I. J. Engineering and Manufacturing, **2024, 6, 1-10**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2024.06.01

Indonesian Sign Language: Detection with Applying Convolutional Neural Network in a Song Lyric

Qurrotul Aini*
Department of Information Systems, UIN Syarif Hidayatullah, Jakarta, 15412, Indonesia
E-mail: qurrotul.aini@uinjkt.ac.id
ORCID iD: https://orcid.org/0000-0002-2722-4755
*Corresponding Author

Ferdian Rachardi
Project Manager, PT. Trakindo Utama, Jakarta Selatan, 12560, Indonesia
E-mail: ferdianrachardi@gmail.com
ORCID iD: https://orcid.org/0009-0001-6921-2242

Zainul Arham
Department of Information Systems, UIN Syarif Hidayatullah, Jakarta, 15412, Indonesia
E-mail: zainul.arham@uinjkt.ac.id
ORCID iD: https://orcid.org/0000-0002-7899-5971

Received: 28 July, 2024; Revised: 25 September, 2024; Accepted: 14 October, 2024; Published: 08 December, 2024

Abstract: Indonesian Sign Language (BISINDO) is one of the visual-based alternative languages used by people with hearing impairments. There are hundreds of thousands of Indonesian vocabularies that sign language gestures can represent. However, because the number of deaf people in Indonesia is only seven million or 3% of the population, sign language has become unfamiliar and challenging for some normal or laypeople to understand. This study aims to classify and detect gestures in sign language vocabulary directly based on mobile. Classification learning techniques are needed to recognize variations in gestures, such as machine learning with supervised learning techniques. The development of this research uses the convolutional neural network method with the help of techniques from the single shot detector architecture as the object of detection and the MobileNet architecture for classification. The object is 32 gestural vocabularies from the lyrics of the song 'Bidadari Tak Bersayap' with a dataset of 17,600 images. Then the images are divided into two parts of the model based on the nature of the biased and non-biased data, amounting to 8 and 24 classes, respectively. The research results in a biased model prediction of 15 out of 16, while a non-biased model of 36 out of 48 correct predictions with a total accuracy of real-time based testing on mobile of 93.75% and 75%, respectively.

Index Terms: Indonesian sign language, BISINDO, Machine learning, Supervised learning, Convolutional neural network.

1. Introduction

With a population of around 270 million, Indonesia has various languages that grow differently based on each person's background. One common ability possessed by ordinary people is to communicate verbally. However, some people have limitations in verbal communication, such as deafness and speech impairment. As the Central Bureau of Statistics issued the results of the Inter-Census Population Survey (SUPAS) 2015, the number of population in all provinces of Indonesia with hearing difficulties is seven million [1].

Because of the high number of deaf people, the Indonesian government continues to pay attention to human rights as stipulated in the regulations of Law No. 8 of 2016 and article 26 paragraph 1 concerning freedom in socializing and interaction. However, the facts in the field are social disparities experienced by persons with disabilities, particularly the deaf, such as difficulties and awkwardness in interacting directly with ordinary people. In addition, there are differences

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

in the types of sign language between Indonesian Sign System (SIBI) and BISINDO. SIBI is a structured and rigid Indonesian sign language system; it grows from ordinary people, while BISINDO is an Indonesian sign language which is flexible, based on visual objects, not rigid or practical, and grows from deaf people [2,3,4].

The previous study identified sign language objects with various methods and tools, including classification-object detection with Support Vector Machine (SVM) on four gestures of Indian sign language vocabulary (ISL) via a Kinect Microsoft desktop camera with an accuracy of 97.5% [5]. Also, the detection-object classification used the Haar Classifier K-Nearest Neighbors algorithm (KNN) via real-time and non-real-time desktop on BISINDO letter and number gestures of 19 and 24 gestures with 89.68% and 91.8% accuracy results [6,7]. In contrast, the classification study using the Convolutional Neural Network (CNN) method on ASL letter and number gestures was 11, 26, and 36 gestures with accuracy results of 97%, 82.5%, and 94.68% [8,9]. According to the findings of reference [10], the utilization of CNN for classifying hand movements into letters yields an impressive accuracy rate of 95.4%. The datasets employed in this study were obtained from Kaggle and were divided into 80% training data and 20% test data. Based on the comparison of these studies, the CNN method produces much higher accuracy values than other methods. Therefore, this study will identify sign language using the CNN method on 32 Indonesian sign language vocabulary (BISINDO) from the song lyrics 'Bidadari Tak Bersayap' by utilizing a real-time-based smartphone camera.

This study has classified and detects images of Indonesian Sign Language vocabulary gestures. First, the collection of image datasets in Indonesian sign language vocabulary, totaling 32 gestures based on the song “Bidadari Tak Bersayap”. Second, image preprocessing consists of resizing, augmentating to increase the variety of image data, and labeling image data. Third, converting image data and configuring the model with CNN single shot detector (SSD) MobileNet, training the dataset, and obtaining performance values. Fourth, designing a user interface (UI) application for implementing an Android-based model and testing the application.

2. Proposed Approach

2.1. Single Shot Detector (SSD)

Single shot or multibox detector is an architecture for object detection models that can detect multiple objects in a single image [11]. The research results [12] demonstrate the speed and accuracy of three CNN architectures: SSD, YOLO, and Faster RCNN. They found that Yolo-v3 is the fastest, followed by SSD, while Faster RCNN is in the last position. However, the use of this architecture influences the chosen algorithm. When using a relatively small dataset that does not require real-time results, it is best to use Faster RCNN. Yolo-v3 is the right choice if there is a need to analyze live video feeds. Meanwhile, SSD provides a satisfactory balance between speed and accuracy. As a result, the researchers selected SSD as the architecture for this study.

2.2. Dataset

This study uses a dataset of image images that defines sign language gestures according to each vocabulary in the song "Bidadari Tak Bersayap" which consists of 32 gestures. Then, the researcher split the dataset into 2 types of data, namely 90% training data and 10% testing data with random technique, which each class totaling 500 training images and 50 testing images, the total dataset amounting to 16,000 and 1,600 image images as in Table 1. The specifications of the smartphone used are Lenovo A7700, OS Android 6.0 Marsmallow, CPU @1 GHz, MediaTek MT6735P, RAM 2 GB.

Table 1. Dataset Sharing

No	Dataset	Split	Total of Number
1	Data training	90%	16,000
2	Data testing	10%	1,600
		100%	17,600

Because machine learning research requires a variety of data variations, the authors carry out various variations in image data by distinguishing the subject's costumes on the image by making gestures as in Table 2. This research does not measure the performance before and after image augmentation. However, image augmentation is useful for reducing overfitting, increasing the robustness of the model, and improving the accuracy of the model used.

Table 2. Data Training and Data Testing

Data Training	
	
	
Data Testing	
	

2.3. Research Process

The research process of classification begins with data gathering, data preprocessing data, model configuration, data training, data testing, and the result as shown in Fig. 1. During the data gathering phase, the author collected image data, specifically gestures in Indonesian sign language vocabulary, which became the object of research. Next, perform data augmentation to increase the variation of image data using data resizing and data augmentation techniques, as well as labelling the image data. Then, the author converted the image data to .csv, then to TFRecord, and configured the model with SSD MobileNet. The author then conducted a training process to learn from the dataset and obtain evaluation scores by studying the objects. The author then designs the application and user interface (UI) to implement the model on a mobile-based platform, specifically Android. Then conduct testing based on scenarios and simulations to obtain results.

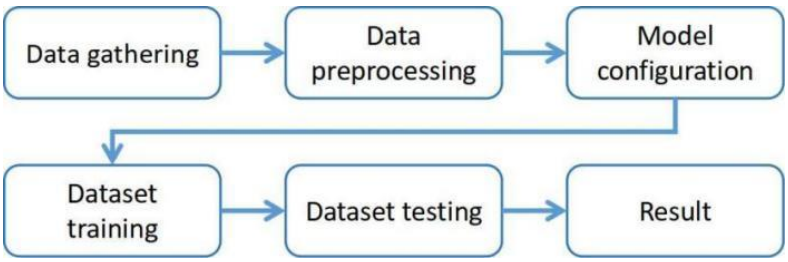

Fig. 1. Research process of classification.

2.4. Data Preprocessing

Before conducting the training process on the model, the writer multiplied the image variations up to 500 image data per class by performing image augmentation techniques such as size, rotation, distortion, light, contrast, and others. After that, carry out the data labeling process on hand gestures as in Fig. 2.

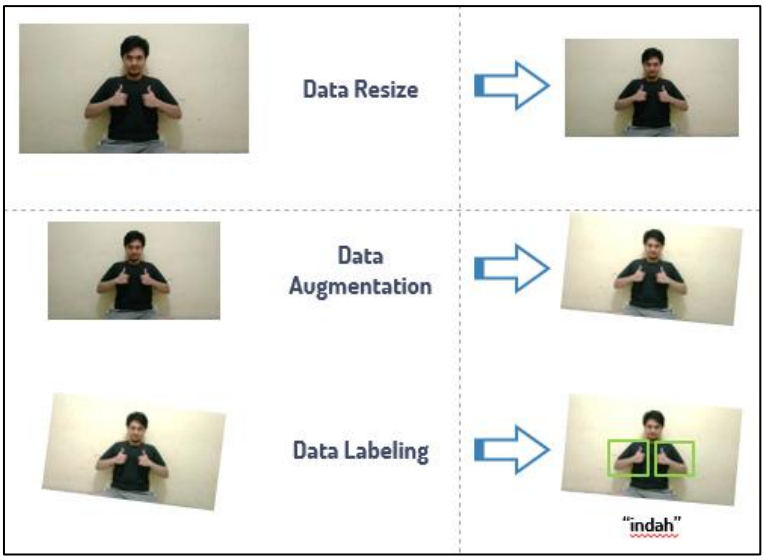

Fig. 2. Data preprocessing and labeling process.

2.5. Model Design

In this study, the authors use the CNN method with the help of the MobileNet SSD architecture model because it produces fairly good accuracy and speed values with a much smaller load parameter so that it can be used for devices that process lighter computing, in particular on smartphones [11,13], as shown in Fig. 3.

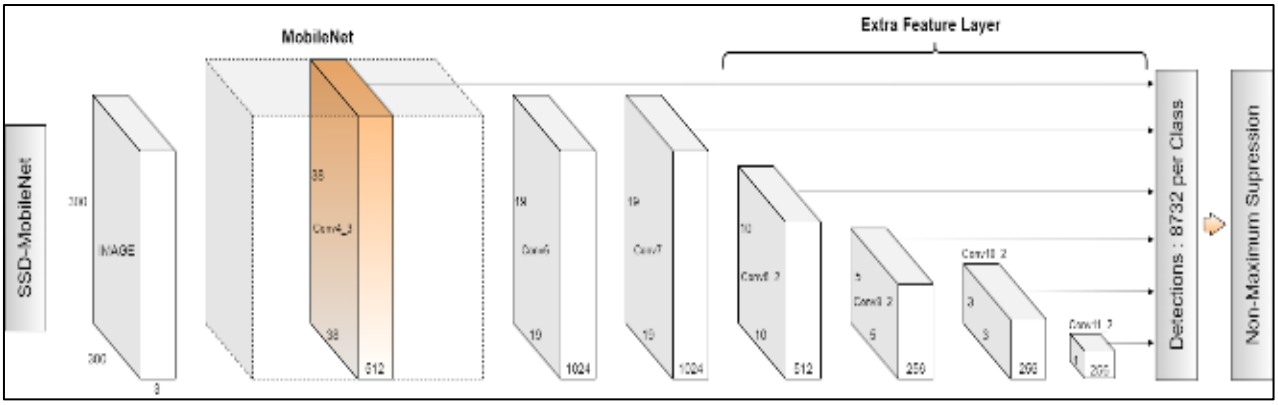

Fig. 3. Model SSD MobileNet.

In Fig. 3, the MobileNet SSD model has 2 types of screens consisting of:

1) Base network to perform the task of classification and extraction of image features.

On this screen it is useful for the model to perform the feature extraction process or take special characters in the image trained with the standard classification model architecture used by VGG-16 [11], but due to the complexity and burden it has, the authors change the model architecture to MobileNet [14,15].

2) Detection network to detect certain areas of the image as well as perform classification.

On this screen (extra feature layer) convolution several times on each screen with various map feature scale sizes, the closer the scale size decreases, which ends with the non-maximum surpression technique as filtering the highest value in the training process.

The model process begins by resizing the pixels in the image to $300 \times 300 \times 3$, then convoluting the 7 layers with the first layer performing the feature extraction process using the mobileNet architecture with the feature layer size being $38 \times 38 \times 512$, then the second layer becoming $19 \times 19 \times 1024$, and so on, making it $1 \times 1 \times 256$. From the results of the convolution, the number of bounding box detections is 8732 per class.

2.6. Train Dataset

In the training process using a total of 16,000 image data labeled 500 data per class each. The amount of training data is more than the test data with different data criteria, this is because the machine will produce better classification performance and accuracy on the test data.

2.7. Test Dataset

At this stage, the test data used is 50 image data per class with a total of 1,600 image data. Then, the testing process uses 2 different stages, namely testing with mean average precision (mAP) after the train dataset process and testing with accumulated accuracy when implemented on real-time based smartphone devices.

2.8. Performance Measure

The algorithm used in the Python 3.7.1 programming language is a convolutional neural network and is applied to a train or test dataset on the Google Colaboratory cloud server with the help of the Tesla K80 GPU specification and various types of libraries in Python such as OpenCV, Augmentor, Tensorflow and Tensorboard.

As for this research, performance measurement uses average precision (mAP) and accuracy. Meanwhile, the formula for average precision is usually followed by precision interpolation [15] as in (1) and (2).

$$p_{interp(r)} = \max p(\tilde{r}) \tag{1}$$

$$AP = \frac{1}{11}\sum_{r\in\{0.0.1,..0.9,1\}} p_{interp(r)} \tag{2}$$

Then, to calculate accuracy based on confusion matrix [16] where there are four variables that determine the performance of the detection and classification object, namely [17].

1) True Positive (*TP*): Correct detection and classification, with an IOU value ≥ threshold or threshold.
2) False Positive (*FP*): false or multiple detection, but correct classification with IOU value <threshold.
3) False Negative (*FN*): The object is not detected and is classified incorrectly with an IOU value ≥ threshold.
4) True Negative (*TN*): Not used.

Then, the accuracy formula is based on the 4 variables above and is calculated as in (3).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{3}$$

Average precision measurement usually requires a precision-recall curve plotting, but it is necessary to find the precison interpolation value for each object, this is because it can reduce the impact of wiggles wobbling caused by small variations in detection ratings [13]. Meanwhile, accuracy functions as the accuracy of a model in object detection.

3. Experiment Result

3.1. Training Process

When carrying out the training process, there are several parameters that measure how well the model is running, namely the loss function. The loss function in training consists of the difference between the localization loss weight value and the confidence loss weight value for each running epoch process:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{4}$$

Equation (1) gives the loss value from the convolution result when the training process consisting of l as the predicted box, g as the ground-truth box, Lconf as the loss confidence, Lloc as the loss location, N as the number of posts, and α as the weight value [8]. The smaller the loss value, the better the training process for the model.

3.2. Evaluation Result

The purpose of this study is to identify object detection on 32 gestures using the MobileNet SSD architecture model. Before the model is implemented on the device, the authors analyze and test the model with the parameters of the loss value during training and the mAP value in the testing data previously described with the following results:

1) Model - 32 Class

In a model that uses 32 classes or direct gestures, the authors get a high loss value during the training process, as in Fig. 4. Even the mAP value gets poor results below the threshold threshold (> 0.5), namely 0.25, as in Table 3.

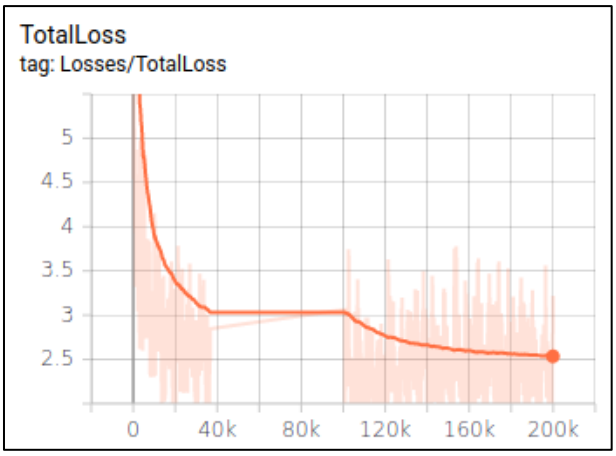

Fig. 4. Loss value in 32 classes.

Table 3. MAP results of the 32 class model

Mean Average Precision (MAP)
0.25

Then, from these results, the authors look for alternative solutions to be able to produce a better loss value and mAP value than the 32 class model. Due to the large number of classes and some gestures that have similar or biased gestures and the nature of the gestures that move statically or dynamically, making the classification process difficult for the model, the authors divide the model into 2 parts, namely:

2) Bias Model

In this model, the writer makes a model based on dynamic gestures and separates the types of movements that are almost similar to other vocabulary gestures so that the class becomes 8 classes or vocabulary gestures. The loss value obtained is quite low and the mAP value is high as in Fig. 5. Even the mAP value gets poor results below the threshold threshold (> 0.5), namely 0.89, as in Table 4.

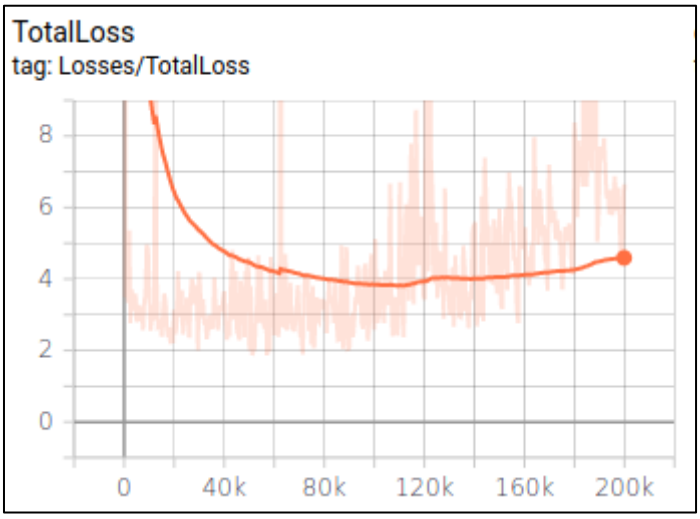

Fig. 5. Loss value in 8 classes.

Table 4. MAP results of the 8 class model

Mean Average Precision (mAP)
0.89

3) Non Bias Model

In this model, it is based on the type of movement that is static so that the number of classes is assembled into 24 classes or vocabulary gestures. The loss value is low and the mAP value is quite high as in Fig. 6 and Table 5.

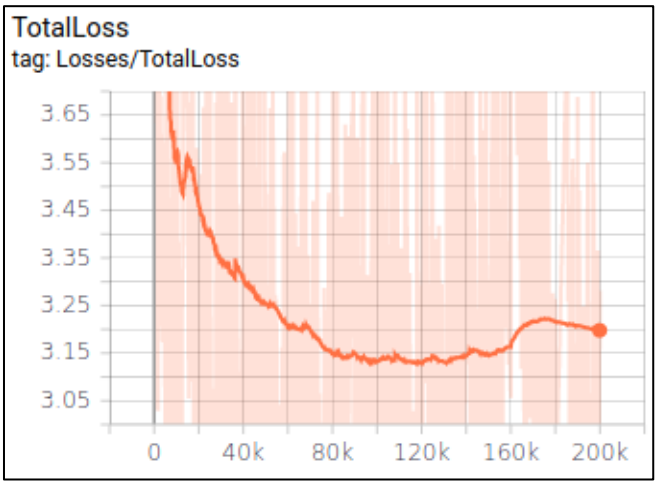

Fig. 6. Loss value in 24 classes.

Table 5. MAP results of the 24 class model

Mean Average Precision (mAP)
0.99

Loss function is a performance metric that measures and recognises how effectively the model conducts learning or training. Therefore, a variable between prediction and actual is required in order to know the error value acquired. If the loss function value is smaller, then the model that is being used is more effective. As the training procedure is carried out, a number of different loss function values will be discovered. In the event that the value of the loss function decreases and reaches the lowest point while maintaining a constant value, this indicates that the classification value is improving.

3.3. Testing

At this stage, the authors tested the testing data to determine the accuracy of each model using a smartphone-based camera. As for testing using the Lenovo A7700 smartphone with an 8 megapixel resolution camera, the test data used is limited to 2 types of costumes per class with the amount of data on the bias and non-biased models being 16 and 48 image data, as shown in Fig. 7.

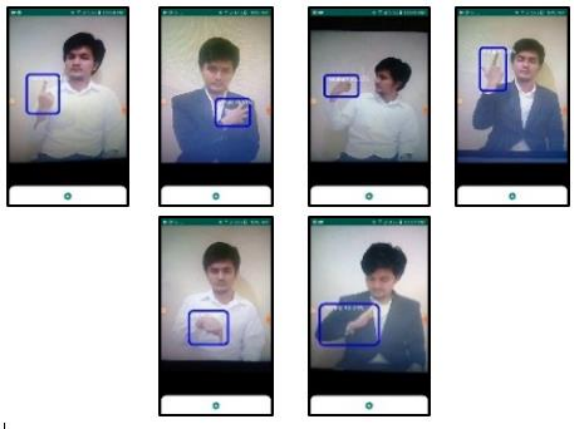

Fig. 7. Test results on smartphones.

The author elucidates the practical challenges associated with smartphone-based sign language detection, both from the perspective of the smartphone and the user. Mobile devices, in comparison to desktop computers or servers, possess less processing power due to their limitations. Power consumption using intensive machine learning algorithms

can drain the battery quickly. Meanwhile, memory and camera quality also have a significant effect. In addition, real-time data processing on smartphones can introduce latency that disrupts the user experience. On the user side, the design of the user interface should be easy because the user is a person with special needs, so it is very important to ensure that the user can interact with the sign language detection application easily. Furthermore, the authors document and calculate the accuracy of all image tests per model with (2) then the results are shown in Table 6.

Table 6. Test Accuracy Results

Class	Model	Accuracy
8	Bias	93.75%
24	Non Bias	75%

3.4. Discussion

After obtaining the results of the accuracy of the bias and non-biased models, a comparison of the accuracy with the SVM, Haar Classifier-KNN, and CNN methods was carried out, as in Table 7.

Table 7. Accuracy Comparison Results

Class	Method			
	SVM	*Haar Classifier - KNN*		CNN – SSD MobileNet
4	97.50%	-	-	-
8	-	-	-	93.75%
19	-	89.68%	-	-
24	-	-	91.80%	75%

Based on the accuracy results, it can be concluded that the CNN-SSD MobileNet method gets an accuracy value that is slightly lower than the SVM method but has the advantage of having more classes. On the other hand, the CNN - SSD MobileNet method has a much lower accuracy value than the Haar Classifier-KNN method with the same number of classes but has different detected objects, namely letters.

There are several limitations of this research, including limited vocabulary size, bias in dataset collection, challenges in scaling the model for larger vocabularies, and more diverse platforms. To overcome these, the use of more complex learning models such as BERT, GPT, GNN, or LSTM, or even their hybrids, is one alternative. Collecting more data from different sources, data augmentation, and working with sign language experts should also be considered.

4. Conclusions

This study has succeeded in determining the application of classification models to gesture gestures, obtaining classification results and accuracy of gesture gestures, knowing the factors that affect the accuracy results and comparing other methods of classification. Also, the study has succeeded in classifying gesture gestures using 2 models, namely 8 class models totaling 15 out of 16 correct predictions and 24 class models totaling 35 of 48 correct predictions. The relationship between the number of classes and the nature of the gesture has a significant effect on accuracy in real time testing, so that making 2 models with different numbers and properties such as the 8 class model which is biased produces an accuracy value of 93.75% and the 24 class model which is non-bias results in an accuracy value of 75%. Tests carried out using 16 image data for 8 class models that can produce the correct number of predictions, which is 15 out of 16 image data with an accuracy value of 0.9375 or 93.75%. Meanwhile, for the 24 class non-bias model, the number of correct predictions was 36 out of 48 image data with a lower accuracy value of 0.75 or 75%. Future work this study to make it even better is to explore the model by adding the number of datasets, data variations, iterations and image data resolutions, then comparing the CNN method and its architecture with others, and improving application features such as compiling vocabulary becomes a sentence that comes from sign language gestures using a sorting algorithm such as insert sorting, quick sorting, and others.

References

[1] Statistics Indonesia, *Intercensus Population Survey Results (SUPAS)*. Jakarta: Statistics Indonesia, 2015.
[2] D. Gerkatin, *Get acquainted with Bisindo*. Jakarta: DPD Gerkatin, 2010.
[3] D. Pradama, "Perancangan Informasi Bahasa Isyarat Bisindo Pada penyandang tuna rungu melalui media interaktif aplikasi Android," *Repository UNIKOM*, 21-Dec-2017. [Online]. Available: https://repository.unikom.ac.id/57710/. [Accessed: 07-Mar-2022].

[4] G. Gumelar, H. Hafiar, and P. Subekti, “Indonesian sign language as a deaf culture through the meaning of movement members for the welfare of the deaf,” *INFORMASI: Kajian Ilmu Komunikasi*, vol. 48, no. 1, pp. 65–78, 2018, doi: 10.21831/informasi.v48i1.17727.
[5] J. L. Raheja, A. Mishra, and A. Chaudhary, “Indian sign language recognition using SVM,” *Pattern Recognit. Image Anal.*, vol. 26, no. 2, pp. 434–441, 2016.
[6] R. Z. Fadillah, A. Irawan, and M. Susanty, “Model penerjemah bahasa isyarat indonesia (BISINDO) menggunakan pendekatan transfer learning,” *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, vol. 15, no. 1, pp. 1-9, 2022, doi: https://doi.org/10.33322/petir.v15i1.1289.
[7] R. I. Borman, B. Priopradono, and A. R. Syah, “Classification of hand coded objects in indonesian sign language alphabet sign recognition (bisindo),” in *Proceeding of National Seminar Computer Crime and Digital Evidence*, September, pp. D1–D4, 2017.
[8] V. Bheda and D. Radpour, “Using deep convolutional networks for gesture recognition in American sign language,” *arXiv.org*, 20-Nov-2017. [Online]. Available: https://arxiv.org/abs/1710.06836. [Accessed: 07-Mar-2022].
[9] S. Ikram and N. Dhanda, “American sign language recognition using convolutional neural network,” *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 2021, pp. 1–12, doi: 10.1109/GUCON50781.2021.9573782.
[10] M. B. Tamam, H. Hozairi, M. Walid, and J. F. A. Bernardo, “Classification of Sign Language in Real Time Using Convolutional Neural Network,” *Appl. Inf. Syst. Manage.*, vol. 6, no. 1, pp. 39–46, 2023, doi: 10.15408/aism.v6i1.29820.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” *arXiv.org*, 29-Dec-2016. [Online]. Available: https://arxiv.org/abs/1512.02325. [Accessed: 07-Mar-2022].
[12] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, “Comparative analysis of deep learning image detection algorithms,” *Journal of Big Data*, vol. 8, Art. no. 66, 2021, doi: 10.1186/s40537-021-00434-w
[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv.org*, 17-Apr-2017. [Online]. Available: https://arxiv.org/abs/1704.04861. [Accessed: 02-Apr-2022].
[14] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3296–3305, 2017.
[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[16] S. J. Putra, Y. Sugiarti, G. Dimas, M. Nur Gunawan, T. Sutabri and A. Suryatno, "Document Classification using Naïve Bayes for Indonesian Translation of the Quran," *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, 2019, pp. 1-4, doi: 10.1109/CITSM47753.2019.8965390.
[17] R. Padilla, S. L. Netto and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237-242, doi: 10.1109/IWSSIP48289.2020.9145130.

Authors’ Profiles

Qurrotul Aini is currently working as an Assistant Professor and Head of Department of Information Systems, Faculty Science and Technology UIN Syarif Hidayatullah Jakarta Indonesia. She received her Doctor of Electrical Engineering degree from Institut Teknologi Sepuluh Nopember Surabaya Indonesia. She is a Member IEEE since 2018. Her research interests include business intelligence, intelligence computation, and multimedia application.

Ferdian Rachardi is currently working as business process manager at PT. Trakindo Utama. He holds a Bachelor’s degree in computer science from Department of Information Systems UIN Syarif Hidayatullah Jakarta in 2020. His expertise in project management office, supply chain management, quality control, human resource information system, and system analysis.

Zainul Arham is currently working as an Assistant Professor in Department of Information Systems UIN Syarif Hidayatullah Jakarta. His research interests include geographic information system (GIS), image processing, and software engineering.

How to cite this paper: Qurrotul Aini, Ferdian Rachardi, Zainul Arham, "Indonesian Sign Language: Detection with Applying Convolutional Neural Network in a Song Lyric", International Journal of Engineering and Manufacturing (IJEM), Vol.14, No.6, pp. 1-10, 2024. DOI:10.5815/ijem.2024.06.01