Translator of Indonesian Sign Language Video using Convolutional Neural Network with Transfer Learning

S Shania[1], M F Naufal[2*], V R Prasetyo[3], M S B Azmi[4]

[1,2,3]Faculty of Engineering, Department of Informatics, Universitas Surabaya, Indonesia

[4]Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Malaysia

E-mail: shaniaharsono@gmail.com[1], faridnaufal@staff.ubaya.ac.id[2*], vincent@staff.ubaya.ac.id[3], sanusi@utem.edu.my[4]

Submitted: 23 May 2022, revised: 11 August 2022, accepted: 18 August 2022

Abstract. Sign language is a language used to communicate by utilizing gestures and facial expressions. This study focuses on classification of *Bahasa Isyarat Indonesia* (BISINDO). There are still many people who have difficulty communicating with the deaf people. This study buildt video-based translator system using Convolutional Neural Network (CNN) with transfer learning which was commonly used in computer vision especially in image classification. Transfer learning used in this study were a MobileNetV2, ResNet50V2, and Xception. This study applied 11 different commonly used vocabularies in BISINDO. The predictions were made in a real-time scenario using a webcam. In addition, the system would also ease the interaction approach between deaf and normal people. From all the experiments, it was found that the Xception architectures has the best F1 Score of 98.5%.

Keywords: BISINDO; CNN; Translator; Sign Language

1. Introduction

Bahasa Isyarat Indonesia (BISINDO) is the sign language most often used by the Indonesian deaf people. Besides BISINDO, another sign language is called the *Sistem Isyarat Bahasa Indonesia* (SIBI). However, BISINDO still tends to be used more by deaf people because it is easier to understand [1]. BISINDO is different from SIBI, which only uses one hand and difficult to understand. This is supported by Mursita et al [1] that as many as 91% of 100 deaf respondents use BISINDO for daily communication. BISINDO consists not only the alphabet A to Z, but there are other vocabularies. Therefore, although SIBI has been inaugurated by the government and taught in Special Schools for disabilities, deaf people prefer BISINDO to communicate each other [2].

Based on Center for Data and Information Ministry of Health Indonesian Republic in 2019, 7.03% of Indonesia's population is deaf people [3]. Based on research and surveys conducted by Fajri et al [4] on respondent consisting of workers and students, 89% of 100 respondents did not know how to communicate using BISINDO. This percentage was very high and quite worrying for deaf people because they experience more challenges to communicate with normal people. Deaf people must always communicate with normal people only in writing.

Computer vision is a solution to make communication between deaf and normal people faster and more effective. Convolutional Neural Network (CNN) [5] is commonly used algorithm for image classification in computer vision. It can perform the image convolution process to extract the feature, which is then inputted directly into the neural network model. As a result, the CNN has good accuracy for image classification. Transfer learning [6] is a technique in which a CNN model created for one job is utilized as the basis for a model on a different task. Given the vast compute and time resources required to develop neural network models on these problems, it is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision. This study classifies 11 commonly different vocabularies in BISINDO which are *"Hari ini (Today)", "Isyarat (Cue)", "Kamu (You)", "Maaf (Sorry)", "Makan (Eat)", "Rumah (Home)", "Sakit (Sick)", "Saya (Me)", "Teman (Friend)", "Tidak (No)" and "Out of Vocabulary (OOV)"* using the MobileNetV2, ResNet50V2, and Xception transfer learning architectures. This study also compares their performances.

Aljabar et al [2] used the CNN and Long Short-Term Memory (LSTM) for classifying BISINDO. The accuracy obtained especially from the combination of CNN-LSTM algorithm was 96%. The dataset used in the research consisted of two alphabets and eight vocabularies. The dataset also included human faces and hand gesture. Pre-processing was done by removing the background using object detection. However, the CNN used in the study did not use transfer learning architectures. Yolanda et al [7] used CNN and Recurrent Neural Network (RNN) methods for classifying BISINDO, which obtained an average accuracy of 60.58%. The dataset used consist of 26 alphabets as labels. The dataset is self-generated with 2,582 videos originating from self-made and internet sources. Tests were carried out in real-time but failed because the environment could not support the architecture. Celsia et. al [8] used the Sign Language Digits dataset, which is the data for each digit of the sign language. There are ten labels from 2,062 images in the dataset. It used 2 methods: CNN and ANN and obtained the resulting accuracy of 99.7%. Bantupalli et. al [9] used CNN in American Sign Language (ASL) recognition yields 90% of accuracy. In this study, the dataset used comes from videos converted into frames. The ASL dataset used are alphabet and numbers It used 150 sign languages. The architecture used are CNN for spatial feature recognition and RNN for temporal features.

From several previous studies that have been carried out, no research tries to apply the transfer learning methods to classify BISINDO. Transfer learning is a fast and accurate method for classifying images because it has a pre-trained model. This study uses three types of transfer learning architecture, namely MobileNetV2, a model with low computation time, ResNet50V2, and Xception, which has high accuracy performance. In addition, there are still no studies that try to analyse and compare the performance of several transfer learning models for BISINDO classification. This study also provides an analysis of the performance of each architecture to provide useful information for researchers to choose the right architecture for classifying BISINDO.

2. Methods

Collecting datasets, dataset augmentation, model training, model testing, and performance calculation were all part of this study's research methodology as seen in Figure 1.

2.1. Dataset Collection

The total number of datasets was 220 images which contained 20 images for each class. Each image was an RGB image and had different pixel size variations with the same ratio 1:1. Each of these classes may have two or more gestures. The collection of datasets was obtained from YouTube and independently data retrieval. The data from YouTube was taken by screen shooting the video. The videos were from Kirana

Salsabila [10] who is an influencer with hearing impairment with more than 133,000 subscribers in November 2021 and Amanda Farliany [11], who is also youtuber with hearing impairment who has 57,400 subscribers. Amanda's video used for this study had a total of 81,000 views. Apart from YouTube, the data collection was done independently by taking personal photos using a smartphone. Figure 2 shows the example of BISINDO dataset.

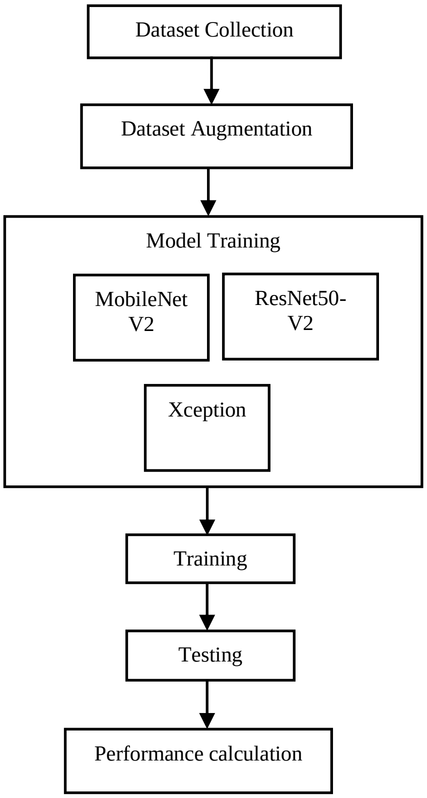


Figure 1. Research methodology

2.2. Data Augmentation

The data augmentation process was carried out before the training process. It was conducted because the number of datasets was relatively small and limited, so it could add some possible images that would occur. First, the data augmentation was performed on the train set and validation set. In the train set, the data augmentation was performed using several parameters: rescale, shift range, shear range, zoom range, horizontal flip, and fill mode. The rescale process was a normalization method so that RGB values ranging from 0 to 255 were divided by 255, which produced a range of values between 0 to 1. For the validation set, it only used the rescale parameter. Then each set had a batch size of 32.

2.3. Model Training

At this stage, the dataset was divided into 80% train set and 20% validation set. During the model training process, the validation sets were used to validate more precise accuracy using 5-Folds Cross-Validation. Then the results of the training model were saved in HDF5 extension format. This file contained the model architecture, weights, and compile information that was done automatically using TensorFlow. Table 1 shows the computer specification for train the model.

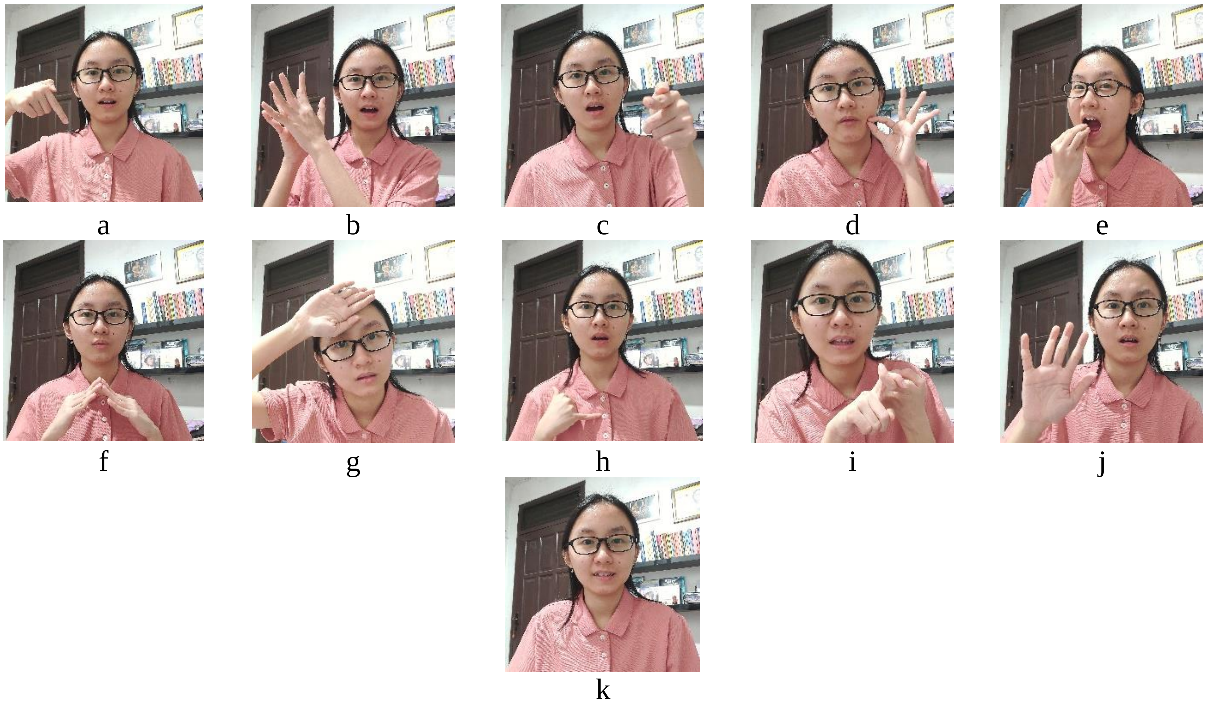


Figure 2. BISINDO example dataset. (a) “Hari ini (Today)”, (b) “Isyarat (Cue)”, (c) “Kamu (You)”, (d) “Maaf (Sorry)”, (e) “Makan (Eat)”, (f) “Rumah (Home)”, (g) “Sakit (Sick)”, (h) “Saya (Me)”, (i) “Teman (Friend)”, (j) “Tidak (No)”, and (k) “Out of Vocabulary (OOV)”

Table 1. Computer specification for model training

Parameter	Specification
CPU	Intel® Xeon®, 2.30 GHz, 2 cores
RAM	12 GB
Space of Disk	358 GB
GPU Model Name	Nvidia K80, 12 GB

MobileNetV2 continued to use deep and directed convolution in the same way as MobileNetV1 [12]. MobileNetV2 offered two additional features, linear bottlenecks and shortcut links between the bottlenecks [12]. There were inputs and outputs between the models at the bottleneck. Simultaneously, the inner layer encapsulated the model's capability of converting inputs from lower-level concepts (i.e., pixels) to higher-level descriptors (i.e., image categories). This method provide faster training and improved accuracy. ResNet50V2 [13] is an improved version of the original ResNet50. There was a change made to the propagation formulation of the links between blocks in ResNet50V2. Using the ImageNet dataset, ResNet50V2 also performed well. Xception is an acronym for Extreme Inception [14]. Xception used the same model parameters as Inception-v3 but outperformed it on the 17,000-class ImageNet dataset. ImageNet is a massive image database based on the structure of WordNet [15]. ImageNet is highly effective for recognizing objects, detecting images, and grouping images. The ImageNet database was utilized as a training dataset for the CNN model, then was used to develop the transfer learning model. This research extracted pre-trained models from the ImageNet database using the Keras library. Adam Optimizer is an extension of Stochastic Gradient Descent (SGD), a technique for picture classification that has gained traction in deep learning [16]. Adam optimizer is computationally efficient, requires little memory, and is simple to implement [16]. This study used an Adam optimizer to modify the neural network's properties to reduce loss. Categorical Cross Entropy was a loss function based on the Cross-Entropy principle for multiclass classification. Categorical Cross Entropy was used in this study to train the CNN probability output on the face mask picture. In addition, softmax was employed in the CNN architecture's output layer since it was the sole activation function suggested for multi-label categorization [17]. This study used a total of 500 epochs. The number of epochs was determined by considering the training duration and the specs of the machine used to train the model. According to the results described in the results chapter, 500 epochs was adequate to converge the training data correctness. After developing the model, the model was compiled using loss parameters with categorical cross-entropy value, and the optimizer was Adam Optimizer. Then it could be continued with the model training process of 500 epochs per fold. Finally, after training the model, it produced output in a weighted model. Transfer learning performed feature extraction and then connected it to a layer that was classified according to the number of existing labels. Table 1 represents the model training parameter.

Table 2. Model training parameter

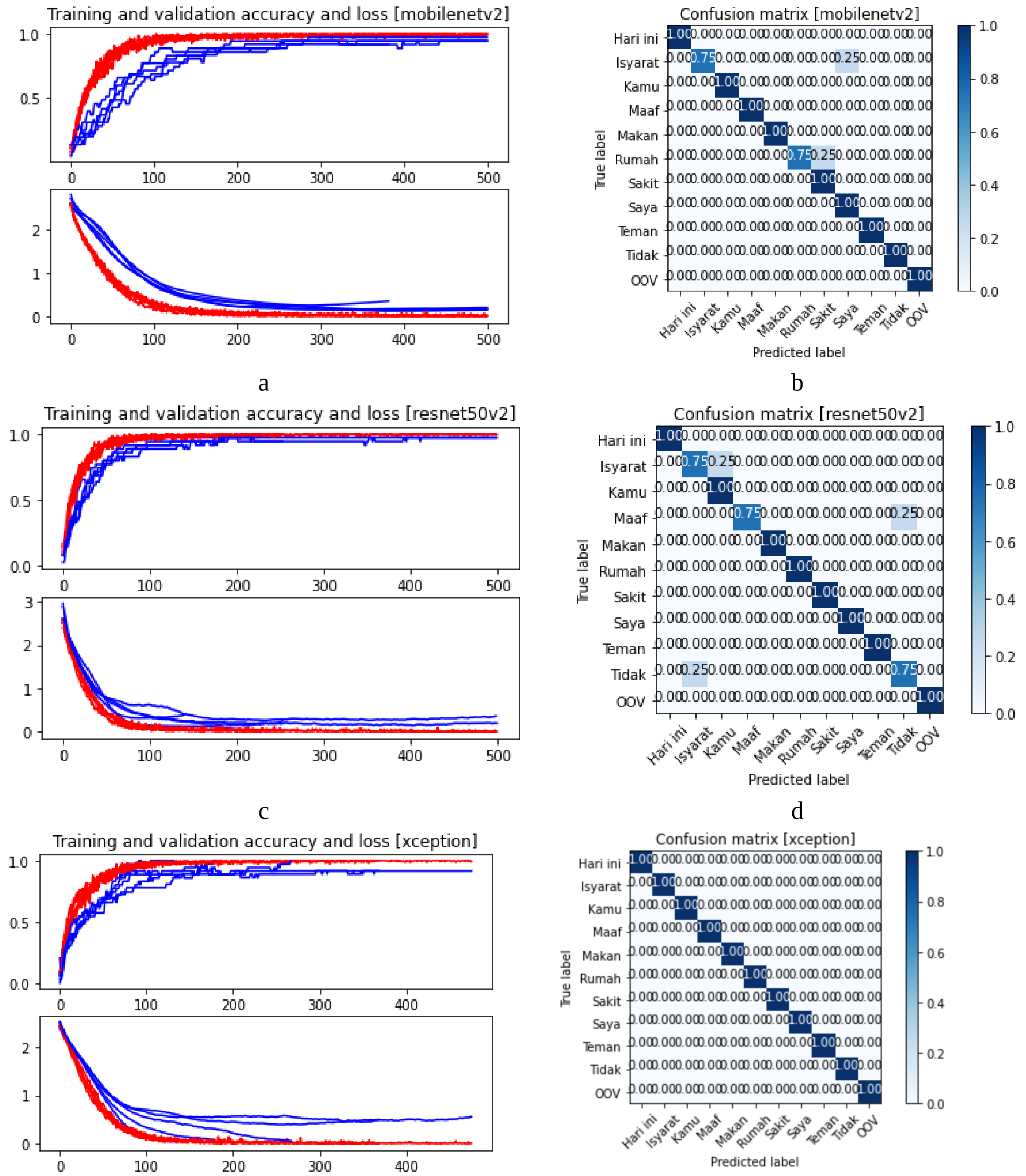
Hyperparameter	**Type**
Transfer Learning Architecture	MobileNetV2, ResNet50V2, Xception
Transfer Learning Database	ImageNet
Optimizer	Adam
Loss Function	Categorical Cross Entropy
Epoch	500

3. Results and Discussion

This section explains the accuracy results obtained during the dataset training process and dataset test. A validation was carried out to prove that the system detected the BISINDO dataset correctly and adequately. The metrics used were accuracy and F1 score. This study used cross validation so that the mean performance was used to calculate performance. Meanwhile, in some result tables, there were various columns such as Mean Validation Accuracy (MTA), Mean Validation Loss (MVL), Mean Test Accuracy (MTA), Mean Test Loss (MTL), and Mean F1 Score. Various experiments were carried out on three different models. The datasets used were dataset with facial expression with hand gesture and hand gesture only. The Xception architecture obtained the best results with 98.2% of F1 Score. Table 2 represent comparison of model performance on facial expression and hand gestures dataset.

Table 2. Comparison of model performance on facial expression with hand gestures dataset

Model	**MTA**	**MVL**	**MTA**	**MTL**	***Mean F1 Score***
MobileNetV2	94.32%	0.217	93.63%	0.56	93.6%
ResNet50V2	96.57%	0.251	91.36%	0.468	91.4%
Xception	96.62%	0.251	98.18%	0.15	98.2%



a b

c d

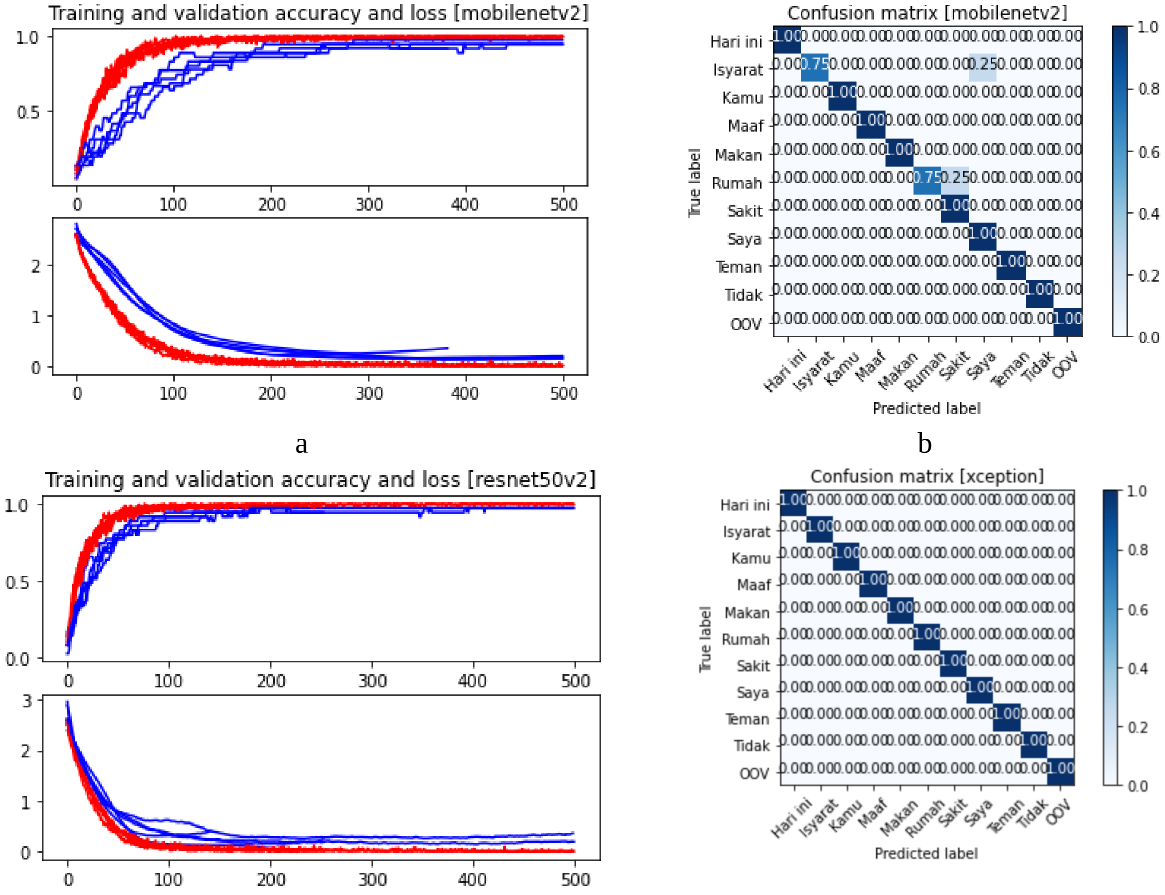
e f

Figure 3. Training loss on the left side and confusion matrix on the right side of each model in facial expression with hand gestures dataset. (a) - (b) MobileNetV2, (b) - (c) ResNet50V2, and (d) - (e) Xception.

This experiment also used the dataset which only contained hand gestures as the main feature target. The experimental results on hand gesture only datasets can be seen in Table 3. Figure 4 shows the training loss and confusion matrix of each model.

Table 3. Comparison of performance on hand gestures dataset

Model	**MVA**	**MVL**	**MTA**	**MTL**	***F1 Score***
MobileNetV2	98%	0,008	98%	0,085	98%
ResNet50V2	99,38%	0,09	95%	0,186	95%
Xception	99,38%	0,075	98,5%	0,047	98.5%

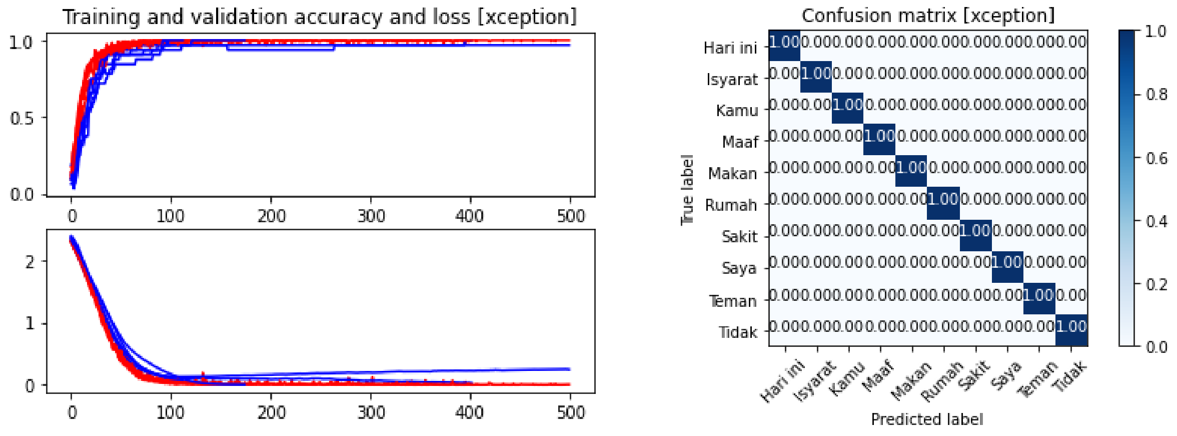


a

b

c

d



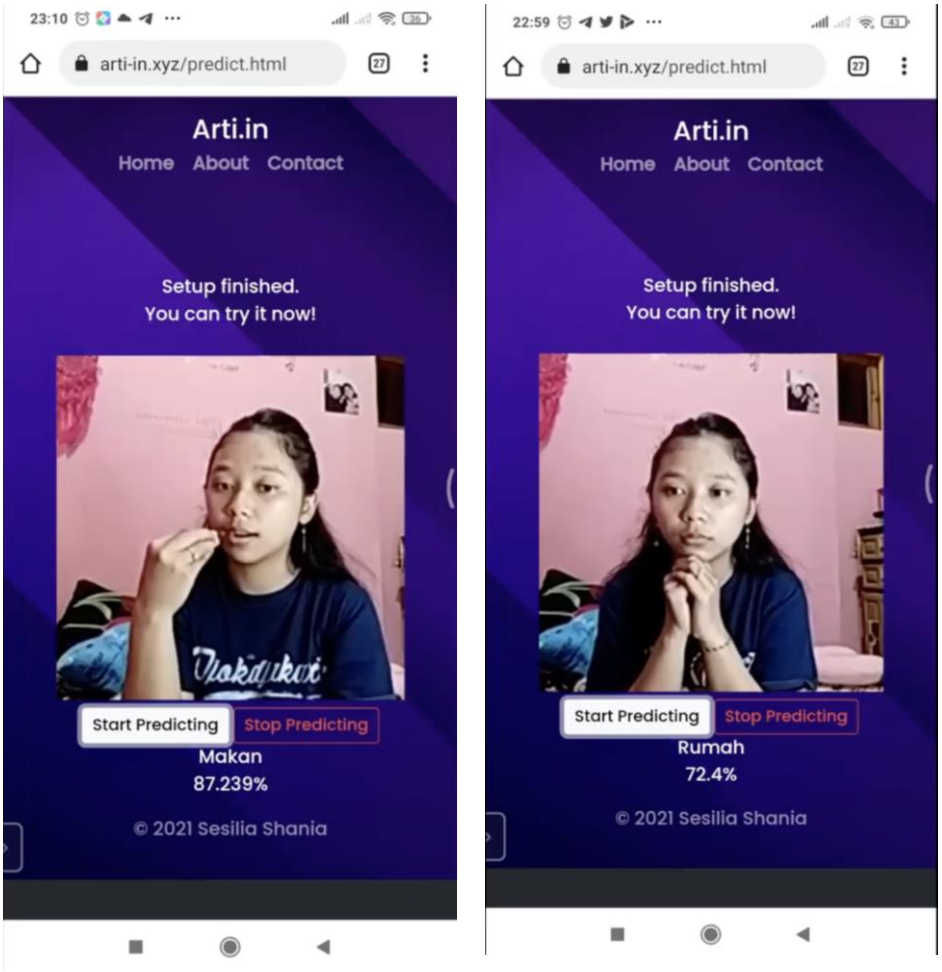
e f

Figure 4. Training loss on the left side and confusion matrix on the right side of each model in hand gestures only dataset. (a) - (b) MobileNetV2, (c) - (d) ResNet50V2, and (e) - (f) Xception.

The whole of this experiment made Xception model as the best model in this section since it showed the F1 Score on the different experiment. Therefore, it can be concluded that in the case of the BISINDO dataset, a complex model architecture was needed in order to produce good accuracy and loss. The second experiment using hand gestures as the main feature could increase the F1 Score significantly. However, hand gestures could not be used as the only solution for translating BISINDO. According to BISINDO practitioners, hand gestures and facial expressions were equally important features. The two target features correlated with each other. Same hand gestures may occur, but the meaning of BISINDO might change when different facial expressions were performed. Although the target hand gesture feature provided better accuracy, this could not be used as a parameter to increase accuracy in the case of BISINDO translation.

The validation was then carried out directly on the application. This trial was conducted through a questionnaire using Google Forms that was distributed to 11 respondents. The respondents were experts in BISINDO. Figure 5 shows the web application of BISINDO classification applied in this study.

Overall, the respondents responded positively to the BISINDO translator system because all respondents answered that the system could translate BISINDO well. The following validation used an approach where there was an interaction between users. There were two subjects in this validation: one person who understood BISINDO and one who did not understand BISINDO. There were 12 scenarios 12, with details of 10 words and two sentences. After that, people who understood BISINDO signaled according to the scenario per number that normal people would answer through the Google Forms provided. The results of the scenarios showed that the system succeeded in translating BISINDO. When this validation was carried out, the translator system was certainly not completely accurate due to various existing factors. However, all scenarios were answered correctly to conclude that the translator system was able to achieve the desired goal.



Setup finished.
You can try it now!

Makan
87.239%

© 2021 Sesilia Shania

Setup finished.
You can try it now!

Rumah
72.4%

© 2021 Sesilia Shania

a b

Figure 5. Web application of video based BISINDO translator.

4. Conclusion

The conclusion is transfer learning with Xception architecture is the best model for classifying BISINDO. Overall, from the responses of 11 experts regarding the translation system, respondents said that the system could translate BISINDO and was very helpful in communication between deaf and normal people. The vocabulary could be added more in further research because the variations in BISINDO's vocabulary were very diverse. In addition, models could also be developed to run smoothly on low specifications hardware. Finally, facial and hand tracking methods based on gesture recognition could be applied in further research to capture the object more precisely.

5. References

[1] R. A. Mursita, R. Tunarungu, M. Pascasarjana, and P. Upi Bandung, "Respon Tunarungu Terhadap Penggunaan Sistem Bahasa Isyarat Indonesa (Sibi) dan Bahasa Isyarat Indonesia (Bisindo) dalam Komunikasi," *INKLUSI J. Disabil. Stud.*, vol. 2, no. 2, pp. 221–232, Dec. 2015, doi: 10.14421/IJDS.2202.

[2] A. Aljabar and Suharjito, “BISINDO (Bahasa isyarat indonesia) sign language recognition using CNN and LSTM,” *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 5, pp. 282–287, 2020, doi: 10.25046/AJ050535.

[3] Kemenkes RI, “Disabilitas Tuna Rungu,” *InfoDATIN Pusat Data dan Informasi Kementerian Keseharan RI*. pp. 1–9, 2019. [Online]. Available: https://pusdatin.kemkes.go.id/resources/download/pusdatin/infodatin/infodatin-tunarungu-2019.pdf

[4] B. B. R. Fajri, B. R. Fajri, and G. Kusumastuti, “Perceptions of ‘Hearing’ People on Sign Language Learning,” in *5th International Conference on Education and Technology (ICET 2019)*, Dec. 2019, pp. 364–367. doi: 10.2991/ICET-19.2019.91.

[5] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/NECO_A_00990.

[6] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11141 LNCS, pp. 270–279, 2018, doi: 10.1007/978-3-030-01424-7_27.

[7] D. Yolanda, K. Gunadi, and E. Setyati, “Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network,” *Infra*, vol. 8, no. 1, pp. 203–208, 2020.

[8] F. K. Celsia and G. A. Sandag, “Implementation of Deep Learning on Number Recognition in Sign Language,” *Sisfotenika*, vol. 11, no. 2, p. 124, 2021, doi: 10.30700/jst.v11i2.1117.

[9] K. Bantupalli and Y. Xie, “American Sign Language Recognition Using Machine Learning and Computer Vision.” Accessed: Mar. 30, 2022. [Online]. Available: https://digitalcommons.kennesaw.edu/cs_etd/21

[10] Kirana Salsabila, “Kirana Salsabila - YouTube,” *Youtube*, 2021. https://www.youtube.com/channel/UCJUvJ_Sc5hsv8CJPcTG_oyw (accessed Mar. 30, 2022).

[11] A. Farliany, “Amanda Farliany - YouTube,” *Youtube*, 2021. https://www.youtube.com/c/AmandaFarliany (accessed Mar. 30, 2022).

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, Jan. 2018, Accessed: May 27, 2021. [Online]. Available: http://arxiv.org/abs/1801.04381

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38/TABLES/5.

[14] F. Chollet, “XCeption: Deep Learning with Depthwise Separable Convolutions,” *Comput. Vis. Found.*, 2016, doi: 10.4271/2014-01-0975.

[15] L. Fei-Fei, J. Deng, and K. Li, “ImageNet: Constructing a large-scale image database,” *J. Vis.*, vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

[16] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.