demonstrate that EfficientNet achieves the highest accuracy of 99.1% in the 60:40 data split scenario, with superior precision and recall, making it ideal for real-time applications. ResNet-50 also performs well with 99.3% accuracy in the 20:80 split but requires longer computation time, while InceptionV3 is less efficient for real-time applications. Compared to traditional CNN methods, HK-CNN significantly enhances accuracy and efficiency. In conclusion, this study provides a robust and adaptable solution for SIBI recognition, facilitating inclusivity in education, public services, and workplace communication. Future research should expand dataset diversity and explore dynamic gesture recognition for further improvements.

1. Introduction

Communication is essential to human life as individuals and social beings [1], [2]. However, not all individuals have standard communication skills, especially deaf people who use deaf devices, such as handwriting, message boards, interpreters, and sign language, as the main methods of communication [3], [4]. In the context of the deaf community, the utilization of sign language serves as a vital communication method, encompassing a multitude of non-verbal elements such as body movements, arms, hands, and facial expressions, as well as incorporating orientation and shape. Sign language, a vital communication method for the deaf community, In Indonesia, there are two dominant sign language systems. The first is *Bahasa Isyarat Indonesia* (BISINDO). The second is *Sistem*

http://dx.doi.org/10.31763/ijrcs.v5i2.1745

Isyarat Bahasa Indonesia (SIBI). Both of these languages are unique compared to other sign languages in the world [5]. BISINDO and SIBI add to the world's richness and diversity of sign languages with their respective uniqueness [6]. BISINDO, with its nature and regional variations, reflects Indonesia's cultural diversity. With its more formal and structured approach, SIBI aims to facilitate education and formal communication [7]. Together, they provide an essential communication tool for the deaf community in Indonesia, demonstrating how sign language can evolve and adapt to various cultural and social contexts [8]. SIBI is officially used in Special Schools that provide students with special needs under the Ministry of Education and Culture auspices, based on the Regulation of the Minister of Education of the Republic of Indonesia, number 0161/U/1994. SIBI is undoubtedly beneficial for deaf people who want to communicate in Indonesia.

Deaf people often face obstacles in communicating with the general public [9], [10]. Not all places or situations provide facilities or systems to help deaf people communicate with hearing people. [11]. This hinders the active participation of deaf people in various activities. These communication barriers highlight the need for a more inclusive and efficient solution. This certainly hinders the active involvement of deaf people in various activities [12]. These problems underlie this research to classify SIBI using image processing. Image processing is a form of implementation that can be used as a communication translation system between people with disabilities and the community [13], [14]. Image processing uses a data capture method in the form of static or dynamic images with 2 or 3 dimensions (2D-3D) that can represent the hand movements of deaf people. [15]-[17]. From commonly conducted research, using CNN architecture integrated with the hand gesture recognition method has proven effective in identifying various hand gestures [18]. According to Pribadi et al. [19] the algorithm used is Deep Learning Convolutional Neural Network (CNN) and the Hand Gesture Recognition method. This method allows the system to recognize gestures with high accuracy, which is especially important in applications such as sign language recognition. However, one of the main drawbacks of this method is the need for a considerable amount of training data to train the model well. Providing this large amount of training data can be very wasteful in terms of storage space and time required for the training process.

Therefore, this study offers an alternative approach using hand key points convolutional neural network (HK-CNN) as the main feature in the SIBI classification process. Using HK-CNN allows for significant savings in the training data space without sacrificing the effectiveness and accuracy of classification [20], [21]. By identifying key points on the hand, this method can reduce the amount of data that needs to be stored and processed while still maintaining high performance in recognizing various gestures [22], [23]. This approach is not only more efficient in the use of storage space but also speeds up the model training process, making it more practical and cost-effective in actual implementation.

However, while HK-CNN offers benefits regarding data efficiency and computational cost, it also presents several challenges. The accuracy of keypoint detection is highly sensitive to variations in hand shapes and lighting conditions. These factors can significantly impact the robustness and generalizability of the model, especially in real-world applications [24], [25]. Variability in hand size, occlusions, and differences in skin tone may affect keypoint detection accuracy, leading to potential misclassification. Additionally, changes in lighting conditions can introduce noise in image processing, reducing model reliability [26], [27]. Addressing these challenges ensures the proposed method performs consistently across diverse environments. Future improvements, such as advanced pre-processing techniques, adaptive keypoint detection models, and robust noise-handling mechanisms, can enhance the overall effectiveness of HK-CNN for SIBI recognition.

2. Method

To build an effective translation system, image processing must be integrated with artificial intelligence and the Hand Keypoint method for data classification. [28]-[30]. Hand Keypoint Detection techniques are often applied using Convolutional Neural Network (CNN) models trained to detect these key points from images or videos. This CNN model can identify relevant visual patterns and extract essential features from image inputs, allowing the system to understand and translate hand gestures into language that computers can understand.

The methods used in this study include four main stages: Data Collection, Data Pre-processing, Classification, and Evaluation, which are shown in Table 1. The methodology consists of important steps from data processing to model creation and evaluation of its performance. These steps are designed to ensure the accuracy and reliability of the model in classifying SIBI. The following is a brief explanation of each stage in the methodology.

Table 1. Research flow

Stage	Process	Output
Data Collection	Collecting data from	Raw Image Dataset
Data Preprocessing	a. Finding The Center of HandKeypoint	Palm Image with HandKeypoint
	b. Image Filtering	Filtered Palm Image
	c. Image Scaling	Image with smaller pixels
	d. Image Labelling	Labeled Image
Classification	Classification with HK-CNN with Architecture: 1.Resnet50 2.EfficientNet 3.Inception V3	Figure with 24 Classes
Evaluation	Evaluation using Confusion Matrix to evaluate HK-CNN Performance	Accuracy, Precision, Recall, and F1 Score

2.1. Data Collection

Data collection is crucial in this study, which focuses on classifying the SIBI using image processing techniques. Valid and representative data are essential to ensure that the model built accurately recognizes hand movements that represent the letters in SIBI. Therefore, selecting the right methods and tools to collect data is important in this study.

To ensure the data collected is of high quality, a Kinect XBOX ONE 2.0 camera capable of recording hand movements with high precision is used. This camera was chosen because it can detect key points on the hand, which is crucial for the subsequent image processing process. The optimal data acquisition distance between 50 and 60 cm from the camera is also considered to obtain clear and detailed images of hand movements [31], [32]. The Kinect camera is configured with an RGB camera resolution of 1920×1080 (Full HD) at 30 FPS, using the BGR (8-bit per channel) format to capture high-quality color images [33], [34]. Additionally, the depth camera operates at a resolution of 512×424 with a frame rate of 30 FPS and 16-bit depth, providing accurate depth information for precise hand keypoint tracking. The camera's Field of View (FoV) of 70° horizontal and 60° vertical ensures a wide and detailed capture area, enhancing the accuracy of hand keypoint detection.

The research subjects comprised 12 models involving students from the Department of Electrical Engineering, State University of Malang. The diversity of subjects, with five female models and seven male models, provided the necessary variety to ensure the models could recognize the hand movements of various individuals [35]. Each model performs a hand gesture for each letter in SIBI, resulting in 2400 frames per letter, with a total of 57,600 frames of data obtained. The data used in this study is 5,760 image data. Frame selection is meticulous, capturing the highest quality start and end frames to ensure the data used in model training and testing has an accurate and consistent representation. The data used was 24 letters, of which 2 letters, namely j, and z, were not used because they had a dynamic shape and were very difficult to classify.

With these steps, the data collection in this study is expected to provide a solid basis for developing an effective Convolutional Neural Network (CNN) model in classifying hand gestures in SIBI. This approach improves the model's accuracy and contributes to creating a more inclusive and efficient sign language translation system that the deaf community in Indonesia can use.

2.2. Data Preprocessing

The acquired image data underwent preprocessing to enhance the system's image-processing capabilities [36]. The first step in this preprocessing involves detecting hand keypoints on hand image data captured by the camera. These key points are essential for various applications, including hand gesture recognition, sign language interpretation, and human-computer interaction. Hand keypoint detection entails pinpointing specific points on the hand, which are vital for constructing models using Convolutional Neural Network [37], [38].

After finding the middle point of the joint on the finger, cropping or cutting the image is carried out. This stage aims to cut the hand image data with an initial size of 1920×1080 pixels, which is cut to a length of 900×900 pixels. This is done to get data that is focused on the hand area. Here is the equation.

$$\begin{aligned} w &= (x + 450\,) - (x - 450)\,, \\ h &= (y + 450) - (y - 450) \end{aligned} \tag{1}$$

x: x-coordinate of the midpoint at the knuckle in the original image.
y: y-coordinate of the midpoint at the knuckle in the original image.
w: width of the cropped image.
h: height of the cropped image.

Equation (1) defines a square cropping region centred at the midpoint of the knuckle, ensuring that the hand remains the primary focus. The knuckle midpoint (x, y) is the central reference point for cropping. The cropped image's width (w) and height (h) are set to 900 pixels. The cropping boundaries are determined by expanding 450 pixels in all directions left and right for width and top and bottom for height. This method ensures consistency across all processed images by maintaining a fixed cropping size of 900×900 pixels. Such standardization is essential for machine learning models, particularly convolutional neural networks, which require uniform input dimensions for optimal performance. The cropping illustration can be seen in Fig. 1. The results of this process can be seen in Fig. 2.

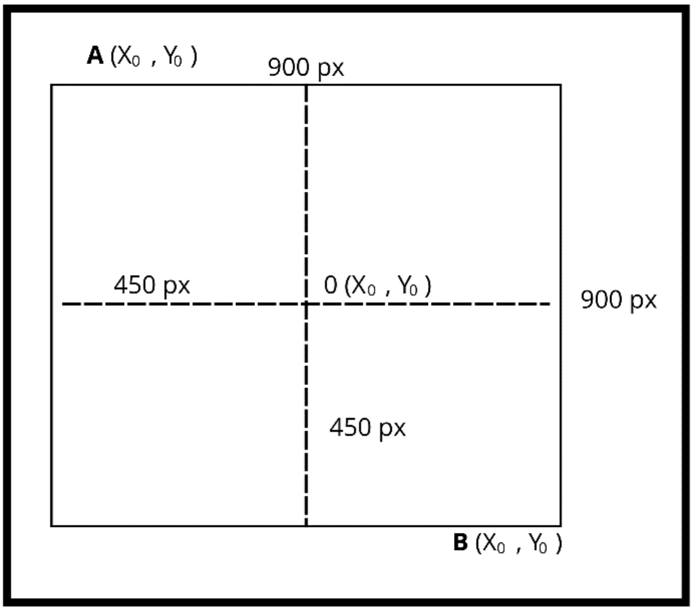

Fig. 1. Cropping illustration

After all data is cropped, next preprocessing stage is filtering using a histogram of the dataset. This process separates low-quality hand images, shown by blurry or blurry hand images. Blurry hand images called blurry distortions, if depicted through a histogram or frequency spectrum, will shift the image frequency spectrum towards low frequencies. The frequency shift in the histogram of the blurry image is lower than that of the standard image [39]. In this research, the histogram is computed using 256 bins with an intensity range of 0–255 for each color channel (B, G, R) [40], [41]. A blurry image is identified based on the variance of the Laplacian, where images with a variance below a predefined threshold (e.g., 100) are considered blurry [42], [43]. Later, from the results, we will see an image that is indicated to be blurry and filtered to get a clean and clear image.

Fig. 3, Fig. 4 a and Fig. 4 b compare blurry and normal images using hand keypoint and histogram results. In the hand keypoint results, the blurry image has an irregular skeleton much different from the standard image. This proves that blurry images are unsuitable for training or testing data in this study [44]. Likewise, the histogram results of blurry images and standard images look different.

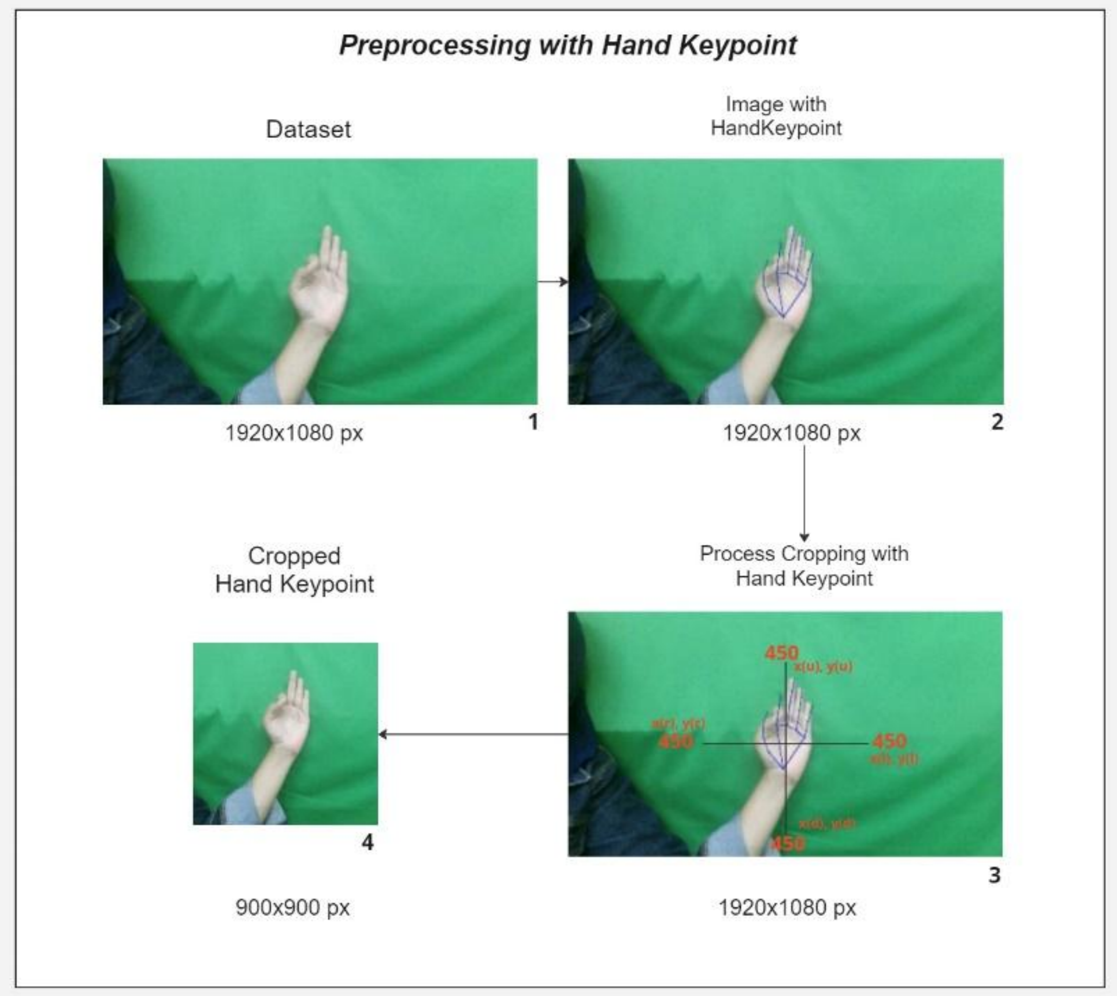

Fig. 2. Preprocessing image with hand keypoint

After obtaining filtered data, data augmentation techniques such as scaling are applied to adjust the size of the hand image. This process enhances system performance and enables faster image processing. In this case, the hand image, initially 900×900 pixels, is scaled down to 150×150 pixels [45]. Scaling results can be seen in Fig. 5.

Last process in preprocessing is data labeling. Each hand-drawn data has a class label used as a parameter in the training process. Each of the labels represents 24 classes of hand image data. The data is divided into lists X and Z, which store image and label datasets. The image data will be stored in list X by converting it into an array, and other hand label data will be stored in list Z. Continuing data labeling process, data in the Z list will be obtained parameters and converted into new data, namely in the form of an array with a total of 24 categories and stored in list Y. While the image that is 57 in the X list will be normalized by dividing the pixel value in the image by 255, so the pixel value will be converted to values of 0 and 1 [46]. Fig. 6 show the results of the data labeling.

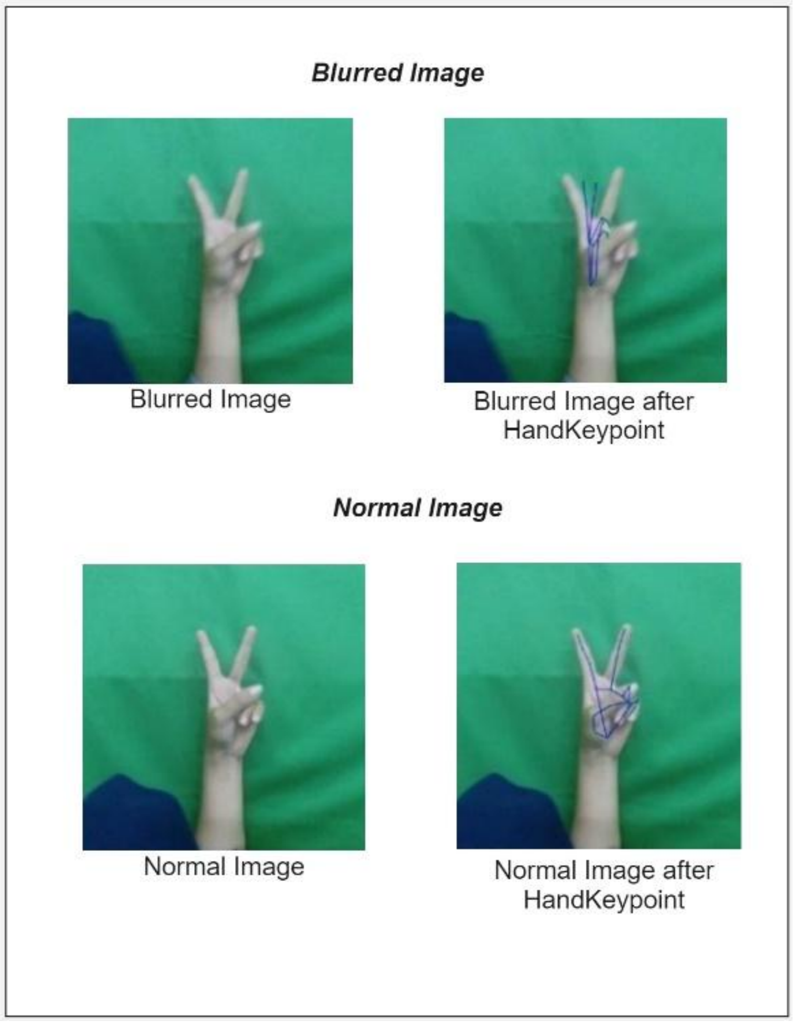
Blurred Image
Blurred Image
Blurred Image after HandKeypoint
Normal Image
Normal Image
Normal Image after HandKeypoint

Fig. 3. Comparison of blurry image hand keypoint results with normal image hand keypoint results

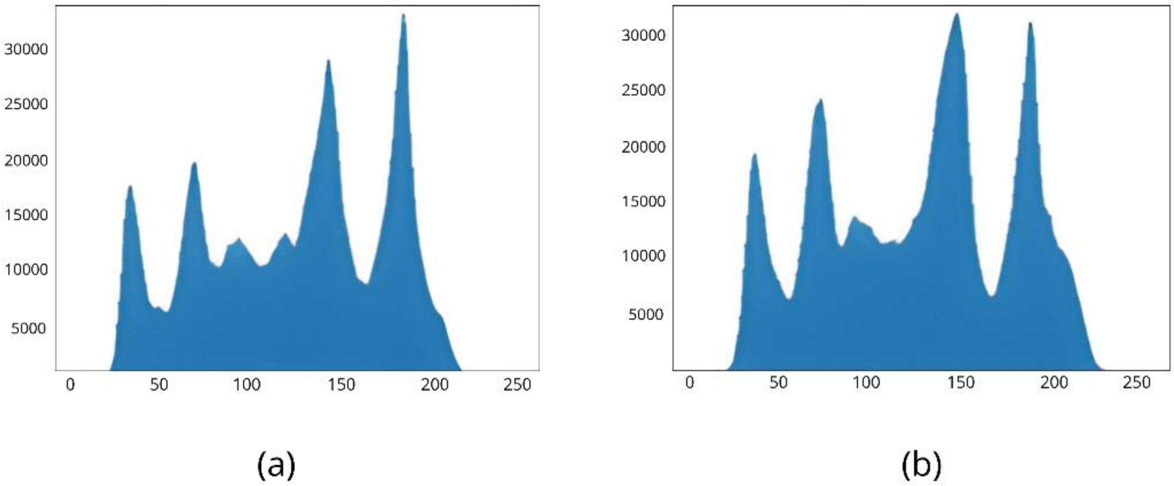

Fig. 4. Comparison of (a) Blurry image histogram with (b) Normal image histogram

This study's data preparation stage is paramount, as it guarantees the quality and precision of the data used to train the Convolutional Neural Network (CNN) model. The research team employed meticulous techniques, including manual key point detection, cropping, filtering, scaling, and data labelling, to obtain accurate and superior data. This method not only enhances the performance and efficiency of the model but also guarantees the proper functioning of the generated system under diverse real-world settings [47], [48]. The data pre-treatment processes established a foundation for subsequent modelling procedures.

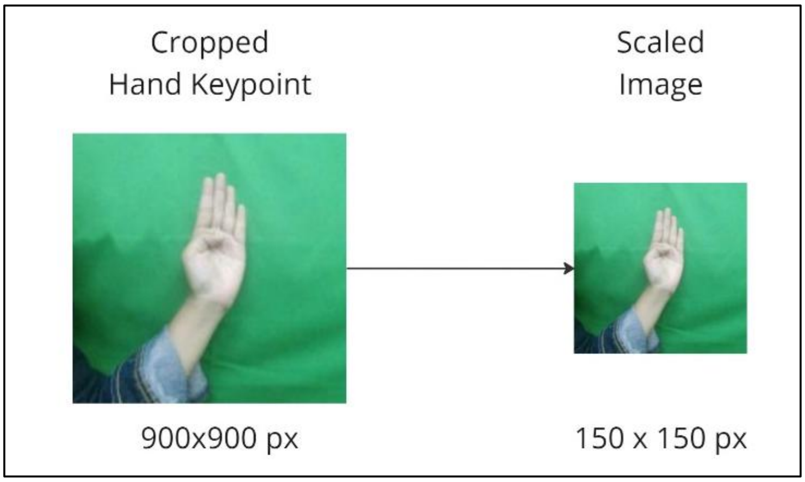

Fig. 5. Result of scaling process

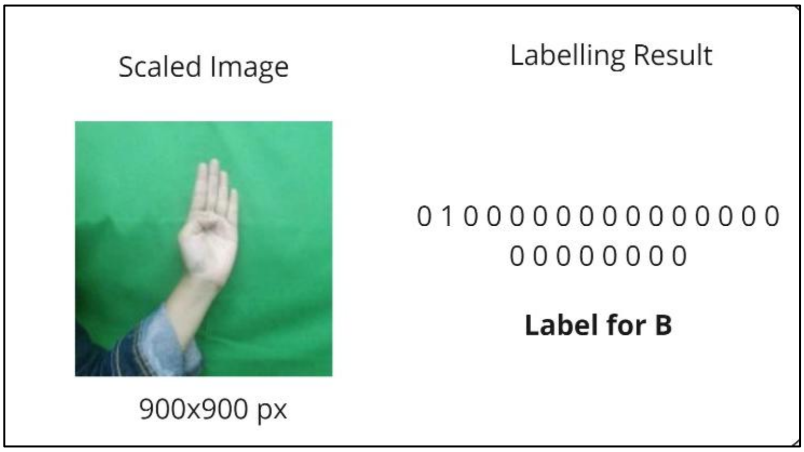

Fig. 6. Result of labelling process

2.3. Modelling with HandKeypointed-Convolutional Neural Network (HK-CNN)

After obtaining the hand keypoint detection results, the next step is to proceed with modeling using Convolutional Neural Networks (CNN). This stage aims to find the joints on the fingers and fingertips in the hand image. CNN is a technique derived from the Multilayer Perceptron (MLP) that is specifically designed to analyze picture data in a two-dimensional format. It draws inspiration from the neural networks found in the human brain [49]. CNNs are part of Deep Neural Networks and are widely used in image data processing because they have a high network depth [50]. CNN's main advantage lies in its ability to learn features from data in a hierarchical manner, ranging from simple features such as edges and textures to complex features such as specific objects or patterns. In this study, modeling was carried out using three well-known CNN architectures, namely ResNet-50, EfficientNet, and InceptionV3. These three architectures were chosen because they have their own advantages in terms of network depth, computing efficiency, and the ability to capture multi-scale features. ResNet-50 leverages residual blocks to overcome gradient vanishing issues, allowing the network to go deeper without loss of performance. EfficientNet uses a compound scaling approach, which is a combined scale of network depth, width, and resolution, making it highly computationally efficient without sacrificing accuracy. InceptionV3, on the other hand, relies on the inception module to capture multi-scale features through different kernel sizes in different convolution paths, while reducing the computational load with factorized convolutions. The modeling process begins with the collection of image datasets that are processed through the resizing stages according to architectural needs, data augmentation to improve generalization, and data division into training sets and test sets with several sharing scenarios such as 80:20, 70:30, and 20:80. In the training process, optimizers such as Adam are used with the loss categorical crossentropy function, and parameters such as the number of epochs, batch size, and learning rate are tested to find the optimal configuration. After training, the model is tested using a test set to evaluate its generalization ability.

The following pseudocode outlines the complete process, starting from data collection, preprocessing, classification using CNN, and finally, the evaluation of the classification results shown in Fig. 7:

```
// HK-CNN Algorithm
START
COLLECT data
STORE as Raw_Image_Dataset

    FOR each image IN Raw_Image_Dataset DO
      FIND center_of_hand_keypoint
      STORE as Palm_Image_with_HandKeypoint
      APPLY image_filtering TO Palm_Image_with_HandKeypoint
      STORE as Filtered_Palm_Image
      SCALE image TO smaller pixels
      STORE as Scaled_Image
      LABEL Scaled_Image
      STORE as Labeled_Image
    END FOR

      INITIALIZE HK-CNN model
      FOR each Labeled_Image DO
        CLASSIFY using HK-CNN
           Apply Architecture Resnet50
           Apply Architecture EfficientNet
          Apply Architecture InceptionV3
        STORE result as Classified_Image
      END FOR

      FOR each CNN architecture (ResNet, EfficientNet, InceptionV3)
      DO INITIALIZE model with corresponding architecture
      TRAIN model USING Training_Set
      TEST model USING Testing_Set
      STORE results as Model_Performance
      END FOR

INITIALIZE confusion_matrix
FOR each Classified_Image DO
  UPDATE confusion_matrix
END FOR

CALCULATE Accuracy, Precision, Recall, F1_Score FROM confusion_matrix
DISPLAY Accuracy, Precision, Recall, F1_Score

END
```

Fig. 7. Pseudocode of HK-CNN process

This pseudocode integrates the detailed steps of image processing and classification using CNN, emphasizing CNN's advantages over MLP in handling image data. It outlines the process from data collection through preprocessing, classification, and evaluation, providing a comprehensive guide to handling and analyzing hand image datasets. Process of CNN classification typically consists of two main stages: feature learning and classification. The CNN model takes an image input with dimensions of 150×150×3. Term "number three" in this context refers to an image that consists of three channels: red, green, and blue, which are popularly known as RGB. The input images will be processed at the feature learning stage, which involves convolution and pooling. Next, it will enter the fully connected layer process [51]. Feature learning is a method that automates converting an image into numerical features. Classification stage is a phase that processes the results of feature learning and transfers them to the classification process based on predetermined subclasses. The model's progression is illustrated in Fig. 8.

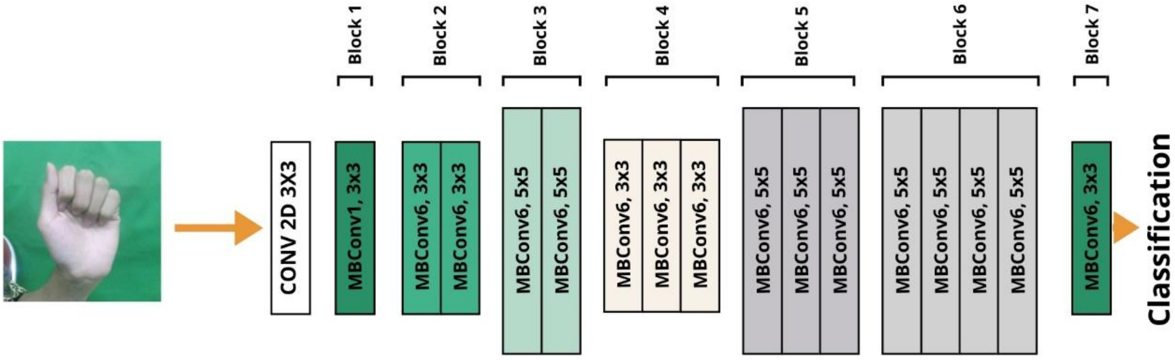

Fig. 8. Architecture model of HK-CNN with EfficientNet

The architecture in Fig. 8 is part of EfficientNet, which uses Mobile Bottleneck Convolution (MBConv) as the main block to process and classify images efficiently. The architecture starts with an initial convolution layer (CONV 2D 3×3), which is in charge of extracting the basic features from the input image using a two-dimensional convolution operation, which can be formulated as:

$$G[x,y] = \sum_{i=0}^{w-1}\sum_{j=0}^{h-1} F[x+i,y+j]\ .\ H[i,j] \tag{2}$$

Where $F[x,y]$ is a feature map input, $H[i,j]$ is a kernel of 3×3 , and $G[x,y]$ is the output feature map after the convolution operation. After that, the input is passed to a series of MBConv blocks consisting of three main stages: expanding convolution, which expands the feature dimension by doubling the channel using 1×1 depthwise convolution, which convolutes each channel separately using a 3×3 or 5×5 kernel to capture broader spatial relationships with high efficiency; and the projection layer, which reprojects the feature dimensions to be smaller to create a bottleneck, thereby reducing the number of parameters.

In some cases, skip connections are used to add information from the input directly to the output when the input and output dimensions are the same. The overall structure of the architecture consists of seven blocks, where the initial block (Block 1-3) captures basic features with a 3×3 kernel, the middle block (Block 4-6) uses the 5×5 kernel to understand more complex features, and the final block (Block 7) prepares the data for classification. The output from the last block is passed to the fully connected layer for classification. Architecture model of HK-CNN with Resnet50 shown in Fig. 9.

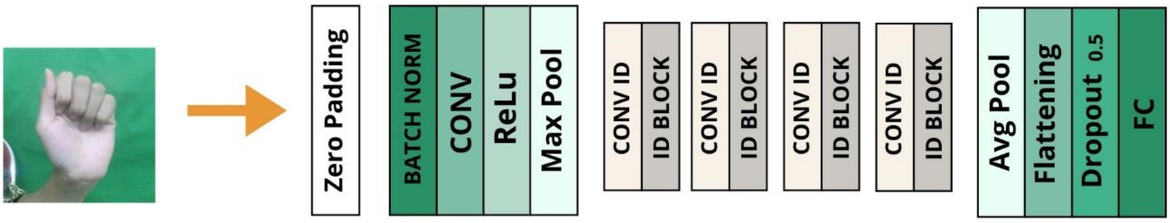

Fig. 9. Architecture model of HK-CNN with Resnet50

The ResNet-50 architecture uses bottleneck blocks, which introduces a more efficient approach by utilizing a combination of 1×1, 3×3, and 1×1 convolution layers [52], [53]. This structure allows for dimensionality reduction, feature processing, and then dimensionality increase, thus reducing computational complexity while maintaining high model performance. Mathematically, the residual block in ResNet-50 can be described by the following equation [54], [55].

$$y = F(x,\{Wi\}) + x \tag{3}$$

Where:

y = output of the residual block

$F(x, \{Wi\})$ = residual function

x = input layer

The main advantage of ResNet-50 is its ability to achieve high performance in image classification with fewer parameters than previous similar models [56], [57]. Architecture model of InceptionV3 shown in Fig. 10.

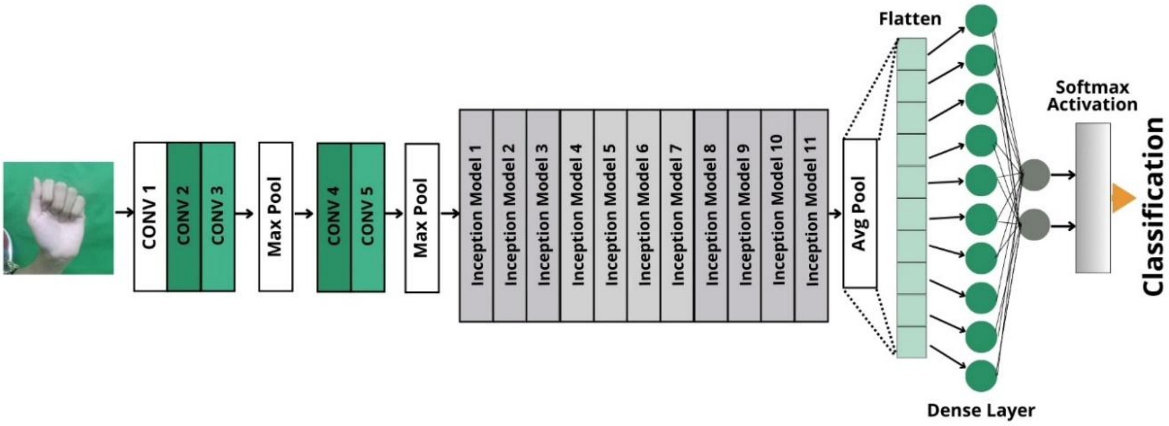

Fig. 10. Architecture model of InceptionV3

CNN has several model parameters that help determine the best parameters [58]. Therefore, searching for the best values for these parameters is necessary. This work aims to determine the optimal model parameters by examining the impact of several factors, such as the number of convolutional layers, the number of epochs, the batch size, the learning rate values, and the input image size. Several scenarios, such as Table 1, are from some of these parameters. Several scenarios for modelling shown in Fig. 11.

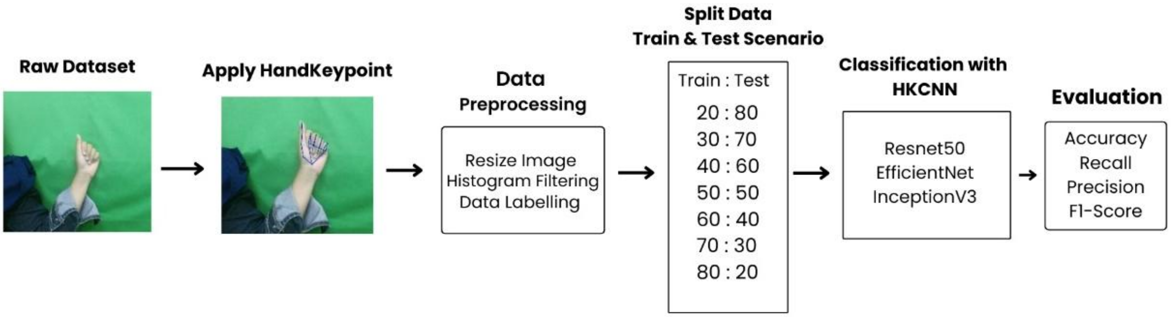

Fig. 11. Several scenarios for modelling

After conducting experiments in several scenarios, evaluations were carried out to measure the effectiveness of each scenario. The method used to evaluate the HK-CNN model is to use a confusion matrix. Confusion matrix is a valuable tool for evaluating the performance of classifiers, including those used for sign language classification. In sign language, a confusion matrix helps understand a classifier's accuracy and errors by displaying the predicted versus actual classes. We can calculate various evaluation metrics that provide deeper insights into the model's performance on sign language classification systems.

3. Results and Discussion

After modeling, implementation and testing were carried out by trying the performance of the Convolutional Neural Network model to classify sign language. The research includes various split data scenarios, such as 80:20, 70:30, and 20:80 ratios, which reflect the proportion of training and testing data. The results show that the performance of the CNN architecture varies depending on the split data scenario and the type of architecture used. Table 2 explains the results of HK-CNN in each scenario and its architecture.

Table 2. HK-CNN results in each scenario and architecture

Scenario	HK-CNN Architecture	Accuracy	Precision	Recall	F1-Score	Computation Time
80:20	EfficientNet	0.990	0.996	0.964	0.980	2 m 44.8 s
70:30	EfficientNet	0.987	**0.997**	0.967	0.981	2 m 22.6 s
60:40	EfficientNet	**0.991**	0.993	**0.979**	**0.986**	2 m 22.7 s
50:50	EfficientNet	0.985	0.995	0.970	0.982	2 m 30.7 s
40:60	EfficientNet	0.979	0.991	0.934	0.961	2 m 15.3 s
30:70	EfficientNet	0.979	0.985	0.931	0.952	2 m 37.1 s
20:80	EfficientNet	0.983	0.993	0.917	0.953	2 m 27 s
80:20	Resnet-50	0.689	0.886	0.385	0.534	5 m 19.6 s
70:30	Resnet-50	0.960	0.986	0.914	0.948	5 m 22 s
60:40	Resnet-50	0.954	0.988	0.892	0.937	4 m 33 s
50:50	Resnet-50	0.938	0.975	0.868	0.917	4 m 37.9 s
40:60	Resnet-50	0.974	0.989	**0.920**	0.953	6 m 11.6 s
30:70	Resnet-50	0.968	0.990	0.888	0.935	5 m 20.4 s
20:80	Resnet-50	**0.993**	**0.992**	0.913	**0.954**	5 m 27.1 s
80:20	InceptionV3	0.878	0.985	0.633	0.768	7 m 6 s
70:30	InceptionV3	0.872	0.980	0.602	0.742	7 m 16.9 s
60:40	InceptionV3	0.904	0.983	**0.710**	**0.821**	7 m 34.4 s
50:50	InceptionV3	0.899	0.990	0.558	0.712	9 m 18.4 s
40:60	InceptionV3	0.899	0.991	0.510	0.668	9 m 5.4 s
30:70	InceptionV3	0.921	0.990	0.504	0.665	8 m 10.1 s
20:80	InceptionV3	**0.934**	**1.000**	0.569	0.722	8 m 54.7 s

In the results in Table 2, EfficientNet provides the best performance with high average accuracy in almost all scenarios. The maximum accuracy reached 99.1% in the 60:40 data split, indicating that the model is able to recognize patterns very effectively, even when the amount of training data is not dominant. The consistency of the model is also evident from the highest Precision value of 99.7% and Recall of 97.9%, indicating the model's ability to provide accurate and relevant results in sign language pattern recognition. In addition, EfficientNet's average computation time of only 2 minutes 30 seconds per experiment demonstrates its efficiency, making it suitable for real-time applications such as automatic sign language interpreters in education and public services.

Meanwhile, ResNet-50, which is known for its residual block approach to overcome the vanishing gradient problem, recorded a maximum accuracy of 99.3% in the 20:80 data sharing scenario. While ResNet-50's accuracy results were competitive, its average computation time reached 5 minutes per experiment, much longer than EfficientNet. ResNet-50 showed superiority in recognizing complex patterns, especially on more limited training datasets. Its performance remained stable on the Recall and F1-Score metrics, although the Precision value was slightly lower than EfficientNet. These advantages make ResNet-50 more suitable for system development facing small but highly complex datasets.

On the other hand, the InceptionV3 architecture recorded a maximum accuracy of 93.4% in the 20:80 data split scenario, which was the lowest result among the three architectures tested. The main advantage of InceptionV3 is its ability to capture multi-scale features through the inception module, but its efficiency is limited as the average computation time reaches 7 minutes per experiment. Despite its ability to recognize basic patterns, InceptionV3 is more suitable for non-real-time applications or exploratory research that does not require a quick response.

Based on the method comparison Table 3, the Hand Keypointed CNN (HK-CNN) approach was shown to yield 99.3% accuracy, much higher than other methods such as CNN with AlexNet (89.04%) and VGG-16 (84.96%). Compared to other CNN-based methods, such as Hand Gesture CNN (85.3%), HK-CNN showed significant improvements in efficiency and accuracy. By utilizing hand keypoints detection, this method not only reduces the training data requirement but also maintains high accuracy. This allows the system to recognize patterns effectively and consistently.

Table 3. Method Comparison Results

Methods	Accuracy
HandKeypointed-CNN (HK-CNN)	99.30 %
CNN + HandGesture [19]	85.30 %
CNN with AlexNet [59]	89.04 %
CNN with VGG-16 [59]	84.96 %

While results show an accuracy of 99.3%, several factors need to be considered, such as the impact of real-world conditions. The model performs well in controlled testing, such as lighting and background, but in real-world scenarios, these factors may affect the model's performance [60], [61]. Therefore, testing in more varied conditions is necessary to ensure consistent results. Additionally, while the system reduces the need for training data through hand keypoint detection, efficiency in terms of training time and resource consumption must also be addressed. Optimizing the model through techniques like transfer learning could speed up the training process and improve performance [62], [63]. Another challenge is the diversity in individual sign language styles, where the model must recognize variations in hand gestures while maintaining accuracy. Ethically, using this technology also requires attention to data privacy, particularly regarding the collection, storage, and protection of hand-image data. Clear privacy policies will help increase public acceptance of this technology. Furthermore, implementing this technology requires user training and raising social awareness about the importance of sign language as an inclusive communication tool.

The results of this research show great potential in real-world applications, such as the development of an automatic sign language interpreter to improve the communication of deaf people in Indonesia. The system can be utilized in education to support interaction between teachers and deaf students, as well as in public services to reduce communication barriers. With a CNN-based approach that is continuously improved through the HK-CNN method, this research successfully presents a more comprehensive and inclusive system in classifying SIBI. Through better accuracy and precision, this research seeks to improve interactions between the deaf community and the general public, while promoting inclusiveness in daily life. The findings are expected to contribute to the development of more effective sign language translator systems, thus supporting better communication and integration for deaf individuals.

4. Conclusion

The results of this study indicate that the Hand Keypointed Convolutional Neural Network (HK-CNN) method is an effective and efficient approach for classifying the Indonesian Sign Language System (SIBI). By utilizing key points on the hand as the main feature, this method can reduce the need for large training data without compromising accuracy. This makes HK-CNN a more resource-efficient and viable solution for real-world sign language recognition systems. The EfficientNet architecture demonstrated the best performance compared to other architectures, with a maximum accuracy of 99.1% in the 60:40 data split scenario. In addition to high accuracy, this model also showed computational efficiency, with an average processing time of only 2 minutes and 30 seconds per experiment, as well as consistency in Precision (99.7%) and Recall (97.9%) values. These advantages make EfficientNet a prime candidate for real-time applications, such as automatic sign language interpreters that can be used in the education sector, public services, and work environments. The ResNet-50 architecture also showed competitive results with a maximum accuracy of 99.3% in the 20:80 data split scenario. Its ability to recognize complex patterns makes it suitable for use on smaller but more varied datasets. However, the longer computation time, averaging 5 minutes per experiment, poses a major challenge for real-time applications. On the other hand, the InceptionV3 architecture achieved a maximum accuracy of 93.4%, but its efficiency was lower than the other two architectures, with an average computation time of 7 minutes per experiment. This model is more suitable for exploratory research or non-real-time applications that do not require a quick response.

Compared to other methods such as CNN with AlexNet (89.04% accuracy) and VGG-16 (84.96% accuracy), the HK-CNN approach provides a significant accuracy improvement of up to 98.4%. Furthermore, compared to other CNN-based methods such as Hand Gesture CNN (85.3% accuracy), HK-CNN exhibits better capability in recognizing complex visual patterns. By utilizing key-point detection, this method not only provides high accuracy but also allows the system to adapt to various hand gesture styles from different individuals. This research demonstrates significant potential in developing an automatic sign language interpreter that can enhance communication for the deaf community in Indonesia. The system can be used to improve interactions between teachers and deaf students in the education sector or to reduce communication barriers in public services. However, it is important to acknowledge certain limitations of this study. The participant group consisted of 12 models from a single university, which may not fully capture the diversity of the deaf community in Indonesia. Therefore, further research with a larger and more diverse sample is necessary to improve the generalizability of the findings. Future studies should also explore the integration of the HK-CNN method with real-time gesture recognition systems, addressing challenges such as diverse sign language variations and environmental conditions. Additionally, incorporating temporal-based algorithms for dynamic gesture recognition would enhance the system’s ability to understand continuous sign language sequences, making it more versatile for practical, real-world communication.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] K. Emmorey, “Ten Things You Should Know About Sign Languages,” *Current Directions in Psychological Science*, vol. 32, no. 5, pp. 387–394, 2023, https://doi.org/10.1177/09637214231173071.

[2] R. Rastgoo, K. Kiani, S. Escalera, V. Athitsos, and M. Sabokrou, “A survey on recent advances in Sign Language Production,” *Expert Systems with Applications*, vol. 243, p. 122846, 2024, https://doi.org/10.1016/j.eswa.2023.122846.

[3] K. K. Sukhadan, V. D. Bakhade, G. S. Thakare, K. G. Dhanbhar, and A. C. Deshmukh, “Sign Language Recognition System,” *International Journal for Research in Applied Science & Engineering Technology*, vol. 12, no. 3, pp. 140–143, 2024, https://doi.org/10.22214/ijraset.2024.58758.

[4] D. Lillo-Martin and J. A. Hochgesang, “Signed languages – Unique and ordinary: A commentary on Kidd and Garcia (2022),” *First Language*, vol. 42, no. 6, pp. 789–793, 2022, https://doi.org/10.1177/01427237221098858.

[5] D. Novaliendry, M. F. P. Pratama, K. Budayawan, Y. Huda, and W. M. Y. Rahiman, “Design and Development of Sign Language Learning Application for Special Needs Students Based on Android Using Flutter,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 16, pp. 76–92, 2023, https://doi.org/10.3991/ijoe.v19i16.44669.

[6] A. Taupiq, M. Wildan Fajri, and Dannylee, “Identification of Indonesian Sign Language System Using Deep Learning in Yolo-based,” *Media Journal of General Computer Science*, vol. 1, no. 2, pp. 40–47, 2024, https://doi.org/10.62205/mjgcs.v1i2.22.

[7] M. Marlina, A. Mahdi, and Y. Karneli, “The effectiveness of the Bisindo-based rational emotive behavior therapy model in reducing social anxiety in deaf women victims of sexual harassment,” *The Journal of Adult Protection*, vol. 25, no. 4, pp. 199–214, 2023, https://doi.org/10.1108/JAP-10-2022-0024.

[8] M. Marlina, Y. T. Ningsih, Z. Fikry, and D. R. Fransiska, “Bisindo-based rational emotive behaviour therapy model: study preliminary prevention of sexual harassment in women with deafness,” *The Journal of Adult Protection*, vol. 24, no. 2, pp. 102–114, 2022, https://doi.org/10.1108/JAP-09-2021-0032.

[9] N. K. I. Wahyuni, I. M. Suarjana, and D. A. P. Handayani, “Development Of Video Learning Based On The Indonesian Language Signing System (SIBI) Method For Class II Deaf Chill At SDN 2 Bengkala Academic Year 2022/2023,” *Journal of Psychology and Instruction*, vol. 5, no. 3, 2023, https://doi.org/10.23887/jpai.v5i3.64832.

[10] R. Rastgoo, K. Kiani, and S. Escalera, “Sign Language Recognition: A Deep Survey,” *Expert Systems with Applications*, vol. 164, p. 113794, 2021, https://doi.org/10.1016/j.eswa.2020.113794.

[11] D. Straupeniece, D. Bethere, E. Ozola, “Sign Language of the Deaf People: A Study on Public Understanding,” *Education. Innovation. Diversity*, vol. 2, no. 7, pp. 109–114, 2024, https://doi.org/10.17770/eid2023.2.7356.

[12] S. Arooj, S. Altaf, S. Ahmad, H. Mahmoud, and A. S. N. Mohamed, “Enhancing sign language recognition using CNN and SIFT: A case study on Pakistan sign language,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 2, p. 101934, 2024, https://doi.org/10.1016/j.jksuci.2024.101934.

[13] N. K. Jyothi, V. Harshitha, M. Puneira, P. S. Nikitha, “Image Processing Model for Sign Language Recognition System,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 12, no. 5, pp. 2235–2237, 2024, https://doi.org/10.22214/ijraset.2024.61671.

[14] S. Renjith and R. Manazhy, “Sign language: a systematic review on classification and recognition,” *Multimedia Tools and Applications*, vol. 83, pp. 77077–77127, 2024, https://doi.org/10.1007/s11042-024-18583-4.

[15] M. Alaftekin, I. Pacal, and K. Cicek, “Real-time sign language recognition based on YOLO algorithm,” *Neural Computing and Applications*, vol. 36, pp. 7609–7624, 2024, https://doi.org/10.1007/s00521-024-09503-6.

[16] E. Aldhahri *et al.*, “Arabic Sign Language Recognition Using Convolutional Neural Network and MobileNet,” *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2147–2154, 2023, https://doi.org/10.1007/s13369-022-07144-2.

[17] S. Dwijayanti, S. I. Taqiyyah, H. Hikmarika, and B. Y. Suprapto, “Indonesia Sign Language Recognition using Convolutional Neural Network,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021, https://dx.doi.org/10.14569/IJACSA.2021.0121046.

[18] A. Osman Hashi, S. Zaiton Mohd Hashim and A. Bte Asamah, "A Systematic Review of Hand Gesture Recognition: An Update From 2018 to 2024," *IEEE Access*, vol. 12, pp. 143599-143626, 2024, https://doi.org/10.1109/ACCESS.2024.3421992.

[19] D. Pribadi, M. Wahyudi, D. Puspitasari, A. Wibowo, R. Saputra, and R. Saefurrohman, “Real Time Indonesian Sign Language Hand Gesture Phonology Translation Using Deep Learning Model,” *Scitepress*, vol. 1, pp. 172–176, 2024, https://doi.org/10.5220/0012446000003848.

[20] G. Chaganava and D. Kakulia, “Keypoint Detector Retraining Techniques for the Communication System of Sign Language Speakers,” *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*, vol. 21, pp. 74–86, 2020, https://doi.org/10.18038/estubtda.822295.

[21] G. Kim, J. Cho, G. Kim, B. Kim, and K. Jeon, “A Keypoint-based Sign Language Start and End Point Detection Scheme,” *KIISE Transactions on Computing Practices*, vol. 29, no. 4, pp. 184–189, 2023, https://doi.org/10.5626/KTCP.2023.29.4.184.

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

[23] T. G. K and M. N. Nachappa, “Sign Language Recognition by Image Processing,” *International Journal of Advanced Research in Science, Communication and Technology*, vol. 4, no. 4, pp. 306–310, 2024, https://doi.org/10.48175/IJARSCT-15954.

[24] S. Mittal, S. Srivastava and J. P. Jayanth, "A Survey of Deep Learning Techniques for Underwater Image Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 6968-6982, 2023, https://doi.org/10.1109/TNNLS.2022.3143887.

[25] A. Sharma, N. Sharma, Y. Saxena, A. Singh, and D. Sadhya, “Benchmarking deep neural network approaches for Indian Sign Language recognition,” *Neural Computing and Applications*, vol. 33, no. 12, pp. 6685–6696, 2021, https://doi.org/10.1007/s00521-020-05448-8.

[26] M. Momeny *et al*., “Learning-to-augment strategy using noisy and denoised data: Improving generalizability of deep CNN for the detection of COVID-19 in X-ray images,” *Computers in Biology and Medicine*, vol. 136, p. 104704, 2021, https://doi.org/10.1016/j.compbiomed.2021.104704.

[27] J. Hai *et al*., “R2RNet: Low-light image enhancement via Real-low to Real-normal Network,” *Journal of Visual Communication and Image Representation*, vol. 90, p. 103712, 2023, https://doi.org/10.1016/j.jvcir.2022.103712.

[28] A. Wadhawan, P. Kumar, “Sign language recognition systems: A decade systematic literature review,” *Archives of computational methods in engineering*, vol. 28, pp. 785-813, 2021, https://doi.org/10.1007/s11831-019-09384-2.

[29] R. Sutjiadi, “Android-Based Application for Real-Time Indonesian Sign Language Recognition Using Convolutional Neural Network,” *TEM Journal*, vol. 12, no. 3, pp. 1541–1549, 2023, https://doi.org/10.18421/TEM123-35.

[30] Y. Obi, K. S. Claudio, V. M. Budiman, S. Achmad, and A. Kurniawan, “Sign language recognition system for communicating to people with disabilities,” *Procedia Computer Science*, vol. 216, pp. 13–20, 2022, https://doi.org/10.1016/j.procs.2022.12.106.

[31] U. Özsoy, Y. Yıldırım, S. Karaşin, R. Şekerci, and L. B. Süzen, “Reliability and agreement of Azure Kinect and Kinect v2 depth sensors in the shoulder joint range of motion estimation,” *Journal of Shoulder and Elbow Surgery*, vol. 31, no. 10, pp. 2049–2056, 2022, https://doi.org/10.1016/j.jse.2022.04.007.

[32] L.-F. Yeung, Z. Yang, K. C.-C. Cheng, D. Du, and R. K.-Y. Tong, “Effects of camera viewing angles on tracking kinematic gait patterns using Azure Kinect, Kinect v2 and Orbbec Astra Pro v2,” *Gait & Posture*, vol. 87, pp. 19–26, 2021, https://doi.org/10.1016/j.gaitpost.2021.04.005.

[33] C. Posner, A. Sánchez-Mompó, I. Mavromatis, and M. Al-Ani, “A dataset of human body tracking of walking actions captured using two Azure Kinect sensors,” *Data in Brief*, vol. 49, p. 109334, 2023, https://doi.org/10.1016/j.dib.2023.109334.

[34] C. Neupane, A. Koirala, Z. Wang, and K. B. Walsh, “Evaluation of Depth Cameras for Use in Fruit Localization and Sizing: Finding a Successor to Kinect v2,” *Agronomy*, vol. 11, no. 9, p. 1780, 2021, https://doi.org/10.3390/agronomy11091780.

[35] I. D. M. B. A. Darmawan, Linawati, G. Sukadarmika, N. M. A. E. D. Wirastuti, and R. Pulungan, “Temporal Action Segmentation in Sign Language System for Bahasa Indonesia (SIBI) Videos Using Optical Flow-Based Approach,” *Jurnal Ilmu Komputer dan Informasi*, vol. 17, no. 2, pp. 195–202, 2024, https://doi.org/10.21609/jiki.v17i2.1284.

[36] A. Albar, H. Hendrick, and R. Hidayat, “Segmentation Method for Face Modelling in Thermal Images,” *Knowledge Engineering and Data Science*, vol. 3, no. 2, pp. 99-105, 2020, http://dx.doi.org/10.17977/um018v3i22020p99-105.

[37] M. A. Ridwan and H. Mubarok, “The Recognition of American Sign Language Using CNN with Hand Keypoint,” *International Journal on Information and Communication Technology*, vol. 9, no. 2, pp. 86–95, 2023, https://socjs.telkomuniversity.ac.id/ojs/index.php/ijoict/article/view/845.

[38] E. Rakun and N. F. Setyono, “Improving Recognition of SIBI Gesture by Combining Skeleton and Hand Shape Features,” *Jurnal Ilmu Komputer dan Informasi*, vol. 15, no. 2, pp. 69–79, 2022, https://doi.org/10.21609/jiki.v15i2.1014.

[39] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, “Machine learning methods for sign language recognition: A critical review and analysis,” *Intelligent Systems with Applications*, vol. 12, p. 200056, 2021, https://doi.org/10.1016/j.iswa.2021.200056.

[40] T. Rabie, M. Baziyad, R. Sani, T. Bonny, and R. Fareh, “Color Histogram Contouring: A New Training-Less Approach to Object Detection,” *Electronics*, vol. 13, no. 13, p. 2522, 2024, https://doi.org/10.3390/electronics13132522.

[41] H. Ohno, “One-shot reflectance direction color mapping for identifying surface roughness,” *Precision Engineering*, vol. 85, pp. 65–71, 2024, https://doi.org/10.1016/j.precisioneng.2023.09.004.

[42] A. Shah, N. Azam, E. Alanazi, and J. Yao, “Image blurring and sharpening inspired three-way clustering approach,” *Applied Intelligence*, vol. 52, no. 15, pp. 18131–18155, 2022, https://doi.org/10.1007/s10489-021-03072-0.

[43] T. Wu, J. Shao, X. Gu, M. K. Ng, and T. Zeng, “Two-stage image segmentation based on nonconvex approximation and thresholding,” *Applied Mathematics and Computation*, vol. 403, p. 126168, 2021, https://doi.org/10.1016/j.amc.2021.126168.

[44] M. M. Tall, I. Ngom, O. Sadio, A. Coulibaly, I. Diagne, and M. Ndiaye, “Automatic detection and counting of fisheries using fish images,” *Bulletin of Social Informatics Theory and Application*, vol. 7, no. 2, pp. 150–162, 2023, https://doi.org/10.31763/businta.v7i2.655.

[45] J. Zhang, X. Bu, Y. Wang, H. Dong, Y. Zhang, and H. Wu, “Sign language recognition based on dual-path background erasure convolutional neural network,” *Scientific Reports*, vol. 14, no. 1, p. 11360, 2024, https://doi.org/10.1038/s41598-024-62008-z.

[46] R. Gupta and A. Kumar, “Indian sign language recognition using wearable sensors and multi-label classification,” *Computers & Electrical Engineering*, vol. 90, p. 106898, 2021, https://doi.org/10.1016/j.compeleceng.2020.106898.

[47] A. P. Wibawa *et al.*, “Frontier Energy System and Power Engineering Forecasting Hourly Energy Fluctuations Using Recurrent Neural Network (RNN),” *Frontier Energy System and Power Engineering*, vol. 5, no. 2, pp. 50–57, 2023, http://dx.doi.org/10.17977/um049v5i2p50-57.

[48] L. Latumakulita *et al.*, “Web-Based System for Medicinal Plants Identification Using Convolutional Neural Network,” *Bulletin of Social Informatics Theory and Application*, vol. 6, no. 2, pp. 158–167, 2022, https://doi.org/10.31763/businta.v6i2.601.

[49] M. C. Bagaskoro, F. Prasojo, A. N. Handayani, E. Hitipeuw, A. P. Wibawa, and Y. W. Liang, “Hand image reading approach method to Indonesian Language Signing System (SIBI) using neural network and multi layer perseptron,” *Science in Information Technology Letters*, vol. 4, no. 2, pp. 97–108, 2023, https://doi.org/10.31763/sitech.v4i2.1362.

[50] L. Zhou, X. Ma, X. Wang, S. Hao, Y. Ye, and K. Zhao, “Shallow-to-Deep Spatial–Spectral Feature Enhancement for Hyperspectral Image Classification,” *Remote Sensing*, vol. 15, no. 1, p. 261, 2023, https://doi.org/10.3390/rs15010261.

[51] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 6, p. 420, 2021, https://doi.org/10.1007/s42979-021-00815-1.

[52] K. Narang, M. Gupta, R. Kumar and A. J. Obaid, "Channel Attention Based on ResNet-50 Model for Image Classification of DFUs Using CNN," *2024 5th International Conference for Emerging Technology (INCET)*, pp. 1-6, 2024, https://doi.org/10.1109/INCET61516.2024.10593169.

[53] J. Liu, “Face recognition technology based on ResNet-50,” *Applied and Computational Engineering*, vol. 39, no. 1, pp. 160–165, 2024, https://doi.org/10.54254/2755-2721/39/20230593.

[54] B. Deng *et al.*, “Application of RESNET50 Convolution Neural Network for the Extraction of Optical Parameters in Scattering Media,” *arXiv*, 2024, https://doi.org/10.48550/arXiv.2404.16647.

[55] M. Yin, X. Li, Y. Zhang, S. Wang, “On the Mathematical Understanding of ResNet with Feynman Path Integral,” *arXiv*, 2019, https://doi.org/10.48550/arXiv.1904.07568.

[56] A. Kumar, L. Nelson and S. Singh, "ResNet-50 Transfer Learning Model for Diabetic Foot Ulcer Detection Using Thermal Images," *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, pp. 1-5, 2023, https://doi.org/10.1109/INCOFT60753.2023.10425447.

[57] L. M. K. Sheikh, A. Shaikh, A. Sandupatla, R. Pudale, A. Bakare, M. Chavan, “Classification of Simple CNN Model and ResNet50,” *International Journal for Research in Applied Science & Engineering Technology*, vol. 12, no. 4, pp. 4606-4610, 2024, https://doi.org/10.22214/ijraset.2024.60677.

[58] D. F. Laistulloh, A. N. Handayani, R. A. Asmara, and P. Taw, “Convolutional Neural Network in Motion Detection for Physiotherapy Exercise Movement,” *Knowledge Engineering and Data Science*, vol. 7, no. 1, pp. 27-39, 2024, http://dx.doi.org/10.17977/um018v7i12024p27-39.

[59] M. Dolla Meitantya, C. Atika Sari, E. Hari Rachmawanto, and R. Raad Ali, "VGG-16 Architecture on CNN for American Sign Language Classification," *Jurnal Teknik Informatika*, vol. 5, no. 4, pp. 1165–1171, 2024, https://doi.org/10.52436/1.jutif.2024.5.4.2160.

[60] S. Sony, K. Dunphy, A. Sadhu, and M. Capretz, "A systematic review of convolutional neural network-based structural condition assessment techniques," *Engineering Structures*, vol. 226, p. 111347, 2021, https://doi.org/10.1016/j.engstruct.2020.111347.

[61] S. Das, Md. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier," *Expert Systems with Applications*, vol. 213, p. 118914, 2023, https://doi.org/10.1016/j.eswa.2022.118914.

[62] A. W. Salehi *et al*., "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustainability*, vol. 15, no. 7, p. 5930, 2023, https://doi.org/10.3390/su15075930.

[63] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies*, vol. 11, no. 2, p. 40, 2023, https://doi.org/10.3390/technologies11020040.