Performance Evaluation of EfficientNetB0, EfficientNetV2, and MobileNetV3 for American Sign Language Classification

1st Jason Hartanto
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
jason.hartanto@binus.ac.id

2nd Sean Matthew Wijaya
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
sean.wijaya@binus.ac.id

3rd Anderies
Computer Science Departement
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
anderies@binus.ac.id

4th Andry Chowanda
Computer Science Departement
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
achowanda@binus.edu

***Abstract*— Sign language is a type of communication system that involves the gestures of our hands and our facial expressions to communicate with people who are deaf. Unfortunately, it is estimated that 1.5 billion people in the world have this disability. Lots of previous research have been done regarding the task of recognizing sign language. For instance, a multi-scale and dual sign language recognition network (SLR-Net) based on a graph convolutional network (GCN) is used and achieves 98.08% accuracy in the CSL-500 dataset. Another approach uses 3D CNNs trained on 100 words on the Boston ASL LVD dataset and achieves 96% precision, 97.1% recall and 96.4% F-measure. Building upon this success, our work investigates the use of transfer learning with pre-trained CNN models, such as EfficientNetB0, EfficientNetV2, and MobileNetV3, to compare and investigate the performance of American Sign Language (ASL) recognition for further development. The dataset that is being used consists of 13,000 images classified into 29 classes. The mentioned models achieve accuracy of 99,54%, 93,93%, and 74.62% for EfficientNetB0, Mobilenetv3, and EfficientNetV2, respectively.**

Keywords—Computer Vision, American Sign Language, EfficientNetB0, EfficientNetV2, MobileNetV3, Image Classification

I. INTRODUCTION

Sign language is a type of communication system that involves the gestures of our hands and our facial expressions to communicate with people who are deaf. Unfortunately, it is estimated that 1.5 billion people in the world have this disability [1]. It is why the development of automatic sign language recognition is important. These days, people either still use the help of sign language experts to translate or learn sign language themselves when trying to communicate with people with hearing disabilities, which is a very noneffective way [2]. With technology that is constantly evolving, we can build an automatic translator that will be a more convenient and effective way to communicate.

Deep learning is a subset of machine learning, specifically a neural network architecture designed to recognize images as well as face recognition in both dark and light areas. Basically, deep learning works by scanning every patch and transforming them into tokens. These tokens will then be fed into a transformer encoder which will be the final step of processing, and it will give a final output. Deep learning performs well when training with large training data, which is suitable when trying to recognize sign language [3].

Several types of deep learning models can be used to recognize sign languages, such as EfficientNetB0, EfficientNetV2, and MobileNetV3. EfficientNet is a model that uses a scaling method where the scales can be customized to achieve better accuracy only if the dataset provided is enough. EfficientNetB0 is the original model of EfficientNet that uses a novel approach to scale a neural network to optimize its performance [4]. It has fewer parameters and computational complexity compared to larger variants. On the other hand, EfficientNetV2 is a newer convolutional network model that has a faster performance and better parameter efficiency compared to previous models. This model's training can be sped up by increasing the image size, as well as doing progressive learning, which adjusts regularization to compensate for the accuracy drop when speeding up the training [5]. Lastly, MobileNetV3 is a neural network architecture model that can recognize images accurately. Adjusting aspects such as depth multipliers and resolution is key to achieving high accuracy when using this model [6].

In this paper, our goal is to make a comparison between EfficientNetB0, EfficientNetV2, and MobileNetV3 [4]–[6] to find which method produces the highest accuracy when recognizing American Sign Language (ASL). The results that we get are intended to help other researchers and ourselves to decide which model works best for recognizing American Sign Language so further developments in the future can be conducted to improve the model.

The study of this paper is structured as follows: Section 2 discusses the related works in the implementation of American Sign Language Recognition. Then, the methodology and our proposed model will be given in Section 3. Followed by Section 4, which will present the result and the discussion of the proposed model. Finally, Section 5 will present the conclusion.

II. LITERATURE REVIEW

Sign Language Recognition (SLR) is one of the popular research areas in computer vision that focuses on creating models that can detect and interpret sign language accurately. However, creating a robust model for sign language recognition is a challenging task since there are many variables that affect the meaning of certain gestures [7]. To address this problem, there have been many approaches/methods in creating a robust system for recognizing sign language accurately.

One approach is a sensor-based approach. A full-fiber auxetic-interlaced yarn sensor (AIYS) is sewn into a glove (16 channels), and it can accurately recognize 26 letters of the alphabet as well as classify 2600 sign language gestures by using ANN algorithm with an accuracy of 99.8% [8]. Flex sensors, Inertial Measurement Units (IMU) and Force Sensing Resistors can also be applied to a glove to detect hand movements. An Arduino micro is used to transmit the data to a computer, and it has an accuracy of 96.9% when using Support Vector Machine (SVM) and 94.5% when using Dynamic Time Wrapping (DTW) [2]. The DTW motion sensor can also be modified into a weighted one to improve the recognition rate. Sign-Glove is a system built by combining a bend sensor to recognize hand shape and WonderSense to recognize hand motion. It resulted in an accuracy of 85.21% [9]. A smart band can also be used to recognize sign language since it has built-in integrated nanocomposite pressure sensors that can detect contractions and relaxations of muscles with high-sensitivity settings. The band will be flexible and can be wireless. It has an accuracy of 93% when tested with 0-9 in sign language with ten tries each [10]. A DIY low-cost dataglove can also be created by using flex sensors, IMU, and a microcontroller. The approach utilized is image-based Spatial Projection using 24 static and 16 dynamic American Sign Language data. It has a static accuracy of 82.19% and a dynamic accuracy of 97.35% [11]. To detect and recognize continuous hand motions, a soft wrist-worn sensor system (SWSS) with electromyography and pressure-sensing capabilities was developed. Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) techniques were used, and it resulted in an average accuracy of 86,8% [12].

Another approach is the use of Kinect and RF sensors to detect sign language. Jimenez *et al*. [13] propose a system that leverages Microsoft Kinect Sensor to acquire sign trajectories and pictures, which are then fed into Hidden Markov Models (HMM) for real-time analysis. The proposed method achieves 99% and 88% accuracy and F1 score, respectively. Gurbuz *et al.* [14] propose a system that uses RF sensors to measure the micro-doppler effect from the signer hand's movement and use machine learning to help identify linguistic properties of ASL signing. Muhammad *et al.* [15] focus on testing different lighting conditions by using a Microsoft Kinect sensor to capture the gestures and using Artificial Neural Network (ANN) to classify hand gestures under different lighting environments. It results in a very high accuracy of 97% with a 2.7% error rate.

A more popular approach is the use of a variety of deep learning architectures to create a robust model for sign language recognition. Peréz *et al.* [16] proposed a system that uses a depth camera (OAK-D) to obtain 3D coordinates of the motions and RNN for classification. Yang *et al.* [17] propose a system that uses a two-layer bidirectional recurrent neural network for recognizing hand gestures captured from a leap motion controller. The proposed method achieve accuracy of 100% and 96.7% on training and testing datasets, respectively, with 360 samples. Mark *et al.* [18] proposed a system that combines two-stream CNN to extract features from raw picture data on a frame-by-frame basis, which is then fed to RNN to exploit temporal information in the video. This system achieved an 18% improvement compared to the existing state-of-the-art. Meng *et al.* [19] proposed a system based on a graph convolutional network (GCN) that employs a multi-scale and dual sign language recognition network (SLR-Net). The proposed system achieves 98.08% accuracy in the CSL-500 dataset and 64.5% accuracy in the DEVISIGN-L dataset. Tunga *et al.* [7] proposed a system that uses GCN to capture spatial interactions in the video and BERT to capture temporal dependencies between frames. The proposed system achieves a 5% improvement in accuracy compared to the existing state-of-the-art. Barbhuiya *et al.* [20] proposed a system that uses pre-trained deep learning CNN AlexNet and VGG16 for feature extraction and SVM for classification. The proposed system achieves an accuracy of 99.82%. Sharma *et al.* [21] proposed a system that uses 3D CNNs trained on 100 words on the Boston ASL LVD dataset and achieves 96% precision, 97.1% recall and 96.4% F-measure. Jiang *et al.* [22] proposed a system that uses Skeleton Aware Multi-modal architecture with a Global Ensemble Model to achieve state-of-the-art performance on three tough, isolated SLR datasets (i.e., AUTSL, SLR500, and WLASL2000). Zhou *et al.* [23] proposed a system that uses a Spatial-temporal multi-cue (STMC) network and achieves state-of-the-art performance on three large-scale CSLR benchmarks. Doan *et al.* [24] analyze that the use of a blender-based augmentation technique can improve most CNN architecture by 9% in terms of accuracy. Renxiang *et al.* [25] use an improved YOLOv5 method and CBAM attention mechanism as well as a layer aggregation network module to accurately detect hand movement even in a complex background. EgoHands and TinyHGR are the datasets that are used in this experiment, with an accuracy of 75.6% and 66.8%, respectively.

Our work will try to use EfficientNetB0, EfficientNetv2, and MobileNetv3 [4]–[6] for recognizing American Sign Language. Finally, the result of each model will be compared to see which is best for the task of recognizing American Sign Language.

III. RESEARCH METHODOLOGY

There are three main sections in the proposed research model which are data collecting & pre-processing, training, and evaluation. Each section is important and needs to be done carefully to provide good outcomes at each stage. Therefore, we can study the main objective, which is to compare and evaluate the performance of EfficientNetB0, EfficientNetv2, and MobileNetV3 [4]–[6] in recognizing American Sign Language. The code is accessible through this link https://github.com/jasonh14/ResearchPaperASL.git.

A. *Data Collecting & Pre-processing*

American Sign Language Dataset is being used in this experiment. The dataset consists of 13,000 images consisting of 29 classes with a size of 200 x 200 pixels. The dataset is gathered from various viewpoints and different lighting conditions as subjects move their hands about on the picture plane and along the z-axis. Different data pre-processing is done to the dataset for each model to ensure that the dataset is usable and reliable. We also do data augmentation to the dataset, which includes five steps (Random Flip, Random Rotation, Random Zoom, Random Height, and Random Width). Data augmentation is a technique to increase the size of a dataset by doing various random transformations [26]. It helps to prevent overfitting by providing more variants of the dataset.

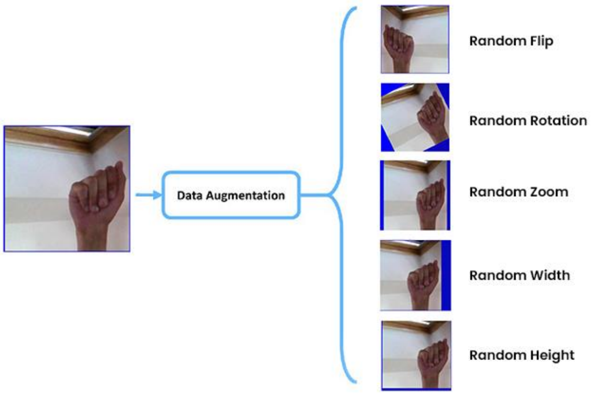

Fig. 1. Data Augmentation

- **Random Flip**: The images in the dataset are flipped both horizontally and vertically. Flipping is one of the easiest data augmentation techniques that can be applied and has been proven useful on datasets such as CIFAR-10 and ImageNet [26].
- **Random Rotation**: The images in the dataset are rotated with the maximum angle of 0.2 radians or 11.5 degrees, both clockwise and counterclockwise direction. Slight rotation could be useful for some recognition tasks [26].
- **Random Zoom:** The images are zoomed in and out randomly with a maximum change is 20% which can help the model to learn different scales of the data.
- **Random Height**: The height of the images is randomly shifted with maximum changes is 20% which can help the model to learn different vertical changes in the data.
- **Random Width**: The width of the images is randomly shifted with maximum changes is 20% which can help the model to learn different horizontal changes in the data.

B. Training

EfficientNetB0, EfficientNetv2, and Mobilenetv3 [4]–[6] are used in this experiment to perform sign language recognition. EfficientNetB0 works by scaling and balancing the depth, width, and resolution of the CNN to achieve better performance [4]. On the other hand, EfficietNetV2 uses the combination of training-aware neural architecture search with scaling in order to jointly enhance training speed and parameter effectiveness [5], while MobileNetV3 mixes the combination of manual design decisions with architectural search methods to produce reliable and effective CNN models [6]. All three models are pre-trained with the ImageNet dataset in order to increase performance and decrease the training time.

Transfer learning technique is used, and the model is fine-tuned for the task of recognizing American Sign Language. The categorical Cross-Entropy loss function is used to improve the accuracy and prevent overfitting of the model, and the Adam (Adaptive Moment Estimation) optimizer with a learning rate of 0.00001 is used to train the deep learning models by updating the model's parameters during training.

With EfficientNetB0, EfficientNetv2, and Mobilenetv3 [4]–[6] pre-trained model, we experiment with training the model with 50 epochs as we find it to be the most balanced in terms of training time and model performance. The three models are tested on the ASL dataset, and the results are analyzed. The dataset is divided into two parts: 80% for training and 20% for testing. Furthermore, the training set is separated into 80% for training and 20% for validation.

C. Evaluation

In this section, to evaluate the model, accuracy and loss functions are used. Accuracy is calculated based on the total number of correct predictions divided by the total number of predictions. The formula is given by:

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

where

TP True Positive

TN True Negative

FP False Positive

FN False Negative

The categorical Cross-Entropy loss function is used for multi-classification problems. The difference between the real probabilities (expressed as one-hot encoded labels) and the true probabilities as predicted by the model is measured. The loss function is given by:

$$J_{cce} = \frac{-1}{M} \sum_{k=1}^{K} \sum_{m=1}^{M} y_m^k \log\left(h_\theta(x_m, k)\right) \quad (2)$$

where

M photos for training examples

K number of American sign language classes

y_m^k target label for class k training example m

h_θ weighted θ neural network model

x m as input for training example

IV. Result

In this section of the paper, we present the outcomes of our experiment using the methodology described in section III. In Research, Evaluation is the most crucial step. By evaluating it, we can determine how accurate or successful our model is. To evaluate the model, we calculated the accuracy of the train, validation, and test, as well as the loss for each model and compared it. Fifty epochs are used in this experiment, and the results are as follows.

A. EfficientNetB0

The EfficinetNetB0 model is tested on 50 epochs, and the training and validation accuracy can be seen the Fig. 2.

Fig. 2. Illustrates the train and validation accuracy across 50 epochs for EfficientNetB0. Overall, it increased over time, starting at 0.6846 for the training accuracy and ending at 0.9994. As for validation accuracy, it starts at 0.9207 and ends at 0.9966. Starting at epoch six, the accuracy plateaued at approximately 0.99.

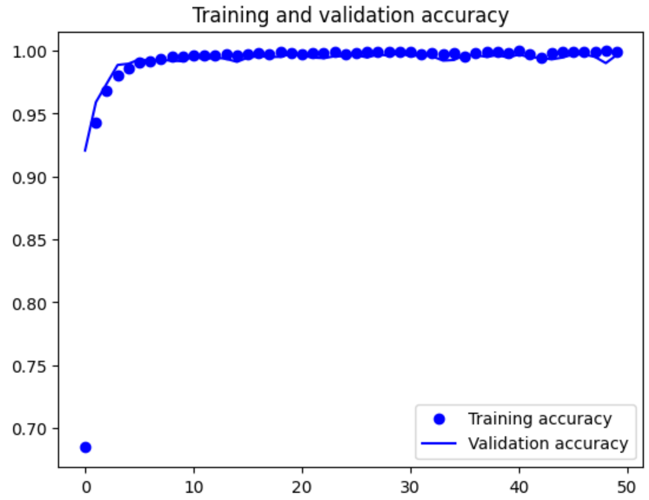

Fig. 2. Training and Validation Accuracy Graph of EfficientNetB0

Fig. 3. Illustrates the loss of the train and validation throughout 50 epochs for efficientNetB0. Overall, it decreased over time, plummeting at 0.0035 after starting at 1.4488 for train loss. The validation loss scale runs from 0.4836 to 0.0119. According to the graph, the model is improving as its losses are minimized along with the number of epochs.

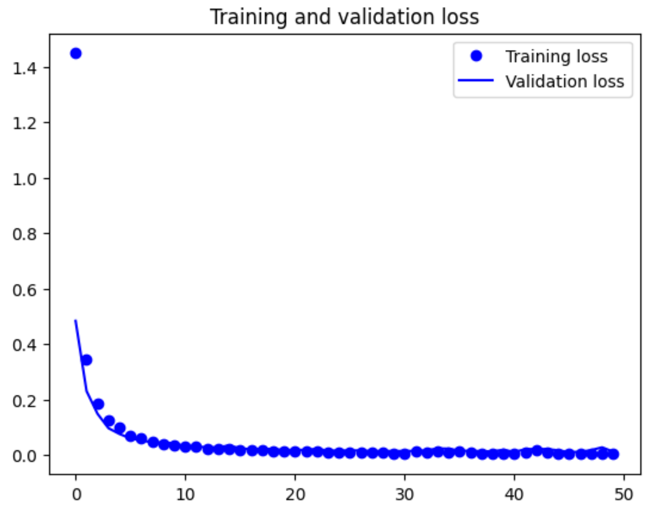

Fig. 3. Training and Validation Loss Graph of EfficientNetB0

B. *EfficientNetV2*

The EfficinetNetV2 model is evaluated on 50 epochs, and the training and validation accuracy is shown in Fig. 4.

Fig. 4. Illustrates the train and validation accuracy across 50 epochs for EfficientNetV2. Overall, it rose with time, peaking at 0.5046 after starting at 0.0350 for train accuracy. In terms of validation accuracy, it ranges from 0.0466 to 0.7516.

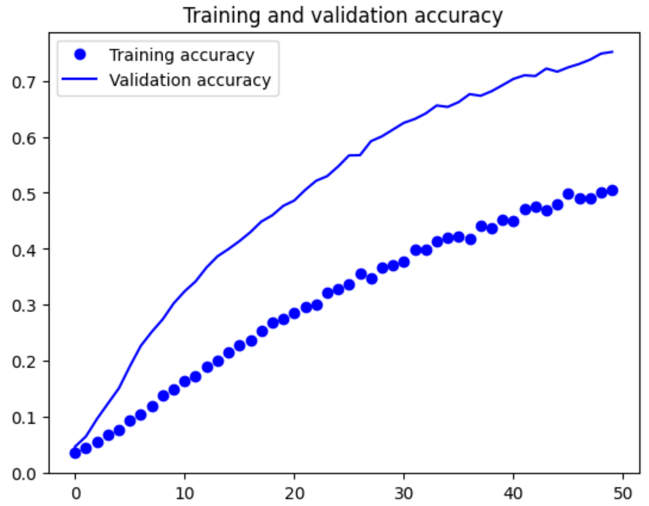

Fig. 4. Training and Validation Accuracy Graph of EfficientNetV2

Fig. 5. Illustrates the loss of the train and validation throughout 50 epochs for EfficientNetV2. Overall, it dropped with time, falling to 1.6497 from 4.2946 for the training loss. The range of validation losses is 3.6349 to 1.1381. The graph shows that the model is becoming better as the number of epochs and the losses are reduced.

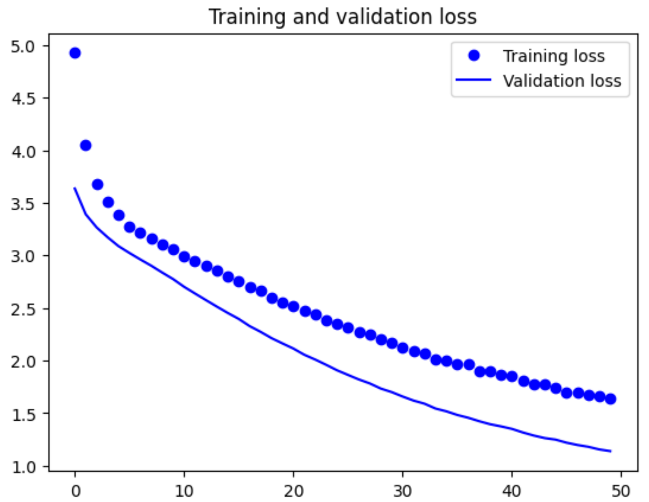

Fig. 5. Training and Validation Loss Graph of EfficientNetV2

C. *MobileNetV3*

The MobileNetV3 model is put to the test across 50 epochs, and Fig. 6 shows the training and validation accuracy.

Fig. 6. Illustrates the train and validation accuracy across 50 epochs for MobileNetV3. Overall, it increased with time, reaching a high of 0.8471 after beginning at 0.0314 for train accuracy. The validation accuracy varies from 0.0302 to 0.9375.

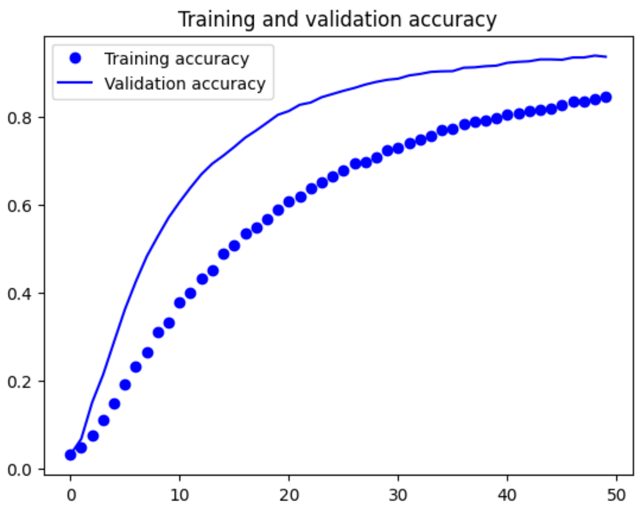

Fig. 6. Training and Validation Accuracy Graph of MobileNetV3

Fig. 7. Illustrates the loss of the train and validation throughout 50 epochs for MobileNetV3. Overall, it fell over time, from 3.6458 to 0.5976 for the training loss. Validation losses vary from 3.4338 to 0.3888. The graph indicates that the model improves as the number of epochs and losses decrease.

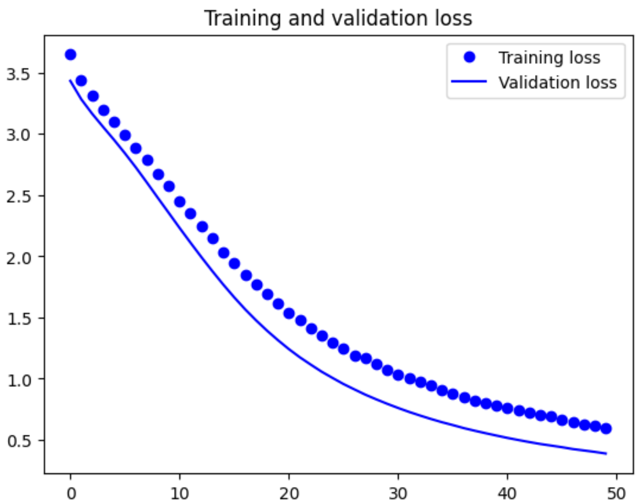

Fig. 7. Training and Validation Loss Graph of MobileNetV3

D. *Model Comparison*

All three models were tested on a test set, and the result can be seen in Fig. 8.

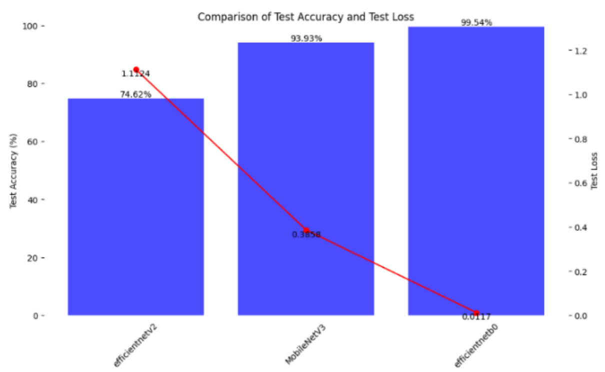

Fig. 8. Test Set Comparison

EfficientNetB0 has the highest accuracy with 99,54% and 0.0117 loss, followed by MobileNetV3 with 93.93% accuracy and 0.3858 loss, and the last one is EfficientNetV2 with 74.62% and 1.1124 loss.

V. Conclusion

In this study, we conducted testing using EfficientNetB0, EfficientNetv2, and Mobilenetv3 pre-trained models to recognize ASL sign language and analyze the performance of each model. The dataset used in this experiment consists of 13,000 data with 29 classes. The model is tested to classify the ASL dataset into the correct classes, and the highest accuracy is obtained by the EfficientNetB0 model, while the lowest is EfficientNetV2. In addition to current research ideas, the model's performance might be assessed on a broader range of hardware platforms, including mobile devices and embedded systems. This would aid in determining the model's scalability and reliability in real-world applications.

Reference

[1] I. Oosthuizen, C. Frisby, S. Chadha, V. Manchaiah, and D. W. Swanepoel, "Combined hearing and vision screening programs: A scoping review," *Front. Public Heal.*, vol. 11, 2023, doi: 10.3389/fpubh.2023.1119851.

[2] X. Chu, J. Liu, and S. Shimamoto, "A sensor-based hand gesture recognition system for Japanese sign language," *LifeTech 2021 - 2021 IEEE 3rd Glob. Conf. Life Sci. Technol.*, pp. 311–312, 2021, doi: 10.1109/LifeTech52111.2021.9391981.

[3] L. Alzubaidi *et al.*, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00727-2.

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[5] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," 2021, [Online]. Available: http://arxiv.org/abs/2104.00298.

[6] A. Howard *et al.*, "Searching for mobileNetV3," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.

[7] A. Tunga, S. V. Nuthalapati, and J. Wachs, "Pose-based Sign Language Recognition using GCN and BERT," *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vis. Work. WACVW 2021*, pp. 31–40, 2021, doi: 10.1109/WACVW52041.2021.00008.

[8] R. Wu, S. Seo, L. Ma, J. Bae, and T. Kim, "Full-Fiber Auxetic-Interlaced Yarn Sensor for Sign-Language Translation Glove Assisted by Artificial Neural Network," *Nano-Micro Lett.*, vol. 14, no. 1, pp. 1–14, 2022, doi: 10.1007/s40820-022-00887-5.

[9] C. Lu, S. Amino, and L. Jing, "Data Glove with Bending Sensor and Inertial Sensor Based on

Weighted DTW Fusion for Sign Language Recognition," *Electron.*, vol. 12, no. 3, 2023, doi: 10.3390/electronics12030613.

[10] R. Ramalingame *et al.*, "Wearable Smart Band for American Sign Language Recognition with Polymer Carbon Nanocomposite-Based Pressure Sensors," *IEEE Sensors Lett.*, vol. 5, no. 6, pp. 10–13, 2021, doi: 10.1109/LSENS.2021.3081689.

[11] M. A. A. Faisal, F. F. Abir, M. U. Ahmed, and M. A. R. Ahad, "Exploiting domain transformation and deep learning for hand gesture recognition using a low-cost dataglove," *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, 2022, doi: 10.1038/s41598-022-25108-2.

[12] W. Dong, L. Yang, R. Gravina, and G. Fortino, "Soft Wrist-Worn Multi-Functional Sensor Array for Real-Time Hand Gesture Recognition," *IEEE Sens. J.*, vol. 22, no. 18, pp. 17505–17514, 2022, doi: 10.1109/JSEN.2021.3050175.

[13] C. O. Sosa-Jimenez, H. V. Rios-Figueroa, and A. L. Solis-Gonzalez-Cosio, "A Prototype for Mexican Sign Language Recognition and Synthesis in Support of a Primary Care Physician," *IEEE Access*, vol. 10, no. November, pp. 127620–127635, 2022, doi: 10.1109/ACCESS.2022.3226696.

[14] S. Z. Gurbuz *et al.*, "American Sign Language Recognition Using RF Sensing," *IEEE Sens. J.*, vol. 21, no. 3, pp. 3763–3775, 2021, doi: 10.1109/JSEN.2020.3022376.

[15] M. Haroon, S. Altaf, S. Ahmad, M. Zaindin, S. Huda, and S. Iqbal, "Hand Gesture Recognition with Symmetric Pattern under Diverse Illuminated Conditions Using Artificial Neural Network," *Symmetry (Basel).*, vol. 14, no. 10, 2022, doi: 10.3390/sym14102045.

[16] K. Mejía-Peréz, D. M. Córdova-Esparza, J. Terven, A. M. Herrera-Navarro, T. García-Ramírez, and A. Ramírez-Pedraza, "Automatic Recognition of Mexican Sign Language Using a Depth Camera and Recurrent Neural Networks," *Appl. Sci.*, vol. 12, no. 11, pp. 1–16, 2022, doi: 10.3390/app12115523.

[17] L. Yang, J. Chen, and W. Zhu, "Dynamic Hand Gesture Recognition Based on a Leap," *Sensors*, vol. 20, no. 7, pp. 1–17, 2020.

[18] K. P. . C. Mark Borg, "SIGN LANGUAGE DETECTION ' IN THE WILD ' WITH RECURRENT NEURAL NETWORKS Mark Borg , Kenneth P . Camilleri," pp. 1637–1641, 2019.

[19] L. Meng and R. Li, "An attention-enhanced multi-scale and dual sign language recognition network based on a graph convolution network," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–22, 2021, doi: 10.3390/s21041120.

[20] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," *Multimed. Tools Appl.*, vol. 80, no. 2, pp. 3051–3069, 2021, doi: 10.1007/s11042-020-09829-y.

[21] S. Sharma and K. Kumar, "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks," *Multimed. Tools Appl.*, vol. 80, no. 17, pp. 26319–26331, 2021, doi: 10.1007/s11042-021-10768-5.

[22] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Sign Language Recognition via Skeleton-Aware Multi-Model Ensemble," pp. 1–12, 2021, [Online]. Available: http://arxiv.org/abs/2110.06161.

[23] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 13009–13016, 2020, doi: 10.1609/aaai.v34i07.7001.

[24] H. G. Doan and N. T. Nguyen, "New blender-based augmentation method with quantitative evaluation of CNNs for hand gesture recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 2, pp. 796–806, 2023, doi: 10.11591/ijeecs.v30.i2.pp796-806.

[25] R. Chen and X. Tian, "Gesture Detection and Recognition Based on Object Detection in Complex Background," *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074480.

[26] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.