JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Performance Comparison Between ResNet50 and MobileNetV2 for Indonesian Sign Language Classification

Feriska Putri Daviana, Aryanti*, Nurhajar Anugraha

Jurusan Teknik Elektro, Teknik Telekomunikasi, Politeknik Negeri Sriwijaya, Palembang, Indonesia
Email: [1]frsptr17@gmail.com, [2,*] aryanti@polsri.ac.id, [3]nurhajar.anugraha@polsri.ac.id
Email Penulis Korespondensi: aryanti@polsri.ac.id
Submitted **25-05-2025;** Accepted **29-06-2025;** Published **30-06-2025**

Abstract

Hearing impairment was considered a significant barrier to understanding verbal communication. Therefore, an alternative communication medium in the form of sign language was required to bridge interactions between Deaf and hearing individuals. One of the sign languages used in Indonesia was the Indonesian Sign Language (BISINDO). The advancement of deep learning technology provided a great opportunity to develop an effective and accurate BISINDO alphabet classification system. This research was conducted to evaluate and compare the performance of two Convolutional Neural Network (CNN) architectures, namely ResNet50 and MobileNetV2, in classifying BISINDO alphabet images consisting of 26 classes from A to Z. Model training wa carried out over 100 epochs and was analyzed using metrics such as training and validation accuracy, precision, recall, F1-score, and confusion matrix. The training process used a dataset that was divided into 80% training data and 20% validation data, and include image preprocessing steps such as resizing and rescaling. The evaluation results showed that ResNet50 achieved 86.42% training accuracy and 98.64% validation accuracy with 98.80% precision, 98.69% recall, 98.57% F1-score, and 31 misclassifications. In contrast, MobileNetV2 showed superior performance with 99.99% training accuracy, 99.65% validation accuracy, 99.69% precision, 99.65% recall, 99.61% F1-score, and only 8 misclassifications. Based on these results, MobileNetV2 was recommended as a more effective and efficient architecture for BISINDO alphabet image classification compared to ResNet50.

Keywords: Indonesian Sign Language; Image Classification; Convolutional Neural Network; ResNet50; MobileNetV2

1. INTRODUCTION

Someone who has had physical problems since birth can yet live a normal life. Hearing loss is one of these restrictions. It makes it hard for people to understand spoken conversations and talk to each other directly. So, to make sure that information gets to people smoothly, there needs to be a way for both deaf and hearing friends to talk to each other. Sign language is a crucial way for deaf friends to communicate their thoughts, feelings, and information through movements that can be seen. Sign language is more than simply a way to talk to people; it's also a way to express yourself and connect with others.

Indonesian Sign Language (BISINDO) is now the main way for deaf and mute people in Indonesia to talk to each other. The research [1] shows that using BISINDO sign language is better for deaf friends in the Madrasah Ibtidaiyah Teacher Education Program at Sunan Kalijaga State Islamic University. There are a few reasons for this, such as the fact that BISINDO is easier to comprehend, is a natural language spoken by deaf friends, is easy to show, and is more effective and expressive. But research [2] shows that hearing friends and deaf friends often have trouble getting along since they don't grasp BISINDO. This difference makes it necessary to use technology-based advances to make communication more open to everyone.

Deep learning is one area of artificial intelligence (AI) that has made a lot of progress. These advances open up a lot of possibilities for creating sign language classification systems that are both fast and accurate. One of the most prominent deep learning algorithms is the Convolutional Neural Network (CNN), which was made to interact with visual input like pictures. CNN employs pooling layers to make the input smaller without sacrificing crucial information and convolutional layers to find local features like edges and textures. CNN is particularly good at many things, such finding objects, recognizing faces, and segmenting objects, because of these two layers working together [3]. With this technique, computers can learn to detect hand motions in BISINDO. This makes it easier for people to understand what their deaf friends are saying.

A lot of research has been done to make sign language recognition models employing deep learning because of this possible. The research [4] compares the Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) methods for sorting photographs of the American Sign Language (ASL) alphabet, with the exception of the letters "J" and "Z." The CNN with Gaussian Low Pass Filter preprocessing had the best accuracy (96.93%) and F1-score (96.97%), showing that the convolution and pooling stages are better at recognizing sign language images.

Research [5] uses the Convolutional Neural Network (CNN) method to compare the LeNet and AlexNet systems for finding the American Sign Language (ASL) alphabet. Twenty-one hundred pictures per letter and one thousand photos per letter were used in the tests. We can see that the LeNet design is the best in the research because it has the highest accuracy (92.468) on the 1,000-character data scheme.

The research [6] used Computer Vision technology built on Convolutional Neural Network (CNN) to make a translator tool for the Indonesian Sign Language System (SIBI). The system was tested in a range of lighting conditions and at various distances using data from 29 different hand movements. The data was first processed by converting it to

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

HSV, cropping it, and setting a boundary. This method seems to have worked 97.2% of the time, which shows that CNN is good at detecting hand movements.

Researchers [7] used finger movements to make a system for deaf people to learn hijaiyah letters based on Convolutional Neural Network (CNN) picture recognition. MnetV2, VGG16, ResNet50, and Xception are four models that have already been trained. Only 82.1% of the time did ResNet50 get it right, while MnetV2, VGG16, and Xception got up to 99.85% of the time. It was the MnetV2 model that could recognize fingerprints in real time the best.

In the meanwhile, the research [8] put the letters of the Indonesian Sign Language System (SIBI) into 24 groups, leaving out the letters J and Z. We looked at three different types of Convolutional Neural Networks (CNN): MobileNetV2, MobileNetV3Small, and MobileNetV3Large. After training for 30 epochs with a batch size of 32, the findings show that MobileNetV3Small has the best performance, with an accuracy of 98.81%.

Many Convolutional Neural Network (CNN) types have been worked on. ResNet50 and MobileNetV2 are two that stand out because of the way they are built. ResNet50 solves the problems that the vanishing gradient causes while training deep networks by adding shortcut connections. This makes the training process more efficient and keeps the model's performance high even with a lot of layers [9] [10]. MobileNetV2, on the other hand, is designed to work well on smartphones with limited resources . It uses inverted residual blocks and depthwise separable convolution. These two methods work together to keep the model correct while using less processing power, which makes it perfect for real-time mobile apps [11].

Many studies have examined how to recognize sign language using CNN, but most focus on ASL or SIBI alphabets instead of BISINDO. Also, there hasn't been any research that directly compares how well ResNet50 and MobileNetV2 work for classifying the BISINDO alphabet, even though they have different architectures and have been shown to work well in other areas. This is a research gap that needs to be filled.

This study aims to compare how well ResNet50 and MobileNetV2 can recognize the BISINDO alphabet using metrics like accuracy, precision, recall, F1-score, and a confusion matrix. The main goal is to test how well the model works, not to directly build an application. This study's main contribution is that it fully compares two popular CNN architectures in the context of BISINDO classification. This information will be useful for developers of technology-based communication systems, as it will help make communication between deaf people and the general public more accessible.

2. RESEARCH METHODOLOGY

2.1 Research Stages

The first step in the research procedure is to acquire hand photographs that depict the BISINDO alphabet, as shown in Figure 1. The next step is preprocessing, which involves resizing, rescaling, and separating the data into training and validation sets. Then, two CNN architectures, ResNet50 and MobileNetV2, train the data to sort the alphabet based on visual patterns. We next used accuracy, precision, recall, F1-score, and confusion matrix metrics to see how effectively both topologies identified the BISINDO alphabet. Figure 1 shows the several structured steps that were taken to do this research.

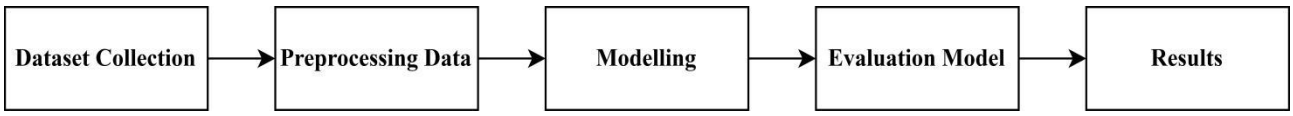
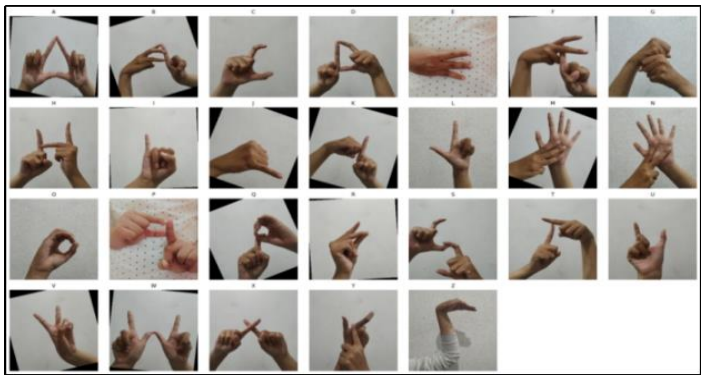


Figure 1. Research Methods

2.2 Dataset Collection

The Kaggle.com website operated by Agung Ma'ruf has the Indonesian Sign Language – BISINDO dataset that this research uses. This collection has 11.471 pictures of the BISINDO alphabet, from letters A to Z, in 26 classifications. Figure 2 shows examples of the dataset for each type.



JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Figure 2. Example of BISINDO Dataset

2.3 Preprocessing Data

This is the process when the images are resized, rescaled, and broken into smaller pieces. Resizing the photos alters their size from 640 × 640 pixels to 224 x 224 pixels. This is done to make sure that the ResNet50 and MobileNetV2 models used in the modeling have input layers that are the same size. The goal of this resizing is to make sure that the input data is all the same, so that the model can handle the images in the same way every time and fully support the training [12]. Rescaling, which is changing the pixel values of an image from 0–255 to 0–1, is another thing that is done. The rescale function makes the pixel value scale smaller and more even, which helps the model analyze the input faster and do a better job [13]. Next, the dataset is split into two groups: 80% of the training data, which is made up of 9,186 photos. We utilize this training data to create a model that can learn patterns from the data we have and then use those patterns to guess the class of fresh data that we haven't seen before [14]. And 20% of the data for validation, which includes 2,285 photos. This is done to test how well the model works on data that wasn't used in training. This is done to make sure that the model doesn't overfit and can still generalize well when it sees new data [15].

2.4 Modeling

This research uses two types of CNN, ResNet50 and MobileNetV2. A transfer learning method is applied with both architectures. This method uses a model already trained on the ImageNet dataset to provide it starting weights. This makes the training process go more quickly. Transfer learning lets you model new tasks using what you've learned from past training, so you don't have to retrain on a lot of data [16] [17].

ResNet50 is one of the CNN architectures used in this study. It is known for solving the problem of vanishing gradients in very deep networks. This model uses residula learning and shortcut connections to deepen network training. Figure 3 below visually presents the architecture of ResNet50 used in this study.

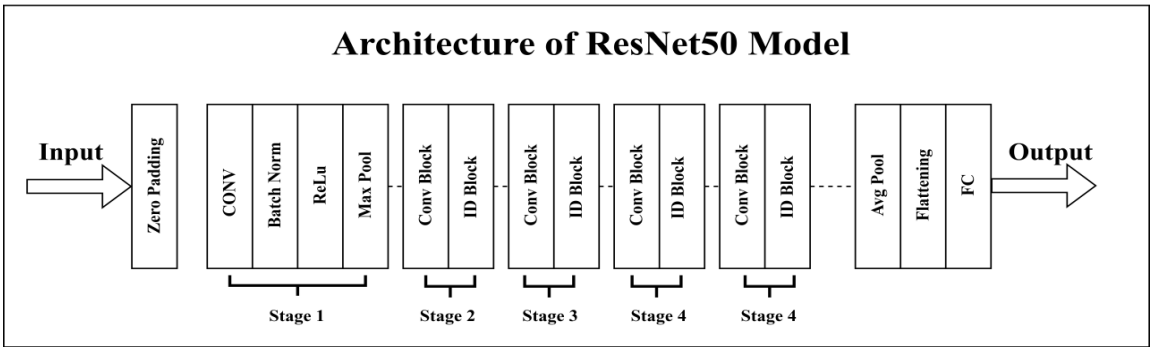
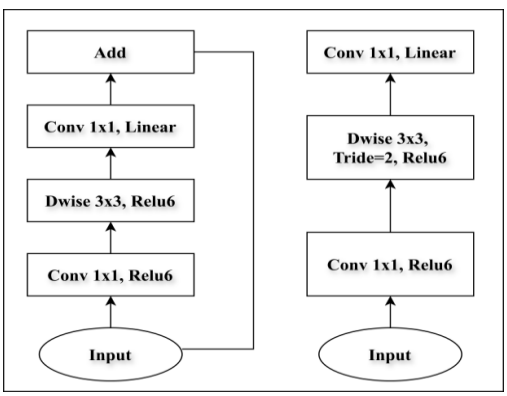


Figure 3. ResNet50 Architecture

ResNet50 consists of 50 layers and is pre-trained on the ImageNet dataset, which contains over a million images. The architecture has five main stages (Stages 1-5). The first step is Zero Padding, followed by Batch Normalization, ReLU activation, and Max Pooling. These steps stabilize the data and reduce the number of features. Stages 2 to 5 consist of both a Convolutional Block and an Identity Block, each with three convolutional layers. The Average Pooling, Flattening, and Fully Connected (FC) layer processes at the end of these convolution stages make the final classification output. More than 23 million parameters in this architecture can be changed during training. The model uses transfer learning to speed up and make the training process on new datasets more accurate by using the initial weights from training on ImageNet. This lets the model learn from features it already knows, so it doesn't need a lot of new data to start with [18].

Inverted residual blocks are the main building blocks of MobileNetV2. Depending on whether residual connections are used, these blocks can be structured in two ways. Figure 4 shows the two main types of block structures that make MobileNetV2 so good at extracting features with very few resources.



JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Figure 4. MobileNetV2 Architecture

The block with leftover connections (shortcuts) on the left side of Figure 4 is used when the input and output dimensions are the same. This block is made up of, 1x1 convolution layer with ReLU6 activation to make the dimension bigger (expansion), ReLU6 with a 3x3 depthwise convolution layer (Dwise 3x3), 1x1 convolution layer with Linear activation (not with non-linear activation), Added a direct shortcut connection from the input to the output through the addition operation (Add). This method allows MobileNetV2 to balance performance and computational efficiency well, making it a good choice for mobile devices. The whole MobileNetV2 architecture used in this study for BISINDO motion classification is built on these layers [19] [20].

We took out the last classification layer and put in a new one that was made for the 26 classes in the BISINDO dataset. This changed the architecture of both models. The new layer has one hidden layer with 512 neurons that use the ReLU activation function to make it non-linear, and one output layer with 26 neurons that use the Softmax activation function, which is good for classification tasks. During training, the weights don't change because the first layers of both models are locked (frozen). So, only the component that sorts things out is retrained. This strategy lets the model keep learning from the ImageNet dataset while also adjusting to the features of the BISINDO data unit.

2.5 Evaluation Model

Model evaluation is used to check how well the deep learning model works once the training phase is over. The objective of this evaluation is to find out how well the model can apply what it has learned to new data and to find problems like underfitting or overfitting [21]. Here are the metrics utilized for evaluation:

a. Accuracy
To find out how accurate something is, you compare the number of correct predictions with the entire amount of data that was looked at.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. Precision
Precision tells us how accurate the predictions are in a certain category by showing us how much of the projected data is actually useful for that category.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

c. Recall
Recall is the ratio of correctly classified data to all the actual data that corresponds to a category. It shows how well the system can discover all the relevant data in that category.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

d. F1-score
The F1-score shows how effectively the model classifies data accurately and completely by measuring the balance between precision and recall.

$$\text{F1-score} = 2\,\frac{\text{Precision x Recall}}{\text{Precision + Recall}} \quad (4)$$

e. Confusion matrix
One way to show how well the model classification works is with the confusion matrix. This matrix shows how well the model worked by comparing the predicted results to the real conditions. The basic shape of the Confusion Matrix used in this study is shown in Figure 5 below.

		True / Actual Class	
		Positive (P)	Negative (N)
Predicted Class	True (T)	True Positive (TP)	False Positive (FP)
	False (F)	False Negative (FN)	True Negative (TN)

Figure 5. Confusion Matrix

In a confusion matrix, there are four possible results for a prediction, True Positive (TP), the model says it's a positive class, and it really is. True Negative (TN), the model says that the class is negative, and it is negative. False Positive (FP), the model says that a class is positive when it is negative, this is also called Type I Error. The model says the class is negative when it is really positive (also called Type II Error). The red boxes in Figure 5 show correct predictions (TP and TN), and the blue boxes show wrong predictions (FP and FN). We can use this confusion matrix to determine different evaluation metrics, like accuracy, precision, recall, and F1-score, which tell us how well the classification model works [22].

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

3. RESULT AND DISCUSSION

This part shows the outcomes of preprocessing, training, and testing two Convolutional Neural Network (CNN) architecture models that were used to sort the Indonesian Sign Language alphabet. ResNet50 and MobileNetV2 are two of these models. We used Jupyter Notebook, which runs in the Visual Studio Code (VSCode) environment, throughout training and evaluation to make it easier to write and supervise code.

3.1 Data Preprocessing Results

A pre-processing step is performed on the images before they are put into the model training process. This ensures that all of the images have the same size and pixel scale. This step includes resizing the image to change its size and rescaling it to change the range of pixel values. The difference in how the image looks at each stage of pre-processing is shown in Figure 6 below.

Figure 6. Results of Data Preprocessing

The left picture shows the original picture before pre-processing, when the pixel values' size and range were not changed. The left picture shows the original picture before pre-processing when the pixel values' size and range were not changed. In the right image, the pixel values that were originally between 0 and 255 have been changed to between 0 and 1 after resizing and rescaling. The goal is to speed up the process of convergence during training while keeping the calculations in the neural network stable.

3.2 Training Results

This part shows the outcomes of training two CNN architecture models, ResNet50 and MobileNetV2, to recognize BISINDO. To make sure the model can generalize, the dataset has been split into training and validation data. Table 1 shows the hyperparameter parameters that were used to train the model for 100 epochs. We check how well the model is learning and look for signs of overfitting by looking at accuracy and loss graphs on both training and validation data during the training process. Training and validation data are used to check how well learning is working and to see if overfitting might be happening. The results from this stage are used to judge how well the model works overall.

Tabel 1. Hyperparameter Model

Hyperparameter	Score
Optimizer	Adam
Learning rate	0.001
Loss function	Categorical Cross Entropy
Batch size	32

a. ResNet50

We trained the ResNet50 model on BISINDO data to see how well it could learn patterns. During this step, the model's performance is measured on the training and validation data using two key metrics: accuracy and loss value. The graph in Figure 7 below demonstrates how the accuracy values and loss have changed over time. This can give us an idea of how stable and successful the model's learning process is. This visualization will help you figure out if the model's performance is becoming better over time or if it is showing signs of overfitting during training.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

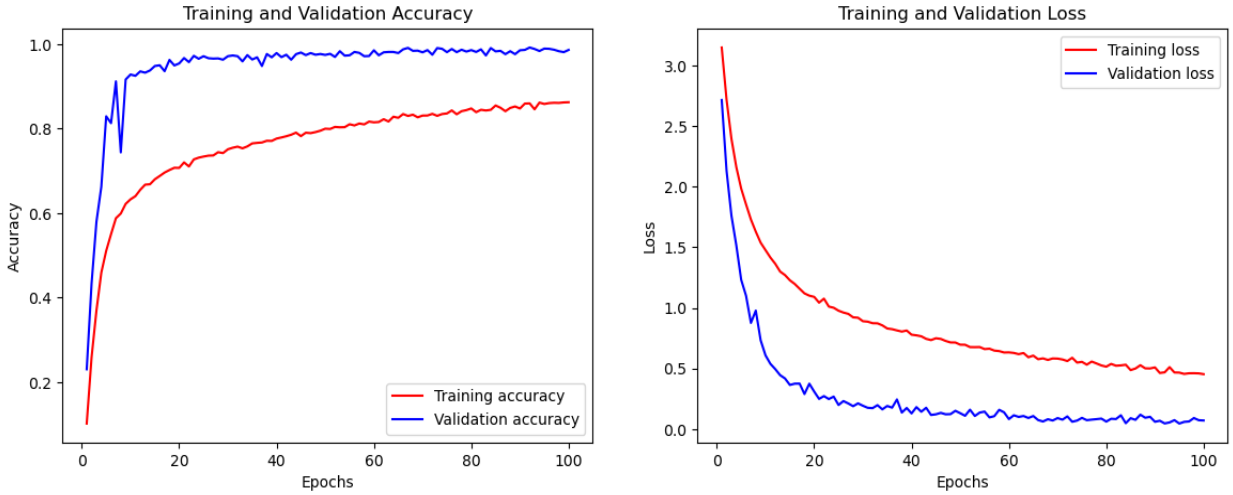


Figure 7. ResNet50 Accuracy and Loss Graph

The ResNet50 model's performance improved a lot throughout the course of 100 epochs of training. Figure 5 shows that the model wasn't very good at recognizing data patterns in the first epoch. The training accuracy is still quite poor, at about 6.66%, which shows that the neural network weights are still at their original levels, which is not good for the early stages. The model is still trying to figure out how the features and the data structure are related.

The loss value is also fairly high: 3.2792 for the training data and 2.7138 for the validation data. The model's predictions don't match the real labels, which is why the loss value is so large. This makes sense at the beginning of training a deep neural network, especially for complicated architectures like ResNet50, because the learning process is still in the exploratory phase. The model doesn't know enough to make reliable predictions, therefore there are big mistakes in both the trained data and the data it hasn't seen yet.

But as the epochs go up, the model's performance keeps getting better and better. During each training epoch, the model can change its weights based on the outcomes of backward propagation and updates utilizing optimization techniques like Adam. This technique slowly lowers the loss value and raises the accuracy. The sixteenth period was when performance improvement started to show. The accuracy of the training went up to about 67%, which is a big jump from the first phase. The current validation accuracy is 94.8%, which is even more fascinating because it shows that the model is starting to generalize data that wasn't in the training set.

At this point, the loss value also went down a lot. The validation loss has dropped a lot, which means that the model's predictions on new data are getting more and more accurate. This is an important signal in deep learning model training because overfitting commonly happens when the training accuracy goes up but the validation accuracy goes down. But in this case, the training and validation data both went up by the same amount, which means that the ResNet50 model trained well without overfitting too much at that point.

The model's performance got more steady after the 25 epoch. At this point, the training loss value went down to 0.9688 and the validation loss value went down to 0.1997. This means that the model is not just memorizing the training data, but also understanding how the images in the dataset are represented by their features. This means that the model's prediction error has gone down a lot. The model finds essential patterns that set classes apart, generates better predictions, and makes good representations of internal data.

The performance kept getting better until the 100 epoch. The model has done really well thus far. The model has accurately predicted most of the training data because it has a validation accuracy of 98.64% and a training accuracy of 86.42%. This capacity to generalize is even more significant for building classification models because it tells you how well the model can be used on real-world data that is different from the training data.

The loss value also went down and stayed the same in the 100 epoch. The training loss was 0.4442, and the validation loss went down a lot to 0.0719. The validation loss value is quite low, which means that the model is very accurate and makes very few mistakes when predicting new data. The ResNet50 model has converged well since the validation loss goes down while the accuracy goes up. The model has finished learning, therefore it is now in the best possible shape to do the classification job.

Training ResNet50 for 100 epochs indicates that deep learning models can work if they have an organized way of training, a strong architecture, and enough training data. ResNet50 slowly learns to spot relevant features in the picture data it is trained on, which greatly improves the accuracy of classification and keeps performance consistent without overfitting. So, the training results suggest that ResNet50 is a good model for producing accurate predictions after going through a deep learning procedure.

b. MobileNetV2

We trained MobileNetV2 for 100 epochs to see how well it learns data over time and how it gets better at accuracy and loss during training. The graph in Figure 8 shows how the model's accuracy and loss change over time on the training

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

and validation data. These changes reflect how well the model can recognize patterns in data it has already seen and how well it can test its skill on data it has never seen before.

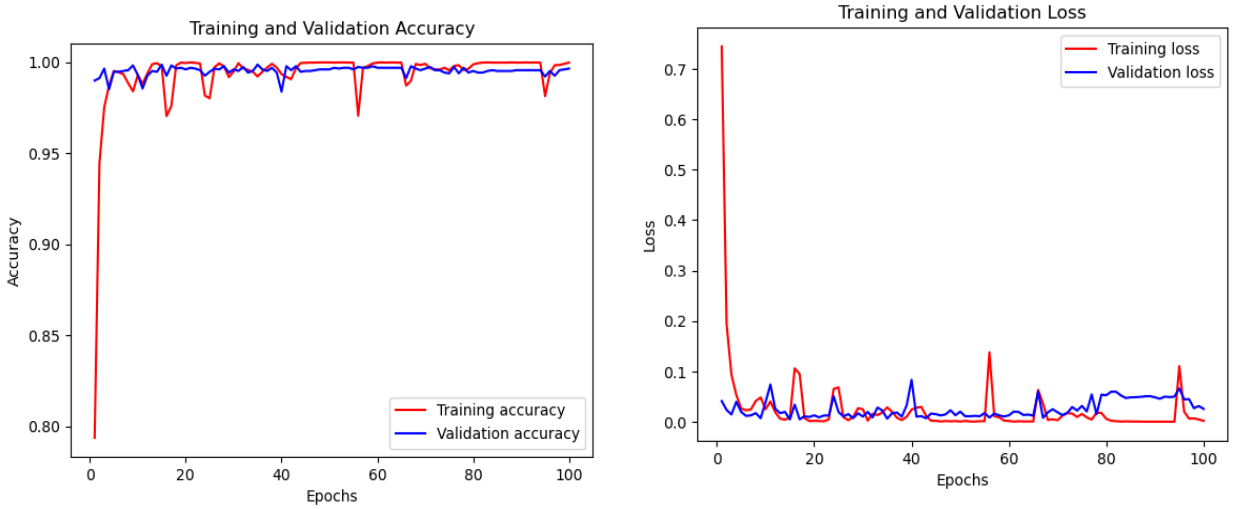


Figure 8. MobileNetV2 Accuracy and Loss Graph

Figure 8 illustrates the performance curve of the MobileNetV2 model. It indicates that the training process is steadily and steadily improving the model's capacity to recognize and sort photos. In the first epoch, the training accuracy was 66.6%, the training loss was 1.2573, and the validation loss was 0.0412. These accuracy numbers show that the model has a rudimentary understanding of how to find patterns, but it is still a long way from being as accurate as it can be. It's common for the model to lose a lot of data in the first stages of learning because it still doesn't fully comprehend how features and labels are related.

The model did much better as the number of epochs went up. The graph depicts this progress: loss keeps going down on both training and validation data, and accuracy keeps getting closer to ideal. At the 15 epoch, for instance, the training accuracy was 99.9% and the validation accuracy was 99.8%. The substantial drop in validation loss up to this point shows that the model can not only remember patterns in training data, but it can also generate new ones.

This tendency keeps going up till the end of the 100 epoch training session. The training accuracy is now almost flawless at 99.99%, while the validation accuracy is still very good at 99.65%. In fact, the training loss value keeps going down until it hits 0.0014, whereas the validation loss value stays the same at 0.0256. These numbers show that the model is not overfitting because the difference in accuracy and loss between the training and validation data is not too big.

This achievement shows how strong the MobileNetV2 architecture is at making picture classification models that are both accurate and fast. MobileNetV2 contains fewer parameters than larger models like ResNet50, yet it still gets very good accuracy. This is particularly crucial for real-world use, especially on mobile devices or systems that are built in and don't have a lot of processing power.

This graph shows that one of the benefits is that the model is stable. The values for accuracy and loss don't move quickly from one epoch to the next. The accuracy curve goes up steadily and smoothly, while the loss curve goes down slowly. This means that the training is going well, with the right learning parameters, and there are no problems like a learning rate that is too high or a batch size that is too small. The model shows that the learning is going in the right direction, is steady, and is correct.

Also, the model's very high validation accuracy shows that it remembers the training data and the overall patterns in the dataset. Because of this, the model can put fresh photos into groups with extremely few mistakes. This illustrates that MobileNetV2 can be used in a wide range of picture categorization systems, both big and small.

The quality of the dataset used also shows that this model works. Clean, well-labeled data that represents different classes is the best way to create a learning model. Regularization or data augmentation during training can also assist keep the model from overfitting and make it better at generalizing to new data.

MobileNetV2's lightweight but very precise performance sets it apart from more complicated models like ResNet50. MobileNetV2 still gives good results, even though it needs less computing resources and trains faster. This means that MobileNetV2 can be used in a lot of real-time applications, like surveillance systems built into edge computing devices, facial recognition, and object detection.

Figure 8 shows how MobileNetV2 learns in the best way possible. MobileNetV2 has shown that it can make a strong, useful picture classification model that can be used in many real-world situations with low loss values and very good training and validation accuracy. This achievement shows that to make a dependable machine learning-based system, you need to choose the right architecture, choose the right training parameters, and process the data well.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Table 2 shows the training results for two Convolutional Neural Network (CNN) models, ResNet50 and MobileNetV2. This table compares the two models based on the accuracy and loss values on the training and validation data.

Table 2. Summary of Training Results

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
ResNet50	86.42%	98.64%	0.4442	0.0719
MobileNetV2	99.99%	99.65%	0.0014	0.0256

Table 2 shows that the MobileNetV2 model is more accurate and has lower loss values than the ResNet50 model. MobileNetV2 has a training accuracy of 99.99% and a validation accuracy of 99.65%. This means the model can generalize very well and doesn't overfit much. At the same time, ResNet50 has a training accuracy of 86.42% and a validation accuracy of 98.64%. The ResNet50 training loss value of 0.4442 is much higher than the MobileNetV2 value of 0.0014. MobileNetV2 is better at learning and more stable on the BISINDO dataset.

3.3 Evaluation Model

Next, we check how well the model works by looking at its precision, recall, and F1-score. The model validation results reveal how many correct and wrong predictions the model made for the tested classes. These findings are used to figure out the metric values. At this point, you may see the confusion matrix for each model.

a. ResNet50

According to Table 3. The ResNet50 model does a great job at classifying the BISINDO alphabet, with a precision value of 98.80%, a recall value of 98.69%, and an F1-score of 98.57%. The high precision number means that the model's predictions are largely right and that there aren't many mistakes in the classification. The high recall also shows that the model can find practically all of the right photos for each class quite well. The F1-score is also high when these two measurements are combined. This result indicates that the model can find a lot of different types of data while still being very accurate.

Tabel 3. Evaluation Matrix ResNet50

Precision	Recall	F1-score
98.80%	98.69%	98.57%

Using the confusion matrix in Figure 9, we found that the ResNet50 model made 31 mistakes when classifying 2.285 BISINDO alphabet images. One mistake was that 3 pictures of the letter D were wrongly called H, 12 pictures of the letter O were wrongly called A, B, P, S, V, W, and Z, and 16 pictures of the letter P were wrongly called B, F, and S.

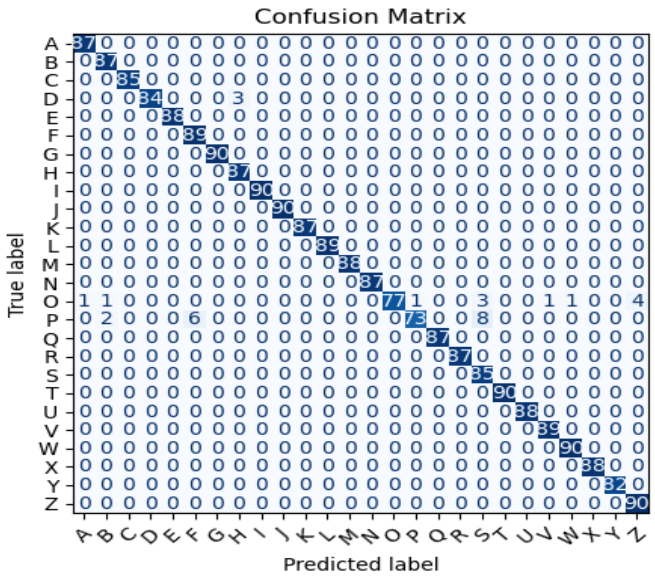


Figure 9. Confusion matrix ResNet50

b. MobileNetV2

The MobileNetV2 model's precision, recall, and F1-score metrics are shown in Table 4. The MobileNetV2 model does very well, with a precision value of 99.69%, a recall value of 99.65%, and an F1-score value of 99.61%. The model's precision value of 99.69% means that practically all of its positive predictions are right, which means that the mistake rate for classifying things is quite low. The recall rating of 99.65% also shows that the model can find practically all the

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 3, Juni 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i3.8667
Hal 319-328
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

right photos in each class. An F1-score that shows how effectively the model can recognize BISINDO alphabet images is made up of an accuracy value of 99.69% and a recall value of 99.65%.

Tabel 4. Evaluation Matrix MobileNetV2

Precision	Recall	F1-score
99.69%	99.65%	99.61%

The confusion matrix showed that there were 8 photos that were incorrectly classified when the MobileNetV2 model was tested using 2.285 BISINDO alphabet images. These mistakes mostly happened in eight pictures of the letter P, which were wrongly labeled as the letters D and V, as shown ini Figure 10.

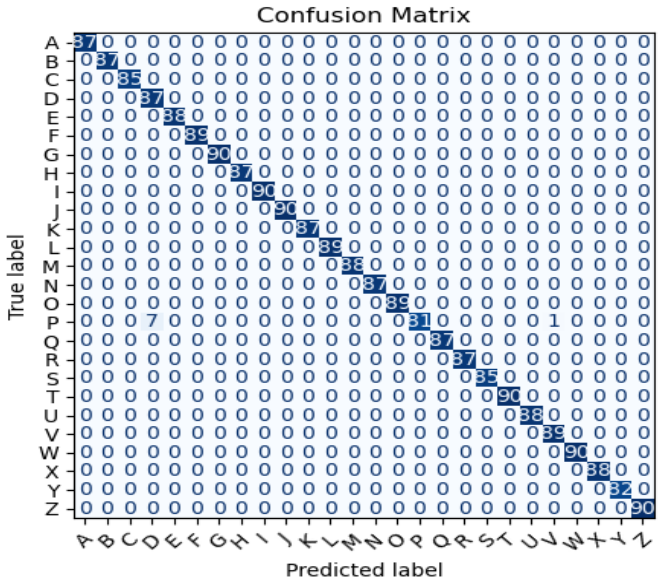
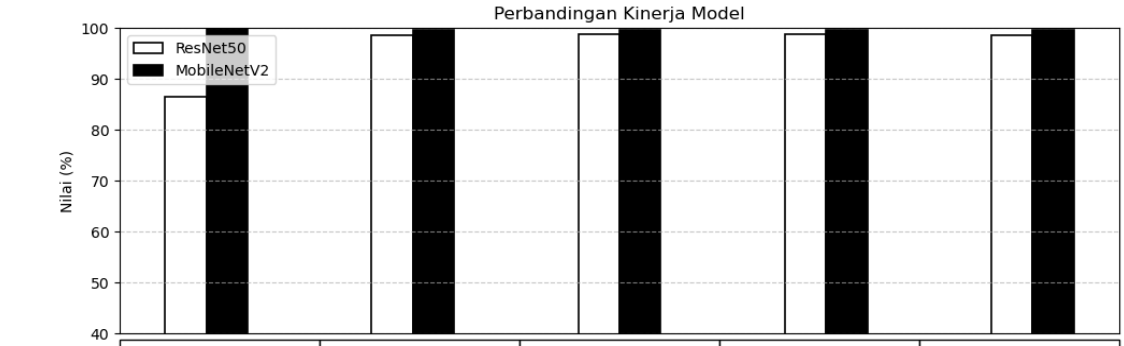


Figure 10. Confusion matrix MobileNetV2

Figure 11 displays the outcomes of the research and compares the accuracy of the training, validation, and evaluation metrics for ResNet50 and MobileNetV2. This chart helps you figure out which model is ideal for BISINDO alphabet classification by showing how well both models work overall.



	Accuracy Training	Accuracy Validation	Precision	Recall	F1-score
ResNet50	86.42	98.64	98.80	98.69	98.57
MobileNetV2	99.99	99.65	99.69	99.65	99.61

Figure 11. ResNet50 vs MobileNetV2 Performance

4. CONCLUSION

The goal of this study is to see how well two Convolutional Neural Network (CNN) architectures, ResNet50 and MobileNetV2, can classify photos of the Indonesian Sign Language (BISINDO) alphabet. We used a number of different metrics to compare the two models, including as accuracy, precision, recall, F1-score, and the confusion matrix. Both models got better as the number of epochs went up during training, and they didn't show any evidence of overfitting. This suggests that the models could learn from the training data without getting worse at the validation data. With a training accuracy of 86.42% and a validation accuracy of 98.64%, ResNet50 does rather well. Also, this model has a precision of 98.80%, a recall of 98.69%, and an F1-score of 98.57%. MobileNetV2, on the other hand, does better on all measures.

This model had a training accuracy of 99.99%, a validation accuracy of 99.65%, a precision value of 99.69%, a recall of 99.65%, and an F1-score of 99.61%. MobileNetV2 also made fewer classification mistakes, with only 8 images being misclassified. In comparison, ResNet50 misclassified 31 images. These results show that MobileNetV2 is a very good choice for the BISINDO alphabet recognition system since it is lightweight and works well. This is especially important for real-world applications that need to be fast and accurate. This study does have some flaws, though, especially when it comes to the amount and variety of data used, which only comes from one dataset source. To make the system more useful in more realistic and inclusive settings, it is suggested that future study integrate datasets with a larger range of lighting, backgrounds, and users.

REFERENCES

[1] A. S. Nugraheni, A. P. Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan Sibi Dan Bisindo Pada Mahasiswa Difabel Tunarungu Di Prodi Pgmi Uin Sunan Kalijaga," *J. Holistika*, vol. 5, no. 1, p. 28, 2023, doi: 10.24853/holistika.5.1.28-33.

[2] R. Mandasari and S. Winduwati, "Upaya Public Relations Pusbisindo dalam Mengampanyekan Penggunaan Bahasa Isyarat Indonesia di Kalangan Masyarakat," *Prologia*, vol. 6, no. 2, pp. 355–361, 2022, doi: 10.24912/pr.v6i2.15572.

[3] N. Tri *et al.*, *Deep Learning : Teori , Algoritma , dan Aplikasi*, no. March. 2025.

[4] Naufal MSesilia ShaniaJessica Millenia et al., "Analisis Perbandingan Algoritma Klasifikasi MLP dan CNN pada Dataset American Sign Language," vol. 1, no. 10, 2021, doi: https://doi.org/10.29207/resti.v5i3.3009.

[5] M. E. Al Rivan and A. G. Riyadi, "Perbandingan Arsitektur LeNet dan AlexNet Pada Metode Convolutional Neural Network Untuk Pengenalan American Sign Language," *J. Komput. Terap.*, vol. 7, no. 1, pp. 53–61, 2021, doi: 10.35143/jkt.v7i1.4489.

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

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

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

[9] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 375–381, 2021, doi: 10.1016/j.gltp.2021.08.027.

[10] Faiz Nashrullah, Suryo Adhi Wibowo, and Gelar Budiman, "The Investigation of Epoch Parameters in ResNet-50 Architecture for Pornographic Classification," *J. Comput. Electron. Telecommun.*, vol. 1, no. 1, pp. 1–8, 2020, doi: 10.52435/complete.v1i1.51.

[11] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification," *Proc. - 2020 2nd Int. Conf. Inf. Technol. Comput. Appl. ITCA 2020*, pp. 476–480, 2020, doi: 10.1109/ITCA52113.2020.00106.

[12] G. W. Intyanto, "Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network)," *J. Arus Elektro Indones.*, vol. 7, no. 3, p. 80, 2021, doi: 10.19184/jaei.v7i3.28141.

[13] S. SATRIA, Sumijan, and Billy Hendrik, "Implementasi Convolutional Neural Netowork Untuk Klasifikasi Citra KTP-El," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 5, no. 1, pp. 169–176, 2024, doi: 10.37859/coscitech.v5i1.6708.

[14] G. Gumelar, Q. Ain, R. Marsuciati, S. Agustanti Bambang, A. Sunyoto, and M. Syukri Mustafa, "Kombinasi Algoritma Sampling dengan Algoritma Klasifikasi untuk Meningkatkan Performa Klasifikasi Dataset Imbalance," *SISFOTEK Sist. Inf. dan Teknol.*, pp. 250–255, 2021.

[15] S. Andika Maulana, S. Husna Batubara, Y. Permata Putri Pasaribu, H. Syahputra, and F. Ramadhani, "Deteksi Burung Menggunakan Convolutional Neural Network (Cnn) Dengan Model Arsitektur Mobilenetv2," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 6108–6114, 2024, doi: 10.36040/jati.v8i4.10126.

[16] D. Martomanggolo, "Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih," *Ultim. J. Tek. Inform.*, vol. 13, no. 1, p. 51, 2021.

[17] A. Derry, M. Krzywinski, and N. Altman, "Convolutional neural networks," *Nat. Methods*, vol. 20, no. 9, pp. 1269–1270, 2023, doi: 10.1038/s41592-023-01973-1.

[18] T. Berliani, E. Rahardja, and L. Septiana, "Perbandingan Kemampuan Klasifikasi Citra X-ray Paru-paru menggunakan Transfer Learning ResNet-50 dan VGG-16," *J. Med. Heal.*, vol. 5, no. 2, pp. 123–135, 2023, doi: 10.28932/jmh.v5i2.6116.

[19] B. Wijayanto, R. M. Mahendra, M. I. Salam, and T. Informatika, "Identifikasi Jenis Ikan Cupang Menggunakan Metode CNN Dengan Arsitektur MobileNetV2 Berbasis Mobile," vol. 4, pp. 519–525, 2025.

[20] D. Kurniawan, D. Ariatmanto, M. T. Informatika, and U. A. Yogyakarta, "Identifikasi varietas bibit durian menggunakan mobilenetv2 berdasarkan gambar daun," vol. 7, no. 2, pp. 231–240, 2024.

[21] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

[22] M. F. Naufal and S. F. Kusuma, "Comparison Analysis of Machine Learning and Deep Learning Algorithms for Image Classification of Indonesian Language Signing Systems (Sibi)," *Jtiik*, vol. 10, no. 4, pp. 873–882, 2023, doi: 10.25126/jtiik.2023106828.