International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5759~5769
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5759-5769

Faster R-CNN implementation for hand sign recognition of the Indonesian sign language system (SIBI)

Paulus Lestyo Adhiatma, Nurcahya Pradana Taufik Prakisya, Rosihan Ariyuana
Department of Informatics and Computer Engineering Education, Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta, Indonesia

Article Info

Article history:

Received Aug 28, 2024
Revised Aug 6, 2025
Accepted Sep 16, 2025

Keywords:

Faster R-CNN
Object recognition
ResNet
SIBI
Sign language

ABSTRACT

The Indonesian sign language system (SIBI) is the authorized sign system in Indonesia that the deaf society uses to convey in Indonesian. However, its use still needs to be expanded and more widespread in the community, causing difficulties in communication for hard-of-hearing people. The product of deep learning technologies such as faster region-based convolutional neural network (Faster R-CNN) in object recognition has the potential to help improve communication between deaf people and the general public. This research will implement the Faster R-CNN algorithm with three different residual network (ResNet) architectures (50, 101, and 152) for SIBI recognition. The comparison of the faster R-CNN algorithm with different architectures is also conducted to identify the best architecture for SIBI recognition, and the results are evaluated using accuracy, precision, recall, and F1-score metrics from confusion matrix calculation and execution time. Faster R-CNN model with ResNet-50 architecture showed the best and most efficient performance with accuracy, recall, precision, and F1-score metrics of 96.15%, 95%, 93%, and 94%, respectively, and an execution time of 36.84 seconds in the testing process compared to models with ResNet-101 and ResNet-152 architectures.

Corresponding Author:

Nurcahya Pradana Taufik Prakisya
Department of Informatics and Computer Engineering Education, Faculty of Teacher Training and Education, Universitas Sebelas Maret
Ir. Sutami Street 36 Kentingan, Jebres, Surakarta, Indonesia
Email: nurcahya.ptp@staff.uns.ac.id

1. INTRODUCTION

Gestures and signs are considered the considerable native way to obtain messages between people through body motions. Although gestures and signs are categorized as non-verbal contact, they can effectively express messages between hard-of-hearing people. Sign language is the most commonly used method of conveying words using body movements [1]. Sign language is vital in communication for the speech-impaired and deaf community [2].

Indonesian sign language system (SIBI) is Indonesia's official sign language, standardized with specific finger and hand movements that follow the grammar and structure of the Indonesian language, as recognized in Ministerial Decree Number 0161/U/1994 [3]. Although SIBI is the authorized sign language in Indonesia, its use is still limited and has yet to be widely spread in the community. Not all Indonesians understand the sign language used by deaf people, so it will cause difficulties in communication for deaf people when interacting with people who do not understand sign language [4]. Many people find it difficult to interact and relate to deaf people, so they do not get regular social interactions because of their limitations, resulting in social inequality [5]. To address these communication barriers and promote social inclusion,

technological innovations in sign language recognition are critical. Recent advancements in deep learning, particularly in computer vision, offer promising solutions to automate SIBI recognition and bridge the communication gap [6].

Unlike isolated gesture recognition, SIBI requires precise detection of dynamic hand configurations and spatial relationships. Faster region-based convolutional neural network (Faster R-CNN), with its ability to localize and classify objects in complex scenes, is particularly suited for this task. The Faster R-CNN algorithm is a deep learning algorithm for object detection algorithm that combines a convolutional neural network (CNN) to detect region proposals and classify objects simultaneously by utilizing the region proposal network (RPN), which enables faster and more accurate processing time [7]. Several previous studies on identification using the Faster R-CNN algorithm have been conducted. In the study by Deng *et al.* [8], Faster-RCNN was applied to detect diabetic retinal disease using ResNet50 and VGG16 for feature extraction. The research demonstrated that the ResNet50-based model achieved a higher mean average precision (mAP) of 97.42% and a precision of 98.96%. Another research was conducted by Sabir *et al.* [9] using a transfer learning strategy with the Faster R-CNN model to detect faces that use masks and those that do not use masks obtained the most elevated average precision (AP) of 81% and the most elevated average recall (AR) of 84%. Research by Cao *et al.* [10] presents an improved algorithm based on Faster R-CNN for small object detection, addressing challenges such as complex backgrounds, occlusion, and low resolution, achieving a recall rate of 90% and an accuracy rate of 87% for traffic signs. Based on these studies, the Faster R-CNN algorithm has excellent potential in object recognition.

Selecting the suitable algorithm is crucial for implementing Faster R-CNN in recognizing the Indonesian language system (SIBI). The residual network (ResNet) CNN architecture, introduced by He *et al.* [11], has proven effective in overcoming the performance degradation problem in deeper CNNs. ResNet can overcome the challenge of learning complex and deep representations using shortcut connections. Frequently used variants of the ResNet architecture are ResNet-50, 101, and 152, each of which has an appropriate number of layers to handle different levels of complexity. The ResNet architecture in the Faster R-CNN algorithm serves as the backbone in performing digital image feature extraction [12]. ResNet architecture as a backbone/foundation has been widely used in complex tasks, such as object detection and instance segmentation [13]. In this study, we employ the Faster R-CNN algorithm due to its strong accuracy performance in object detection tasks, particularly when dealing with detailed spatial features such as hand gestures. Compared to other detection models like YOLO and SSD, which prioritize inference speed, Faster R-CNN is more suitable for scenarios that demand high precision. This makes it ideal for recognizing fine-grained hand sign variations in SIBI, where detection accuracy is paramount. Despite its slower inference speed, Faster R-CNN demonstrates superior accuracy, making it ideal for applications where detection precision is crucial [14]. Within this framework, we integrate three variants of the ResNet backbone ResNet-50, ResNet-101, and ResNet-152 and conduct a structured comparative analysis to recognize multi-class SIBI hand signs. The novelty of this research is reflected in evaluating the trade-offs between detection accuracy, inference time, and model complexity under limited data conditions, providing practical insights for selecting efficient backbone architectures for real-world assistive technology applications.

2. RESEARCH METHOD

The study method employed is the research and development (R&D) method as illustrated in Figure 1. This research implements the Faster R-CNN algorithm to identify SIBI sign language in digital images. The implementation results will then be compared to determine the best architecture as a backbone for Faster R-CNN. The research process begins with data collection, where images of SIBI signs are gathered, followed by object annotation to label the regions of interest within each image. The annotated data undergoes data preprocessing to standardize and augment the images, ensuring their suitability for the training model phase, where the Faster R-CNN is trained to recognize SIBI signs. Finally, in the evaluate phase, the model's performance is assessed through various metrics. The source code for this work is publicly available at: *https://github.com/paul-lestyo/faster-rcnn-sibi-handsign*.

Figure 1. Research flow

2.1. Datasets

The dataset used in this research consists of 546 labeled digital images representing 26 hand sign classes (letters A to Z) from the Indonesian Sign Language System (SIBI), with 21 images per class, as shown in Figure 2. This dataset was obtained from an open-source collection on Kaggle [15]. All images are in JPG format with a uniform square resolution of 2000×2000 pixels. Each image was manually annotated using the LabelImg tool, producing XML files that contain bounding box coordinates and associated class labels. The annotation files were then converted to CSV format and subsequently into TFRecord format to prepare the dataset for TensorFlow's object detection pipeline. To introduce slight variation, data augmentation was applied during training, including horizontal flipping and brightness adjustment. These techniques produced variants image and improved model robustness by simulating slight environmental changes. However, the dataset presents limitations in diversity and representativeness. Most of the images feature similar lighting conditions, backgrounds, and hand characteristics, with minimal variation in skin tone, hand size, or environmental context. These factors may introduce dataset bias and limit the model's generalization ability in real-world settings. A more comprehensive and varied dataset would be necessary to improve robustness, particularly in practical assistive applications intended for diverse user populations.

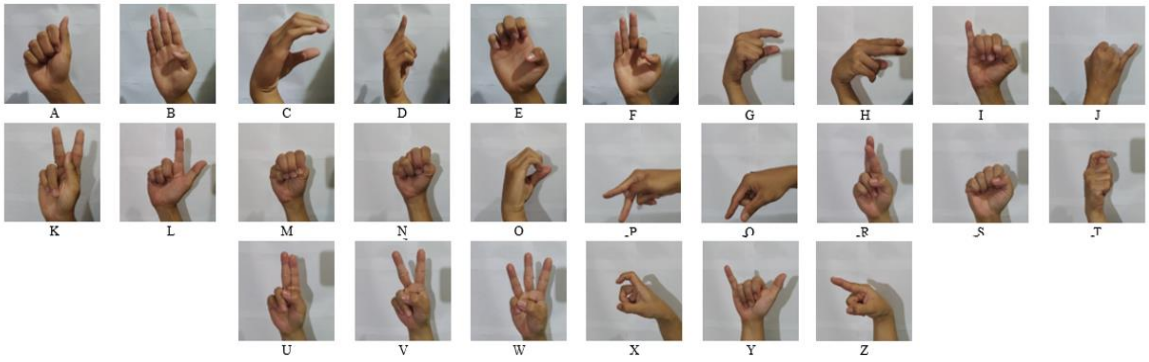


Figure 2. Example of the SIBI

2.2. Object annotation

Object labeling or annotation is used to determine information related to the class and special features that represent objects in each image by creating a bounding box. This aims to train the model to be able to recognize objects that will be predicted. Object labeling in this research is done manually using the Labelimg tool. Labelimg is run in the Anaconda tools environment installed on the Windows operating system. Object labeling results in a .xml file with information regarding the object's class and the bounding box coordinates. The object labeling process produces one object class per image with 546 objects. After labeling objects on digital images using LabelImg, an XML file is produced that contains information related to the class and coordinates of the object.

2.3. Datasets preprocessing

In the data preprocessing phase, the collected Indonesian Sign Language System (SIBI) digital images are prepared for the needs of model training using Tensorflow. The initial step in the data preprocessing process involves the data that has been prepared along with the coordinate point information of objects and classes. Both data are converted into TFRecord (TensorFlow record) format. This process begins with the *.xml* file created from the object labeling process transformed into a *.csv* file. Furthermore, the *.csv* file created is used as the basis for the conversion process to the TFRecord format. In this process, the digital image information in the *.csv* file is processed in a TFRecord specification format.

2.3.1. Dimension resizes

There is preprocessing that occurs in the data pipeline, where resizing is performed on both image size and object coordinate points. Resize on image size aims to ensure that all images have uniform dimensions, while resize on object coordinate points aims to maintain object proportions after image resize.

2.3.2. Image augmentation

The augmentation performed during model training consists of two main types of transformations. First, horizontal flipping with a probability of 0.3, producing a horizontally mirrored image to improve

orientation invariance. Second, brightness adjustment with a maximum delta of 0.3, enhancing robustness to lighting variations. Figure 3 shows image augmentation with horizontal flip and brightness adjustment.

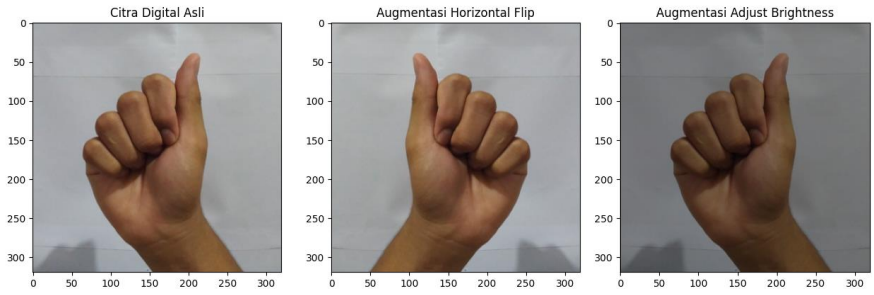


Figure 3. Data augmentation examples horizontal flip and adjust brightness

2.4. Proposed faster R-CNN with residual network

Faster R-CNN is a network that combines Fast R-CNN and RPN to decrease duration complexity and generate high-quality region proposals, bounding boxes, and objectness scores simultaneously [16], which is illustrated in Figure 4. In study [17], the phases of the Faster R-CNN are as follows:

- The input digital image undergoes convolution and pooling operations through feature extraction to obtain a feature map.
- The feature map is then given to the RPN network, which performs objectness prediction.
- The RPN network provides several proposed anchors through coordinate points and objectness prediction scores.
- The RPN output is sent to the Fast R-CNN network, where the object output is processed in the fully connected layer for classification.

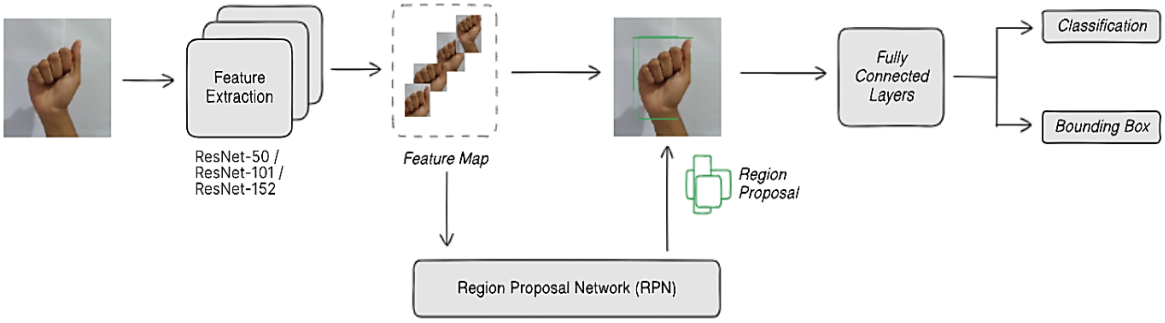


Figure 4. Steps of the faster R-CNN process

Faster R-CNN is a development of the previous methods called R-CNN and Fast R-CNN. R-CNN was first presented by Girshick *et al.* [18] in 2014 as an object detection method that uses a selective search algorithm to create around 2000 region proposals per image, then extracts features from each region proposal using a pre-trained convolutional neural network (CNN), and finally classifies the region proposals utilizing a support vector machine (SVM). After that, Girshick [19] introduced Fast R-CNN. The Fast R-CNN algorithm was developed to improve some of the shortcomings of R-CNN by allowing convolution to be performed only once on each image and using the resulting feature map to generate predicted region proposals, thus improving computational efficiency and object detection accuracy [20]. The algorithm was further improved to Faster R-CNN introduced by Ren *et al.* [20], utilizing an additional CNN called RPN to generate region proposals straight from vision features, thus stopping the need to use the selective search.

2.4.1. Residual network architecture as backbone

Faster R-CNN generally utilizes CNN architectures such as ResNet as a backbone to perform feature extraction, which is then used in the RPN and classification stages [12]. ResNet-50 has been widely

operated as a backbone in object detection, such as Faster R-CNN [21]. The ResNet architecture was formed to overcome the obstacles in deep learning training because, in general, it takes a long duration and is restricted to a specific number of layers [22]. Residual networks enhance deep network training with shortcuts that ease gradient flow, resulting in quicker training and increased accuracy of significant depth [23], as shown in Figure 5(a). The more representative the feature map extracted by the backbone, the better the performance of the detector, as a better feature map will help increase the accuracy level of object detection [24].

There are various versions of the Residual Networks architecture, including ResNet-50, ResNet-101, and ResNet-152, which have different levels of depth and complexity to meet specific needs. Slimene *et al.* [25] mentioned that the name of the ResNet architecture depends on the number of layers, as in ResNet50, ResNet101, and ResNet152, which indicate having 50, 101, and 152 layers. These layers are calculated from the total number of skip connections, convolutional layers, pooling layers, and fully connected layers that comprise the network, which is detailed in Figure 5(b).

The application of ResNet architecture to digital images has been used in various studies. As in the research of Sarwinda *et al.* [22], the detection of colorectal cancer was conducted using ResNet-50, which demonstrated an accuracy above 80% and a sensitivity above 87% on the 20% and 25% test sets. Another study by Zhang *et al.* [26] proposes a heartbeat classification method based on hybrid time-frequency analysis and transfer learning with ResNet-101, achieving 99.75% accuracy and an F1-score of 0.9016 for 14 types of heartbeats from the MIT-BIH database. This study by Goh *et al.* [27] evaluates deep transfer learning models for detecting four categories of face mask-wearing, with ResNet-152 achieving the highest accuracy of 86.67% on a testing set and 84.47% on smartphone-captured videos.

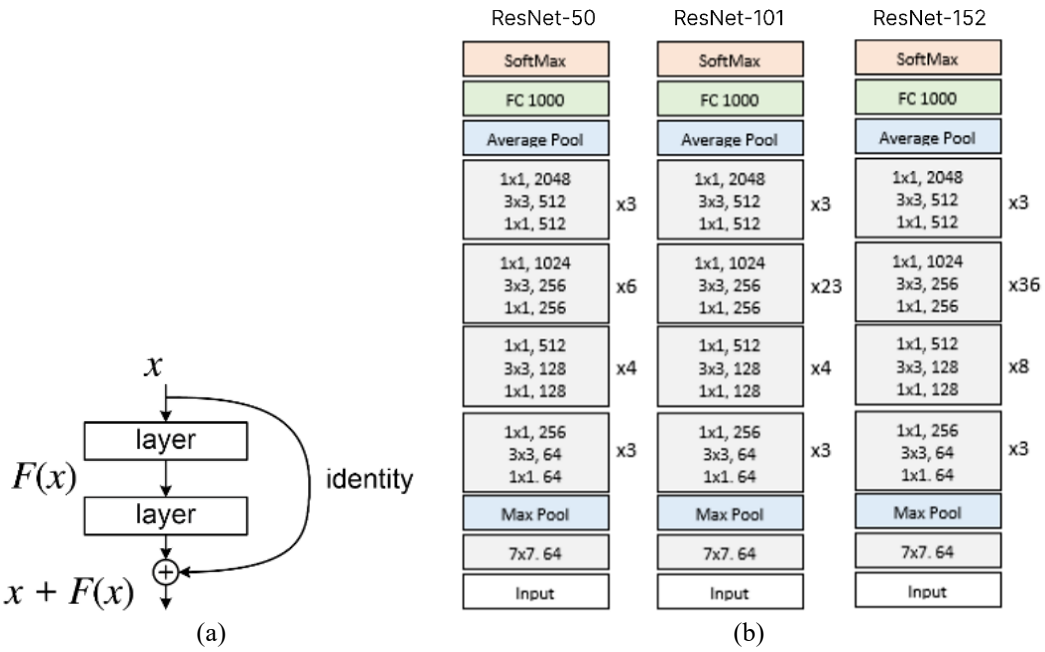


Figure 5. Residual network architecture (a) skip connections in ResNet and (b) network architecture of ResNet-50, ResNet-101, ResNet-152

2.4.2. Model configuration and training pipeline

This research will use ResNet-50, ResNet-101, and ResNet-152 as the backbone of the Faster R-CNN algorithm in SIBI recognition and will be compared to get the best architecture of the three models. The modeling was conducted using TensorFlow version 2.10 with built-in GPU support. Training was performed on a personal laptop equipped with a 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz CPU, NVIDIA GeForce RTX 3050 Laptop GPU, and 16 GB RAM. GPU acceleration was enabled, with an average GPU memory usage between 3 to 4 GB. To streamline the experiments and simplify the evaluation, the three models were integrated in a pipeline using TensorFlow tools. This pipeline is designed to automate the workflow from data pre-processing, model training, to result evaluation. The following is the configuration of the 3-model training pipeline used in this study, which is described in the Table 1.

Table 1. Configuration of training pipeline

Configuration	Value
Dimension size	320×320
Num of class	26
Input label map	"$/content/dataset/object-detection.pbtxt$"
Learning rate	Cosine Decay Learning Rate: base=0.04 warmup=0.013333 total steps=46800 warmup_steps: 2000
Data augmentation	Horizontal Flip (Probabilitas=0.3), Adjust Brightness (Delta maks=0.3)
Batch size	1
Num steps	46800 num steps (100 epoch)

2.5. Performance metrics

This study compares the performance of three different ResNet architectures (ResNet-50, ResNet-101, and ResNet-152) when used as the backbone for the Faster R-CNN algorithm. The goal is to detect Indonesian Sign Language (SIBI) signs in digital images. To evaluate each model's effectiveness, a confusion matrix will be used, which is a tool that compares the actual sign with the model's predicted sign [28]. A confusion matrix is a square matrix where the rows define the actual class of the instance, and the columns are the predicted class [29]. Identifying SIBI sign language consisting of 26 classes requires a multiclass confusion matrix. Confusion matrix for multiclass classification uses a $k \times k$ contingency table where cells $[i,j]$ $(i = 1,\dots,k, j = 1,\dots,k)$ represent the frequency of class observations with actual class Ki and predicted class Kj [30]. Table 2 illustrates an example of a confusion matrix for multiclass classification, where each cell shows the relationship between the actual and predicted classes.

In the confusion matrix for multiclass classification, there are 4 sections that show the results of the test as follows.

- True Positive (TP) refers to instances where the actual class is correctly predicted.
- True Negative (TN) is the actual class predicted to be true in the negative class.
- False Positive (FP) happens when a negative class is incorrectly predicted as positive.
- False Negative (FN) is when a positive class is incorrectly predicted as negative.

To measure the quality of a multiclass classification model, you can compare its architecture with a confusion matrix technique. This process generates accuracy, precision, recall, and the F1-score, which are all used to evaluate the model's performance.

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 * (recall * precision)}{recall + precision}$$

3. RESULTS AND DISCUSSION

3.1. Training model

The faster R-CNN model was trained 3 times with different architectures. Each model was trained for 100 epochs or 46.800 steps, with a total training time of approximately 4,783 seconds (47.83 seconds per epoch) for ResNet-50, 7530 seconds (75.30 seconds per epoch) for ResNet-101, and 11811 seconds (118.11 seconds per epoch) for ResNet-152. Model training uses loss as a measurement of the quality and performance of the model in making predictions on training data. Figure 6(a) to (c) represents a combination of loss in the box classifier and RPN for each architecture.

In the total loss graph of the Faster R-CNN model training above, it is known that the three models show a significant decrease in total loss until the 100th epoch. The results of the total loss of model training obtained a total loss value of 0.002 in the ResNet-50 model, 0.037 in the ResNet-101 model, and 0.0005 in

the ResNet-152 model. This shows that the model successfully converges and performs well in detecting objects in the training data.

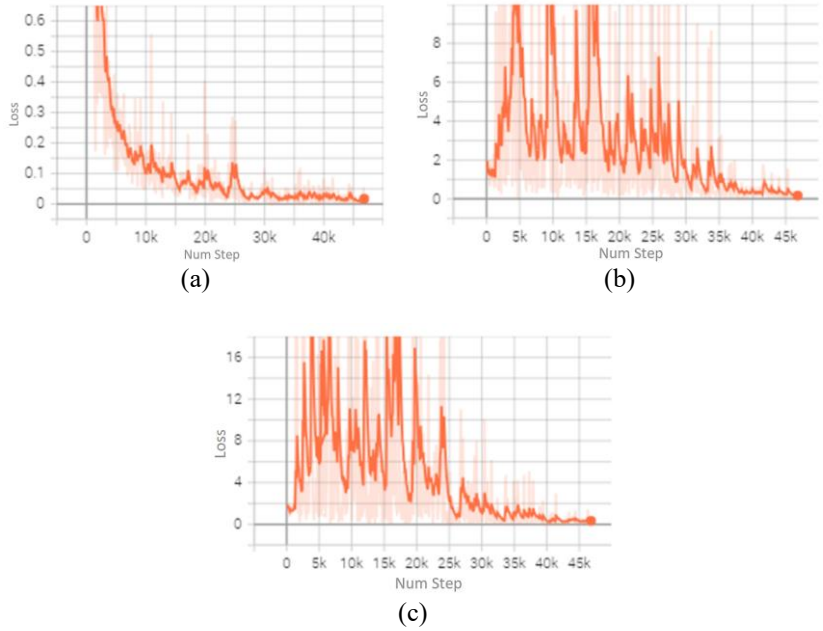


Figure 6. Total loss graph Faster R-CNN training (a) ResNet-50, (b) ResNet-101, and (c) ResNet-152

3.2. Testing model

Each model with three different architectures is then tested using test data. The test data consists of six data in each class, each of which contains one original data set and five different digital image augmentation results. Therefore, 156 data will be used for testing. Testing is done to see the model's performance in recognizing SIBI digital images. An example of model prediction result assessment during testing can be seen in Table 2.

Table 2. Model prediction results assessment during testing

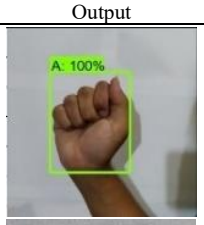
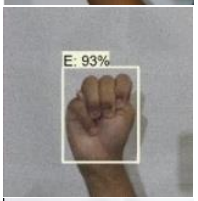
Output	Actual class	Prediction class	Result
	A	A	True
	M	E	False
	D	None	False

Figure 7(a) to (c) shows confusion matrix for each model used to calculate the evaluation value: ResNet-50, ResNet-101, and ResNet-152. These results are in the form of a multiclass confusion matrix with 27 rows and 27 columns. The row value in the confusion matrix shows the actual class, while the column value shows the model's prediction. Classes consist of the letters A to Z and plus the None class, which indicates that the model does not recognize any objects in the input image.

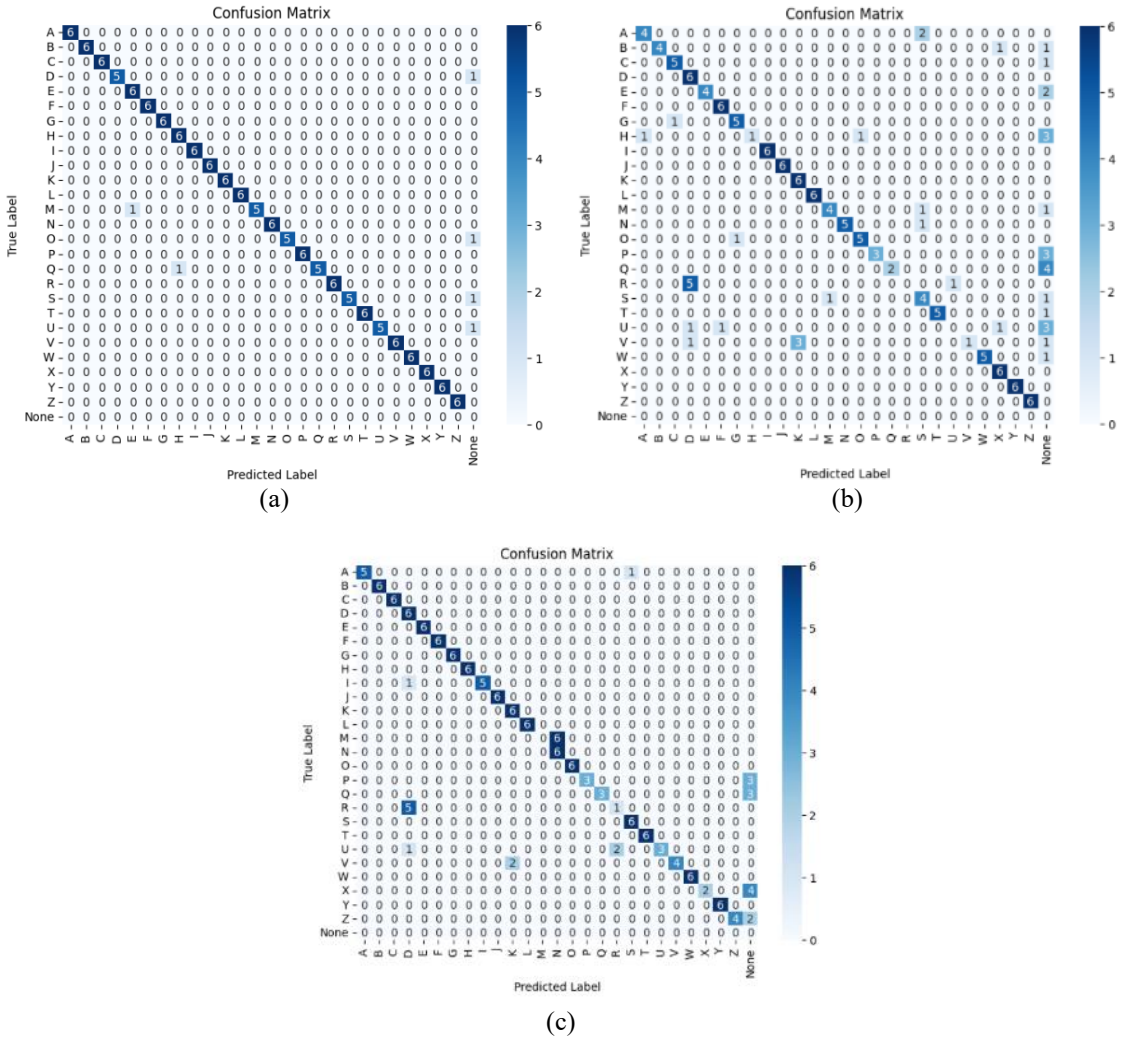


Figure 7. Confusion matrix of test results (a) ResNet-50, (b) ResNet-101, and (c) ResNet-152

3.3. Performance

Confusion matrix from the test results is calculated to obtain the model performance evaluation value. The evaluation metrics used are accuracy, precision, recall, and F1-score. Table 3 presents the evaluation metrics for all three models.

Based on Table 3, architectures in the Faster R-CNN algorithm for recognizing SIBI, it is known that the performance of the three models differs significantly. ResNet-50 outperformed ResNet-101 and ResNet-152 in both accuracy and efficiency, processing faster and delivering superior results. ResNet-101 showed the lowest performance, while ResNet-152 improved upon it but still fell short of ResNet-50.

Table 3. Performance evaluation of the models

Model Architecture	Accuracy	Precision	Recall	F1-Score	Time (second)
ResNet-50	96%	95%	93%	94%	36.84
ResNet-101	71%	79%	69%	70%	38.82
ResNet-152	81%	85%	78%	79%	40.20

3.3.1. ResNet-50

The Faster R-CNN algorithm with ResNet-50 architecture as the backbone has reliable performance in detecting each class in 26 letters of the Indonesian sign language system (SIBI). This is shown through the evaluation results where the model is able to predict test data with metric values of accuracy 96%, precision 95%, recall 93% and F1-score 94%. In terms of test execution time, it can be seen that the model with ResNet-50 architecture is superior to the models with ResNet-101 and ResNet-152 architectures. The simple architectural layers in ResNet-50 provide the benefit of less resource utilization.

The lower architecture complexity in ResNet-50 allows the model to learn essential features without overfitting, especially on data with a limited variety and amount of data. However, testing on blurred digital image scenarios such as classes D, O, S, and U were identified less accurately and were not recognized as objects by RPN. This is because the object in the image is too blurred to be recognized by the model.

3.3.2. ResNet-101

The Faster R-CNN algorithm with ResNet-101 architecture shows less than optimal performance with metrics accuracy of 71%, precision of 79%, recall of 69%, and F1-score of 70%. Of the 26 existing classes, this model can only accurately predict classes in 9 classes, namely classes D, F, I, J, K, L, X, Y, and Z. In fact, there is class data that cannot be predicted. There are even class data that cannot be predicted correctly, which are the R and U class data.

The execution time of the ResNet-101 architecture model is 38.82 seconds. This shows that this model is longer than the ResNet-50 architecture model, with an increase in test execution time of 5.37%. This is due to the higher complexity of the ResNet-101 architecture compared to ResNet-50, which results in greater resource utilization and computational processes.

The lack of optimization of the model with ResNet-101 architecture is due to the overfitting of the model to the limited training data. In addition, it appears that the variations in the test data are poorly recognized by the model, indicating that the model cannot generalize well to variations not present in the training data. Another factor contributing to this low performance is the lack of training data as the complexity of the ResNet-101 architecture requires more data for effective training.

3.3.3. ResNet-152

Feature extraction on ResNet-152 is the most complex compared to ResNet-50 and ResNet-101. The high complexity of the architecture makes the test execution time of this model the longest compared to the other two models. Based on the research, the execution time takes 40.20 seconds for the testing process. The ResNet-152 architecture model is good in the testing process, with an accuracy metric value of 80.75%, precision of 85%, recall of 78%, and F1-score of 79%. The evaluation values show that this model is better at predicting test data than the ResNet-101 architecture model but lower than the ResNet-50.

The model quality with ResNet-152 architecture tends to experience overfitting, similar to the ResNet-101 architecture model. This is because the small amount of data used in training makes it less effective for models with high complexity, such as ResNet-152, to generalize the data. As a result, this model performs very well in training but less optimally in testing with new data variations.

4. CONCLUSION

This research successfully implements the ResNet-50, ResNet-101, and ResNet-152 architectures in the Faster R-CNN algorithm to recognize the Indonesian language sign system (SIBI). This can be proven through the smooth training process until it reaches the final epoch and the entire model experiences a consistent decrease in loss in the training process. From the evaluation results, the Faster R-CNN model with ResNet-50 architecture showed the best and most efficient performance with an accuracy value of 96.15% and an execution time of 36.84 seconds in the testing process. Therefore, ResNet-50 was chosen as the best backbone/feature extraction architecture in the Faster R-CNN algorithm for recognizing the SIBI. This study contributes to the field by empirical evaluation of backbone selection based on detection performance, inference time, and model complexity under limited data conditions, providing practical insights for real-world deployment.

Based on this research, ResNet-50 is recommended as the top choice for the Faster R-CNN algorithm's backbone in applications designed to recognize Indonesian sign language (SIBI). These applications could be mobile apps or communication devices that help bridge the communication gap between the deaf community and the general public, promoting more inclusive interactions. For future research, it's suggested to use larger datasets and experiment with different Faster R-CNN configurations. The current findings lay a strong foundation for building SIBI recognition systems that can be integrated into assistive technologies. While this study's prototype was tested in a controlled environment, future work should focus on real-time deployment and usability studies, including implementation on mobile or embedded platforms for practical, real-world use.

REFERENCES

[1] W. Aly, S. Aly, and S. Almotairi, "User-independent american sign language alphabet recognition based on depth image and PCANet features," *IEEE Access*, vol. 7, pp. 123138–123150, 2019, doi: 10.1109/ACCESS.2019.2938829.

[2] Suharjito, N. Thiracitta, and H. Gunawan, "SIBI sign language recognition using convolutional neural network combined with transfer learning and non-trainable parameters," *Procedia Computer Science*, vol. 179, no. 2019, pp. 72–80, 2021, doi: 10.1016/j.procs.2020.12.011.

[3] E. Rakun, I. G. B. H. Widhinugraha, and N. F. Putra Setyono, "Word recognition and automated epenthesis removal for Indonesian sign system sentence gestures," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 26, no. 3, pp. 1402-1414, June 2022, doi: 10.11591/ijeecs.v26.i3.pp1402-1414.

[4] A. R. Syulistyo, D. S. Hormansyah, and P. Y. Saputra, "SIBI (Sistem Isyarat Bahasa Indonesia) translation using convolutional neural network (CNN)," *IOP Conference Series: Materials Science and Engineering*, vol. 732, no. 1, 2020, doi: 10.1088/1757-899X/732/1/012082.

[5] P. Sharma, A. Ranjan, A. Shekhawat, and M. Raj, "A novel approach for identification of sign languages using deep learning," vol. 5, no. 4, pp. 1284–1288, 2023, doi: 10.35629/5252-050412841288.

[6] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2351–2377, 2022, doi: 10.1007/s10462-021-10066-4.

[7] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications*, vol. 172, no. April 2020, p. 114602, 2021, doi: 10.1016/j.eswa.2021.114602.

[8] L. Deng, S. Liu, Y. Cheng, G. Zhao, and J. Xu, "Algorithm for diabetic retinal image analysis based on deep learning," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 47559–47584, 2023, doi: 10.1007/s11042-023-15503-w.

[9] M. F. S. Sabir *et al.*, "An automated real-time face mask detection system using transfer learning with Faster-RCNN in the era of the COVID-19 pandemic," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 4151–4166, 2022, doi: 10.32604/cmc.2022.017865.

[10] C. Cao *et al.*, "An improved Faster R-CNN for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019.2932731.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[12] D. Avola *et al.*, "MS-faster R-CNN: Multi-stream backbone for improved faster R-CNN object detection and aerial tracking from UAV images," *Remote Sensing*, vol. 13, no. 9, pp. 1–18, 2021, doi: 10.3390/rs13091670.

[13] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," *Proceedings - International Conference on Pattern Recognition*, pp. 9415–9422, 2020, doi: 10.1109/ICPR48806.2021.9412193.

[14] D. D. Aboyomi and C. Daniel, "A comparative analysis of modern object detection algorithms: YOLO vs. SSD vs. Faster R-CNN," *ITEJ (Information Technology Engineering Journals)*, vol. 8, no. 2, pp. 96–106, 2023, doi: 10.24235/itej.v8i2.123.

[15] L. Afkaar, "Datasets SIBI sign language alphabets," *Kaggle*, 2021. https://www.kaggle.com/datasets/mlanangafkaar/datasets-lemlitbang-sibi-alphabets (accessed May 30, 2024).

[16] M. Maity, S. Banerjee, and S. Sinha Chaudhuri, "Faster R-CNN and YOLO based vehicle detection: A survey," *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, no. April, pp. 1442–1447, 2021, doi: 10.1109/ICCMC51019.2021.9418274.

[17] Y. Su, D. Li, and X. Chen, "Lung nodule detection based on Faster R-CNN framework," *Computer Methods and Programs in Biomedicine*, vol. 200, p. 105866, 2021, doi: 10.1016/j.cmpb.2020.105866.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.

[19] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2015, vol. 2015 Inter, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[20] M. M. Mijwil, K. Aggarwal, R. Doshi, K. K. Hiran, and M. Gök, "The distinction between R-CNN and Fast R-CNN in image analysis: A performance comparison," *Asian Journal of Applied Sciences*, vol. 10, no. 5, 2022, doi: 10.24203/ajas.v10i5.7064.

[21] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," *arXiv preprint arXiv:1804.06215*, pp. 1–17, 2018.

[22] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer," *Procedia Computer Science*, vol. 179, no. 2019, pp. 423–431, 2021, doi: 10.1016/j.procs.2021.01.025.

[23] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An efficient brain tumor image segmentation based on deep residual networks (ResNets)," *Journal of King Saud University - Engineering Sciences*, vol. 33, no. 6, pp. 404–412, 2021, doi: 10.1016/j.jksues.2020.06.001.

[24] T. Liang *et al.*, "CBNet: A composite backbone network architecture for object detection," *IEEE Transactions on Image Processing*, vol. 31, pp. 6893–6906, 2022, doi: 10.1109/TIP.2022.3216771.

[25] I. Slimene, I. Messaoudi, A. E. Oueslati, and Z. Lachiri, "Cancer disease multinomial classification using transfer learning and SVM on the genes' sequences," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 9, no. 1, pp. 1–11, 2023, doi: 10.4108/eetpht.9.3220.

[26] Y. Zhang, J. Li, S. Wei, F. Zhou, and D. Li, "Heartbeats classification using hybrid time-frequency analysis and transfer learning based on ResNet," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 11, pp. 4175–4184, 2021, doi: 10.1109/JBHI.2021.3085318.

[27] P.-J. Goh, M.-H. Hoo, and K.-C. Khor, "Evaluating deep transfer learning models for detecting various face mask wearings," in *International Conference on Soft Computing and Data Mining*, 2024, pp. 43–52.

[28] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jul. 2020, doi: 10.1016/j.aci.2018.08.003.

[29] O. Caelen, "A Bayesian interpretation of the confusion matrix," *Annals of Mathematics and Artificial Intelligence*, vol. 81, no. 3–4, pp. 429–450, 2017, doi: 10.1007/s10472-017-9564-8.

[30] S. Ruuska, W. Hämäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behavioural Processes*, vol. 148, pp. 56–62, 2018, doi: 10.1016/j.beproc.2018.01.004.

BIOGRAPHIES OF AUTHORS

Paulus Lestyo Adhiatma is a student in the informatics and computer engineering education program at Sebelas Maret University. He graduated from SMK Negeri 2 Surakarta with a major in software engineering (RPL) in 2020. He won third place in the web design competition with his team and was a finalist in the Capture the Flag Joints 2019 competition. In 2022, he interned at the information and communication technology center (UPT TIK) of UNS for one year. He has skills in backend development and an interest in machine learning, particularly in computer vision. He is also a participant in the MBKM Bangkit program by Google, Tokopedia, Gojek, and Traveloka in the machine learning path in 2023. Additionally, he co-authored chapter 5 of the book *Sistem Rekomendasi: Konsep dan Implementasi* with three lecturers. He can be contacted at paulus.lestyo@student.uns.ac.id.

Nurcahya Pradana Taufik Prakisya is a lecturer in informatics engineering at Universitas Sebelas Maret's faculty of teacher training and education. In addition to being a teacher, he enjoys developing websites and exploring artificial intelligence. He graduated from Universitas Sebelas Maret's undergraduate degree in informatics in 2013 and from Universitas Gadjah Mada with a master's degree in computer science in 2017. He holds national and international certifications in web app development fundamentals from Microsoft Technology Associate and programming and software development with the competency in program analyst from BNSP. His main research interest is computer vision in clinical pathology images. His study has been published in many leading journals such as Open Engineering and International Journal on Advanced Science, Engineering, and Information Technology. He can be contacted at email: nurcahya.ptp@staff.uns.ac.id.

Rosihan Ariyuana obtained a mathematics bachelor's degree from Universitas Sebelas Maret in 2001. He graduated with a master's degree in computer science in 2004 from Gadjah Mada University. He currently teaches databases, object-oriented programming, and structured programming at the Department of Informatics Education for bachelor's degree candidates. Artificial intelligence, computational thinking, and computer-assisted learning are his areas of interest in research. He can be contacted at email: rosihanari@staff.uns.ac.id.