REKAYASA 20 (2) (2022): 108-117

REKAYASA

Jurnal Penerapan Teknologi dan Pembelajaran

http://journal.unnes.ac.id/journals/rekayasa

Sign Language Detection Application to Facilitate Communication for Speech and Hearing Impaired Individuals Based on Computer Vision Technology Using Inception Resnetv2

Arief Arfriandi[1*], Yoga Agung Prabowo[2], Moh. Dafi Najuda[2], Tri Anggi Ratna Puspita[2], Shakira Wahyu Virnanda[3], Said Sunardiyo[3]

[1]Program Studi Teknik Komputer, Universitas Negeri Semarang, Semarang 50229, Indonesia

[2]Program Studi Teknik Kimia, Universitas Negeri Semarang, Semarang 50229, Indonesia

[3]Program Studi Teknik Elektro, Universitas Negeri Semarang, Semarang 50229, Indonesia

*E-mail: arfriandi@mail.unnes.ac.id

DOI: https://doi.org/10.15294/rekayasa.v20i2.19433

Abstract

Communication is a fundamental human need, yet not everyone possesses perfect communication abilities. People Communication is a fundamental human need, yet individuals with speech and hearing impairments face challenges due to the limited understanding of sign language among the general public. This study applies Artificial Intelligence and Computer Vision to enhance communication accessibility by detecting hand gestures and converting them into text. The lack of real-time sign language translation remains a barrier for individuals with disabilities. Existing systems often struggle with accuracy and device compatibility. This research develops and evaluates HARDISC, an Android-based application that recognizes letters A–Z through hand movement detection using a camera. The goal is to provide an effective and inclusive communication tool for the speech and hearing impaired. HARDISC utilizes Transfer Learning with Inception ResNetV2 and VGG16 for gesture classification. Image processing enables the camera to detect and translate hand movements into text. Model evaluation was based on accuracy, loss values, and device compatibility. Results show Inception ResNetV2 achieved 98.98% accuracy with a 0.0417 loss value, while VGG16 recorded 99.40% accuracy with a 0.0146 loss value, demonstrating high performance. HARDISC is compatible with Android KitKat 4 to Android 12, ensuring accessibility. This application provides an innovative, real-time solution to bridge communication gaps for individuals with speech and hearing impairments, improving their interaction with the general public.

Keywords: Artificial Intelligence, Computer Vision, Disabilities, HARDISC, real-time

INTRODUCTION

Not all individuals possess perfect communication abilities, particularly those with speech and hearing impairments. More than 5% of the world's population—approximately 432 million adults and 34 million children—require assistive devices for hearing loss or deafness. According to the World Health Organization, by 2050, over 700 million people (one in ten) will experience hearing or speech impairments. This condition can result from congenital factors, genetic disorders, infections, birth complications, or prolonged exposure to loud noises (Renauld & Basch, 2021). Fortunately, assistive technologies such as hearing aids, cochlear implants, and sign-language recognition systems can enhance communication for individuals with hearing impairments (Frush Holt, 2019).

Despite these advancements, speech and hearing-impaired individuals still face challenges in daily communication, mainly because the general public has a limited understanding of sign language (Fajri & Kusumastuti, 2019). This highlights the urgent need for a technological solution that bridges the communication gap.

Computer vision-based hand gesture recognition is a promising approach to assistive communication (Liu & Kehtarnavaz, 2016). Visual-based gesture recognition systems utilize computer vision techniques to capture and interpret sign language movements in real time (Ojeda-Castelo et al., 2022). Existing systems rely on cameras to translate gestures into readable text, enabling seamless interaction between deaf and non-sign language users.

Sign language users from diverse backgrounds naturally create shared communication spaces without a common language, forming the basis of multilingual-multimodal interactions (Zeshan, 2015). The HARDISC application aligns with this concept by employing computer vision technology with Inception ResNetV2 to recognize and translate hand gestures into text. This technology enables real-time, AI-powered sign language detection, improving accessibility for speech and hearing-impaired individuals.

Given the visual and spatial complexity of sign language, its interpretation can be enhanced using Natural Language Processing (NLP) combined with machine learning and computer vision (Ananthanarayana et al., 2021). Additionally, neuroscientific insights help bridge the connection between sign language and phonetics, leading to more accurate AI-driven sign recognition (Papastratis et al., 2021). With the increasing global reliance on sign language and technological advancements, systematic mapping of sensors, algorithms, and emerging technologies is crucial for improving accessibility in healthcare and everyday communication.

This research builds upon previous studies that could only interpret one-handed gestures, enabling the recognition of two-handed sign language with immediate responsiveness. The HARDISC application is designed to convert sign language into readable text, significantly improving communication speed and accessibility for speech and hearing-impaired individuals. The application offers high-accuracy sign detection, real-time processing, and an interactive tutorial menu, ensuring ease of use. By leveraging advanced computer vision algorithms such as Inception ResNetV2, HARDISC aims to redefine accessibility and inclusivity in communication.

This research aligns with SDG 10 (Reduced Inequalities) by promoting inclusive communication for individuals with speech and

hearing impairments through AI-driven sign language detection. It also supports SDG 9 (Industry, Innovation, and Infrastructure) by leveraging computer vision technology to enhance accessibility and technological innovation.

METHOD

The development of the HARDISC application follows a structured process consisting of several stages, beginning with the design of a flowchart diagram, as illustrated in Figure 1. The process starts with problem analysis and solution identification.

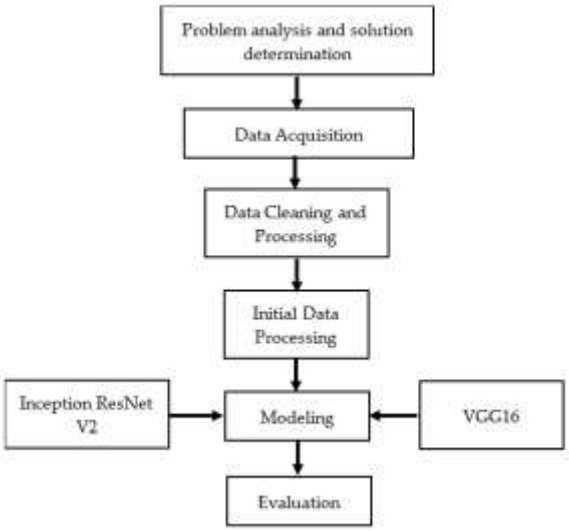

Figure 1. Flowchart for HARDISK Application Design

One of the most common difficulties for speech and hearing-impaired individuals is their ability to communicate with the general public, especially those who do not understand sign language (Razalli et al., 2019). This barrier limits social interactions and access to essential services.

To address this issue, the HARDISC application is designed for individuals with disabilities and the general public. The proposed solution involves a camera-based feature that detects hand gestures and translates them into alphabet letters. This functionality significantly enhances communication accessibility for speech and hearing-impaired individuals, allowing for a more inclusive interaction experience.

Data Cleaning and Acquisition

The dataset used in this study was obtained from Kaggle. The dataset consists of a total of 2,159 data points categorized into 27 classes, each representing different hand gestures corresponding to the letters A to Z. Each class contains 16 images in the test dataset and 64 images in the training dataset. The dataset captures hand gestures that depict various letters, as illustrated in Figure 2.

(a) (b)

Figure 2. Dataset Used
(a) letter E and (b) letter A

For VGG16, the transfer learning preparation involved adding a set of fully connected layers at the end of the VGG16 network. In the model architecture, additional fully connected layers were incorporated to create the base model for VGG16. The Flatten function was used to flatten the feature maps. A Dense layer (Layer 1) with 1,024 neurons was added, utilizing the ReLU activation function. A Dense layer (Layer 1) with 1,024 neurons was added, utilizing the ReLU activation function. A Dropout layer of 0.3 (30%) was included to prevent overfitting during training. The final layer consisted of a Dense layer with 27 neurons, corresponding to the number of classes, and employed a softmax activation function, which is suitable for multiclass classification.

The model was then compiled using categorical_crossentropy as the loss function,

RMSprop as the optimizer with a learning rate of 0.0001, and accuracy as the metric for evaluation. The final model architecture for InceptionResNetV2 is presented in Table 1, while the final model architecture for VGG16 is shown in Table 2. The model was designed and tested for execution.

Data Cleansing and Data Augmentation

The Data Cleansing process, also known as the preprocessing stage, involves loading the dataset and adjusting the image size before proceeding to the augmentation phase. During this stage, the dataset images are loaded, and the training and testing directories within the dataset document are defined, containing subdirectories for training (train) and testing (test) data. To ensure consistency, all images are resized to 256×256 pixels, with a batch size of 32 and a total of 25 epochs for training.

Additionally, a Confusion Matrix Graph Function is created to visually compare the classification results generated by the model with the actual classification outcomes. This function provides a clearer understanding of the model's performance and accuracy in distinguishing between different sign language gestures.

Data Augmentation

The augmentation process is carried out because the dataset we obtained is relatively small, which could negatively impact the model's accuracy (Maharana et al., 2022). To overcome this, augmentation is applied to maximize accuracy by expanding the dataset through random transformations. This ensures that the model does not see the same image twice in an identical form. This process helps prevent overfitting and enables the model to generalize better, ultimately leading to optimal accuracy.

The following augmentation techniques are applied to the dataset anticlockwise_rotation – Rotates images counterclockwise, clockwise_rotation – Rotates images clockwise, flip_up_down – Flips images vertically from top to bottom, sheared – Applies random shear transformation to images, blur – Adds a blur effect to images, wrap_shift – Applies a curved shift transformation to images, brightness – Adjusts the brightness of images within a range of 0.5 to 1.

After defining the augmentation functions, each transformation is applied to the image dataset. The augmented images are then stored in their respective dataset directories, such as Dataset-BISINDO/datatrain/A, Dataset-BISINDO/datatrain/B, Dataset-BISINDO/datatrain/C, and so on.

The greater the number of images augmented, the longer the code execution time. Once the additional images are generated, they are stored in a ZIP format for further processing.

Dataset Preparation and Model Development

In this stage of dataset preparation, the first step is defining the training and testing directories within the dataset, which contain subsets for training data (data train) and testing data (data test) Next, we ensure uniformity in image size, setting it to 256 pixels, with a batch size of 32 and a total of 25 epochs for training.

The model development in this study utilizes InceptionResNetV2 and VGG16 architectures. In the case of InceptionResNetV2, the preparation phase involves configuring the model for transfer learning by adding fully connected layers to the final section of the network, making it the base model. GlobalAveragePooling2D is employed for pooling operations, followed by a flattened layer for feature flattening. The first dense layer consists of 512 units with ReLU activation,

accompanied by a Dropout of 0.3 (30%) to prevent overfitting. Similarly, the second dense layer comprises 256 units with ReLU activation and an additional Dropout of 0.3 for further regularization. The final output layer is a Dense layer with 27 units, corresponding to the number of classes, and uses Softmax activation, which is well-suited for multiclass classification tasks. The model is then compiled using categorical cross-entropy as the loss function, RMSprop as the optimizer with a learning rate of 0.0001, and accuracy as the evaluation metric.

Model: "sequential 6"

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Functional)	(None, 6, 6, 1536)	54336736
Global_Average_Pool (GlobalAveragePooling2D)	(None, 1536)	0
flatten (Flatten)	(None, 1536)	0
dense_1 (Dense)	(None, 512)	786944
dropout_12 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_13 (Dropout)	(None, 256)	0
output (Dense)	(None, 27)	6939

Total params: 55,261,947
Trainable params: 925,211
Non-trainable params: 54,336,736

Model: "sequential 1"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 8, 8, 512)	14714688
flatten (Flatten)	(None, 32768)	0
dense_1 (Dense)	(None, 1024)	33555456
dropout_2 (Dropout)	(None, 1024)	0
output (Dense)	(None, 27)	27675

Total params: 48,297,819
Trainable params: 33,583,131
Non-trainable params: 14,714,688

Evaluation

In the evaluation stage, precision, recall, F1-score, and accuracy are measured, and a confusion matrix is applied to both models. Precision is defined as the ratio of relevant items selected to all selected items. The precision value can be calculated using Equation

1. Recall is determined as the ratio of relevant items selected to the total number of relevant items available, which can be computed using Equation 2. The F1-score represents the harmonic mean of precision and recall, and its value can be obtained using Equation 3.

Accuracy is defined as the percentage of correctly classified data records after the testing stage in the classification process. The accuracy value can be calculated using Equation 4, wherw TP is True Positive, FN is False Negative, FP is False Positive, TP is True Negative.

$$\text{Precision} = \frac{TP}{FP+TP} \text{x} 100\% \quad (1)$$

$$\text{Recall} = \frac{TP}{FN+TP} \text{x} 100\% \quad (2)$$

$$\text{F1} = 2\text{x}\frac{Precision\ x\ REcll}{Precision+Recall} \quad (3)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \text{x} 100\% \quad (4)$$

Meanwhile, the confusion matrix is the most widely used decision measurement method in supervised machine learning, which visualizes the level of confusion of the algorithm in each different class and does not depend on the classification algorithm.

Deployment

This stage aims to access and run models that have been created with Python, applications are created to facilitate the use of deep learning model results and are implemented in everyday life. This project uses Android Studio 2021, the library used is openCVLibrary3413, and the database used is MySQL Firebase. Android Studio is an official integrated development environment (IDE) tool resulting from a collaboration between Google and JetBrains which is specifically designed for developing Android applications

(Sibuea et al., 2022). This application is open source or free. The launch of Android Studio was announced by Google on May 16, 2013, at the Google I/O Conference event for 2013. Since then, Android Studio has replaced Eclipse as the official IDE for developing Android applications (Tran, 2021). Android Studio has complete components including a source code editor, compiler and debugger. This application can be used for a minimum of Android KitKat4 and a maximum of Android 12.

RESULT AND DISCUSSION

The performance of the accuracy and loss results of the Inception ResNetV2 architecture where the red line is for training data and the blue line is for validation data Shows graphs and tables. The results of the graphs and tables show that from epoch 1 to epoch 25 the accuracy results are increasing with an accuracy value of 98.98% and the loss results are getting lower with a loss value of 0.0417, so the model can be said to be working well. The results of the InceptionResNet V2 Modeling can be shown in Figure 3, and the translation of the InceptionResNet Training Results can be shown in Table 3.

The results of the VGG16 modelling, presented as training and validation accuracy graphs, are shown in Figure 4, while training and validation loss results are displayed. The corresponding training outcomes are detailed in Table 4. In the accuracy and loss graphs for the VGG16 architecture, the red line represents training data, whereas the blue line corresponds to validation data. The graphs and tables indicate a consistent increase in accuracy from epoch 1 to epoch 25, while the loss values decrease over time.

This study employs Transfer Learning with Inception-ResNetV2 and VGG16, yielding highly satisfactory results. The Inception-ResNetV2 model achieves an evaluation accuracy of 98.98% with a loss value of 0.0417, while VGG16 attains 99.40% accuracy with a loss value of 0.0146, demonstrating the model's effectiveness. The trained model is successfully deployed on Android devices, supporting a minimum specification of Android KitKat 4 and up to Android 12.

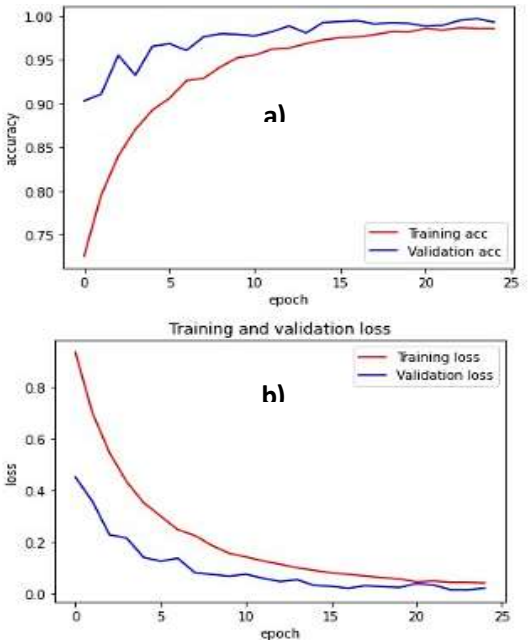

Figure 3. Training and Validation (a) Accuracy and (b) Loss of InceptionResNetV2

A comparison with the Indonesian Sign Language recognition system using video data reveals system performance metrics of 77.14% precision, 93.1% recall, 72.97% accuracy, and an F1-score of 84.38%, operating at a speed of 8 FPS (Daniels et al., 2022). Additionally, Inception-ResNetV2 achieves 91.09% accuracy, while an analysis of loss values across CNN models, including Xception, ResNetV2, VGG16, and VGG19, highlights specific performance trends (Kherraki & El Ouazzani, 2022). Notably, ResNet outperforms other models with 93.36% accuracy, facilitating effective two-way communication between individuals with hearing or speech impairments and the general public through

text-to-speech and sign language conversion (Raghavachari & Sundaram, 2021).

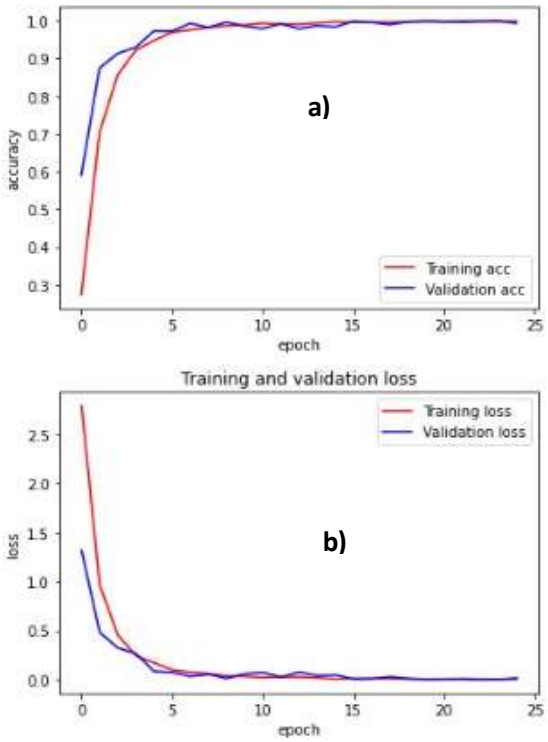

Figure 4. Training and validation
(a) Accuracy and (b) Loss of VGG16

Figure 5 illustrates the initial interface of the HARDISC application, displaying both the main page (Figure 5a) and the login page (Figure 5b). The main page features a visually engaging design with colourful hand illustrations, symbolizing inclusivity and the application's mission to facilitate communication for individuals with disabilities. The login page, on the other hand, is designed with a simple and intuitive layout, enabling users to enter their email and password effortlessly before accessing the main features. Additionally, the "Belum Punya Akun?" The (Don't have an account?) button enhances accessibility by allowing new users to register with ease. The increasing integration of digital technology highlights the importance of accessibility and equal opportunities, especially for individuals with disabilities.

Sign language detection applications, particularly those leveraging computer vision technology with Inception ResNetV2, serve as vital tools in bridging communication gaps for speech and hearing-impaired individuals. However, differing interpretations of accessibility across various disciplines, cultures, and interest groups lead to inconsistencies in implementation and standardization. To ensure broader inclusivity and usability, a universally accepted and well-defined concept of accessibility is crucial, facilitating measurable, standardized, and effective solutions for the target users (Persson et al., 2015).

(a) (b)

Figure 5. Application display (a) Main page and (b) HARDISC Application login page

The HARDISC application offers an intuitive main menu with essential functionalities for sign language translation and learning. The translation feature in Figure 6a converts sign language gestures into text or speech, facilitating communication for individuals with hearing impairments. The tutorial section provides interactive learning materials to enhance users' understanding of sign language. The camera and video functions in Figure 6b enable real-time gesture recognition, improving accessibility and usability. Additionally, the account and profile section in Figure 6c ensures a personalized experience by allowing users to manage their

settings and preferences.

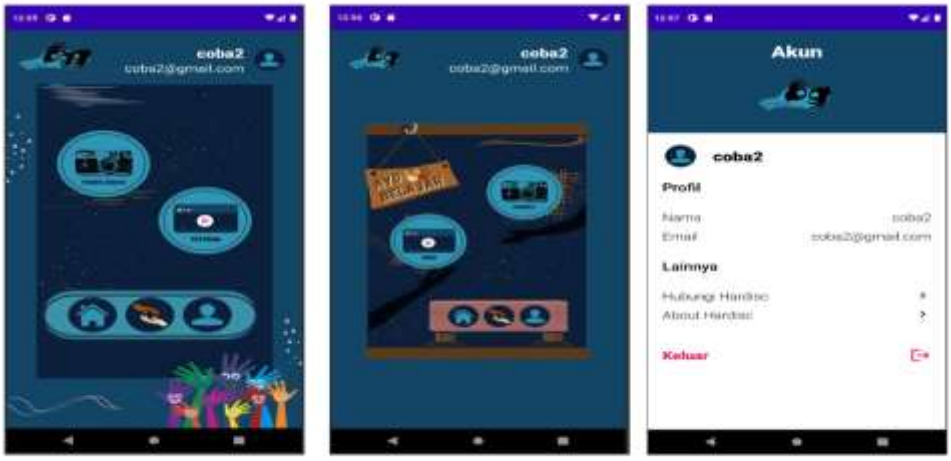

Figure 6. Main Menu of the HARDISC Application which appears in (a) translation and tutorial (b) camera and video (c) account and profile

The translator menu serves as the core component of the HARDISC application, where the OpenCV and TensorFlow libraries operate to process hand sign detection, requiring a GPU-equipped smartphone. Upon accessing the translator menu, the camera automatically activates, detecting sign language gestures in real-time while displaying bounding boxes and recognized letters for immediate feedback. Compared to YOLOv3, which excels in real-time gesture detection even in low-resolution environments, VGG16 provides higher classification accuracy but demands more computational resources and lacks real-time efficiency (Mujahid et al., 2021). The fingertip detection method enhances gesture recognition robustness, achieving a 95% success rate at distances of 35 cm and 55 cm under varying light conditions (90–100 lux) against a plain light green background (Triyono et al., 2018).

Furthermore, the detection results, initially captured as hand movements by the detector, generate output in the form of letters. When the camera is directed at the user's hand, the green line tracks and aligns the movement with the dataset images, instantly displaying the corresponding letters on the screen. Various applications also employ computer vision techniques to analyze similarities and distinctions, including hand segmentation methods, classification algorithms and their limitations, the number and type of gestures, the dataset used, detection range, and the type of camera utilized (Oudah et al., 2020). Hand gesture recognition undergoes detection and identification processes using six image segmentation methods, with optimal segmentation lighting results based on accuracy achieved through the Canny and HSV colour spaces (Fadel & Kareem, 2022). HARDISC applies a camera-based object detection method combined with a Convolutional Neural Network (CNN), ensuring high accuracy and speed. This principle is similarly applied in the YOLOv3-based application for recognizing Indonesian Sign Language (BISINDO) (Daniels et al., 2021). Consequently, this system facilitates communication between individuals with disabilities and the general public by

translating sign language in real-time through a camera-based application (Ojeda-Castelo et al., 2022). Additionally, the application provides a tutorial feature to assist users in navigating and utilizing HARDISC effectively. Designed as a socially beneficial innovation, the HARDISC application is free to use and does not require cellular data for access, making it available to anyone, anywhere.

CONCLUSION

HARDISC is an Android-based application that recognizes hand gestures and translates them into letters from A to Z using a camera. This AI-powered system helps both individuals with disabilities and the general public communicate more easily. The application utilizes Transfer Learning models, specifically Inception ResNetV2 and VGG16, which have demonstrated high accuracy. Inception ResNetV2 achieved an evaluation accuracy of 98.98% with a loss value of 0.0417, while VGG16 recorded an accuracy of 99.40% with a loss value of 0.0146, proving the effectiveness of these models. Unlike many other applications, HARDISC provides real-time sign language translation without requiring mobile data, making it accessible to a wider audience.

REFERENCES

Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications*, *12*, 200056.

Daniels, S., Suciati, N., & Fathichah, C. (2021, February). Indonesian sign language recognition using yolo method. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1077, No. 1, p. 012029). IOP Publishing.

Fadel, N., & Kareem, E. I. A. (2022). Detecting Hand Gestures Using Machine Learning Techniques. *Ingenierie des Systemes d'Information*, *27*(6), 957.

Fajri, B. B. R., Fajri, B. R., & Kusumastuti, G. (2019, December). Perceptions of 'Hearing'people on sign language learning. In *5th International Conference on Education and Technology (ICET 2019)* (pp. 364-367). Atlantis Press.

Frush Holt, R. (2019). Assistive hearing technology for deaf and hard-of-hearing spoken language learners. *Education sciences*, *9*(2), 153.

Kherraki, A., & El Ouazzani, R. (2022). Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring. *IAES International Journal of Artificial Intelligence*, *11*(1), 110.

Liu, K., & Kehtarnavaz, N. (2016). Real-time robust vision-based hand gesture recognition using stereo images. *Journal of Real-Time Image Processing*, *11*, 201-209.

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91-99.

Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021). Real-time hand gesture recognition based on deep learning YOLOv3 model. *Applied Sciences*, *11*(9), 4164.

Ojeda-Castelo, J. J., Capobianco-Uriarte, M. D. L. M., Piedra-Fernandez, J. A., & Ayala, R. (2022). A survey on intelligent gesture recognition techniques. *IEEE Access*, *10*, 87135-87156.

Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. *Journal of Imaging*, *6*(8), 73.

Papastratis, I., Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2021). Artificial intelligence technologies for sign language. *Sensors*, *21*(17), 5843.

Raghavachari, C., & Sundaram, G. S. (2020, July). Deep learning framework for fingerspelling system using CNN. In *2020 International Conference on Communication and Signal Processing (ICCSP)* (pp. 469-473). IEEE.

Razalli, A. R., Rakoro, J. U., Ariffin, A., Hashim, A. T., & Mamat, N. (2019). Factors affecting sign language acquisition in hearing impaired learners during primary education. *Religación: Revista de Ciencias Sociales y Humanidades*, *4*(15), 202-209.

Renauld, J. M., & Basch, M. L. (2021). Congenital deafness and recent advances towards restoring hearing loss. *Current protocols*, *1*(3), e76.

Sibuea, S., Saputro, M. I., Annan, A., & Widodo, Y. B. (2022). Aplikasi Mobile Collection Berbasis Android Pada Pt. Suzuki Finance Indonesia. *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, *2*(1), 31-42.

Tran, A. D., Nguyen, M. Q., Phan, G. H., & Tran, M. T. (2021). Security issues in android application development and plug-in for android studio to support secure programming. In *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications: 8th International Conference, FDSE 2021, Virtual Event, November 24–26, 2021, Proceedings 8* (pp. 105-122). Springer Singapore.

Triyono, L., Pratisto, E. H., Bawono, S. A. T., Purnomo, F. A., Yudhanto, Y., & Raharjo, B. (2018, March). Sign language translator application using openCV. In *IOP Conference Series: Materials Science and Engineering* (Vol. 333, No. 1, p. 012109). IOP Publishing.

Zeshan, U. (2015). "Making meaning": Communication between sign language users without a shared language. *Cognitive Linguistics*, *26*(2), 211-260.