



CNN Performance Improvement for Classifying Stunted Facial Images Using Early Stopping Approach

Yunidar Yunidar^{1,2}, Y Yusni^{1,3}, N Nasaruddin^{1,2} Fitri Arnia*^{1,2}

¹Program Doktor Ilmu Teknik, Universitas Syiah Kuala, Banda Aceh, Indonesia

²Jurusan Teknik Elektro dan Komputer, Universitas Syiah Kuala, Banda Aceh, Indonesia

³Bagian Fisiologi, Fakultas Kedokteran, Universitas Syiah Kuala, Banda Aceh, Indonesia

Abstract

Stunting, a condition characterised by short stature, is a growth disorder caused by chronic malnutrition, which often begins in the womb. Children affected by stunting usually show different physical and cognitive characteristics compared to their peers. Research shows that these physical differences can also be observed in facial features. Because faces provide important information and are commonly studied in digital image processing, in this study, we will compare the facial image classification performance of stunted children versus normal children using various Convolutional Neural Network (CNN) architectures. The evaluated architectures include MobileNetV2, InceptionV3, VGG19, ResNet18, EfficientNetB0, and AlexNet. To improve the learning process, augmentation techniques with Haar cascade and Gaussian filters were applied so that the data set increased from 1,000 to 6,000 images. After adding the dataset, training is carried out with an early stop approach to minimise overfitting. The main aim of this research is to identify the CNN model that is most effective in differentiating facial images of stunted children from normal children. The results show that the EfficientNetB0 architecture outperforms other models, achieving 100% accuracy. Early stopping has been shown to improve training efficiency and help prevent overfitting.

Keywords: CNN; early stopping; faces; Haar Cascade; stunting; stunted

How to Cite: F. Arnia, Y. Yunidar, Y. Yusni, and N. Nasaruddin, "CNN Performance Improvement for Classifying Stunted Facial Images Using Early Stopping Approach", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 1, pp. 62 - 68, Jan. 2025.

DOI: <https://doi.org/10.29207/resti.v9i1.6068>

1. Introduction

Stunting is a serious health problem that impacts children's physical growth and cognitive development. Early detection of stunting is critical for timely and effective intervention. Traditional approaches to detecting stunting generally involve measuring body height, which can be time-consuming and require special equipment. However, technological developments in image processing have become an alternative for detecting stunting through facial image analysis. [1]-[4].

Convolutional Neural Networks (CNN) have been proven effective in a variety of image classification tasks, including identifying medical conditions through image analysis [5]. Several CNN architectures have been built for stunting detection, namely the AlexNet architecture [4], malnutrition detection, and similar using the Inception and VGG19 [2] and the ResNet18 architecture for malnutrition detection in children [6].

In this research, we compare the performance of several popular CNN architectures, namely MobileNetV2, InceptionV3, VGG19, ResNet18, EfficientNetB0, and AlexNet, in classifying facial images of stunted children. The Haar Cascade technique is used to detect the location of faces from an image, while early stopping is applied to prevent overfitting during model training [7], [8]. Previously, we succeeded in building a facial image classification model using ResNet50 and AlexNet architecture [9].

2. Research Methods

2.1. Dataset

In this study, a data set was used that included facial photos of normal children and stunted children obtained from previous research [9]. Some examples of the images are shown in Figure 1. The collected data was 1000 images of children's faces, which were then augmented to increase the accuracy of the CNN model.

The data is then processed using the Haar Cascade technique to detect the eye, head, and hair areas, which are characteristic of stunted children. After detection, the image is cropped from the top to the bottom of the child's eyes [10], [11]. Data augmentation was carried out using five different techniques to enrich the dataset. First, the zoom technique is applied with a factor of 0.1. Second, rotation is carried out randomly with angles of 5, 10, and 15 degrees. Third, a translation of 2% is applied, resulting in a shift of approximately 5 pixels both horizontally and vertically [12], [13]. Fourth, horizontal flipping is performed on the image. Finally, Gaussian noise was added with a sigma value of 25. Some examples of augmented images are shown in Figure 2. This augmentation process aims to increase the variety of the dataset, which in turn can increase the generalization ability and accuracy of the CNN model in detecting stunting in children. This dataset augmentation process produces significant variations from the original data. Each augmentation technique generates 1,000 new images, and the application of five different augmentation techniques yields a total of 5,000 augmented images. Thus, the total number of datasets used in this research reached 6,000 images, consisting of 1,000 original images and 5,000 augmented images. Next, the dataset is divided into three, namely training, validation, and testing, with a ratio of 80:10:10.

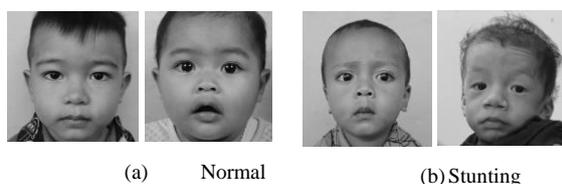


Figure 1. Samples of Normal and Stunting Images

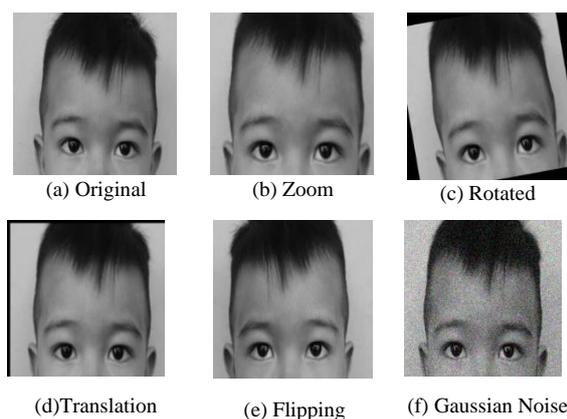


Figure 1. Samples dataset with augmentation

Data augmentation techniques such as zoom, rotation, translation, flipping, and Gaussian noise are very important in improving the performance and robustness of stunting classification models on child facial images [14], [15]. These methods allow the model to adapt to variations that may arise when taking photos in real-life situations. Zoom replicates the difference in distance between the camera and the child's face, allowing the model to distinguish indicators of stunting at different

scales. Rotation allows the model to recognize stunting features from different angles, improving its ability to identify signs of stunting regardless of facial orientation [16]. Translation or shifting of the image horizontally or vertically simulates situations where the child's face is not always in the center of the frame, teaching the model to recognize the characteristics of stunting without relying on a particular facial position. Horizontal flipping provides additional variety by mirroring the image, helping the model recognize stunting features from both sides of the face. Finally, the addition of Gaussian noise increases the model's robustness to minor image disturbances that may occur due to varying camera quality or lighting. By applying these five augmentation techniques, the model becomes more flexible and adaptive in handling various shooting scenarios in the real world, thereby increasing accuracy and reliability in detecting stunting in children.

The focus of the results to be achieved is: 'Stunting' and 'Normal' by utilizing a Deep Learning model based on Convolutional Neural Network (CNN) to classify facial images of children who are stunted, especially based on grayscale facial images. The main strength of CNN lies in its ability to recognize visual elements such as edges, textures, corners, and patterns.

2.2. Simulation

Figure 3 shows the simulation process conducted to achieve the goal namely, to classify the facial images into 'Stunting' and 'Normal' classes. The first stage was to input the image dataset. The input dataset was divided into three stages, namely training, validation, and testing. Training images were 4800, while each validation and testing image was 600 images. With the dataset, we conducted the simulation using the training hyperparameters shown in Table 1. In the second stage, the upper eye area from facial features was extracted using the Haar Cascade Viola-Jones algorithm, resulting in ROI. The images were resized to 227x227 pixels for use in the AlexNet architecture and 224x224 pixels for the other five architectures. In the third stage, the processed facial images were used to train six CNN architectures. Finally, model validation and evaluation were performed. The model's performance was evaluated based on accuracy, precision, recall, and F1-score metrics for each architecture.

Table 1. Hyperparameter Training

Epoch	Learning rate	Optimizer	Batch size	Loss function
10, 20, 40, 60, 80, 100	0,001	Stochastic Gradient Descent (SGD)	32	Binary Cross Entropy

In neural network training, several key parameters play an important role in model optimization. Epoch refers to one complete cycle in which the entire dataset has gone through the process of forward and backward propagation in a neural network [17]. The learning rate determines the number of steps taken by the

optimization algorithm during the training process [18], while the batch size determines the number of data samples processed by the model in each iteration. Stochastic Gradient Descent (SGD) is an optimization algorithm used to update model parameters, especially weights and bias, with the binary cross-entropy loss function as an evaluation metric [19].

The selection of parameters such as epoch 10, 20, 40, 60, 80, and 100, learning rate 0.001, SGD optimizer, and batch size 32 is based on the careful consideration to optimize the model learning process [20]. The epoch

that produces the best results will be chosen to give the model enough time to learn patterns in the data without excessive risk of overfitting. A learning rate of 0.001 is a commonly used value as a starting point, offering a balance between stability and learning efficiency. The Stochastic Gradient Descent was chosen because of its simplicity and effectiveness in various cases, especially when combined with a relatively small learning rate. Meanwhile, batch size 32 provides a good balance between training speed and memory usage while still allowing stable gradient estimation and good generalization.

Our ultimate goal is to create an accurate model for classifying stunting on children's faces through facial image analysis. To achieve this, we carefully consider the characteristics of each parameter and their interactions. This approach allows the model to achieve

optimal performance in the classification task, bringing us closer to our goal.

After setting the training hyperparameters, the dataset was trained using several architectures, namely, AlexNet, VGG, ResNet, EfficientNet, MobileNet, and Inception. Following this, the model underwent a rigorous validation process. This step is crucial as it evaluates the model's performance using data that is not used in the training process. By doing so, we ensure that the model not only learns patterns from the training data but also generalizes well to new data [14], thereby enhancing its robustness.

Once the training and validation process is complete, the model is generated. To test a model's accuracy, it is necessary to process it with test images, where the resulting output is a stunting or normal class. Then, an evaluation process is carried out using the parameters accuracy, precision, recall, and F1-score.

3. Results and Discussions

Using the prepared dataset and according to the simulation flow illustrated in Figure 3, we present the performance of the models, using the graphs of training and validation losses over 100 epochs, with the early stopping approach applied to all architectures. In addition, an overview of the prediction results for each architecture will be displayed via a confusion matrix, and the performance analysis is summarized in Table 2.

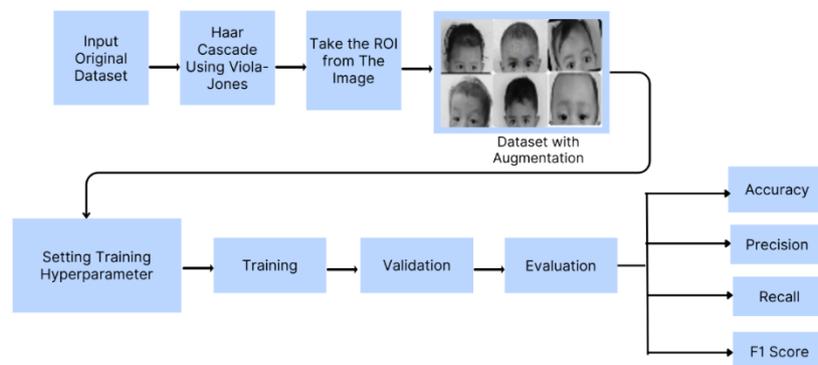


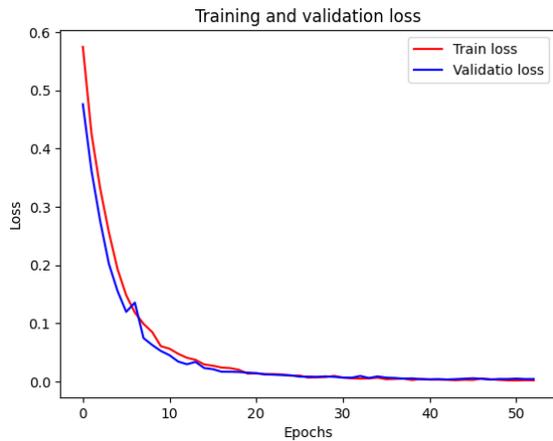
Figure 3. Simulation flow

Figure 4 shows the graph of training and validation losses. In Figure 4(a), the loss graph was trained using 300 epochs with a patience of 5. This means that if the validation loss does not decrease after 5 epochs, the system will stop. In this case, MobileNet stopped at epoch 53. Figure 4(b) shows the loss curve of the early stopping implementation on the InceptionV3 architecture, where the system stops at epoch 30. From Figure 4(c), the loss curve of the early stopping implementation on the VGG architecture is visible, where the system stops at epoch 19. The ResNet architecture shown in Figure 4(d) has a loss curve that performs early stopping at epoch 43. Figure 4(e) displays the loss curve implementing early stopping on the EfficientNetB0 architecture, where the loss graph appears unstable, causing the system to stop too early at

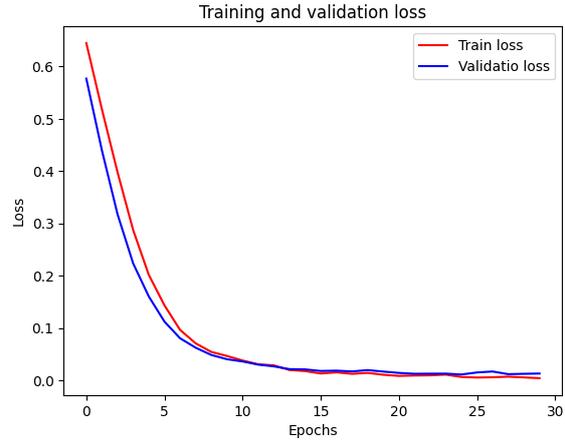
epoch 13. Finally, Figure 4(f) presents the loss curve on the AlexNet architecture with early stopping at epoch 60. Early stopping is a strategy to avoid "overfitting." When the model stops improving and begins to perform poorly during training, one solution is to halt the training process. In other words, the model will "terminate" training earlier to prevent overfitting [21].

Figure 5 shows the confusion matrix for the six architecture models with the use of early stopping to calculate the proportion of accurate and inaccurate predictions. In Figure 5(a), the MobileNetV2 confusion matrix shows the evaluation of the model using the test data. It can be seen that 299 images were correctly classified as stunting (True Positive/TP), 1 image was incorrectly classified as stunting (False Positive/FP),

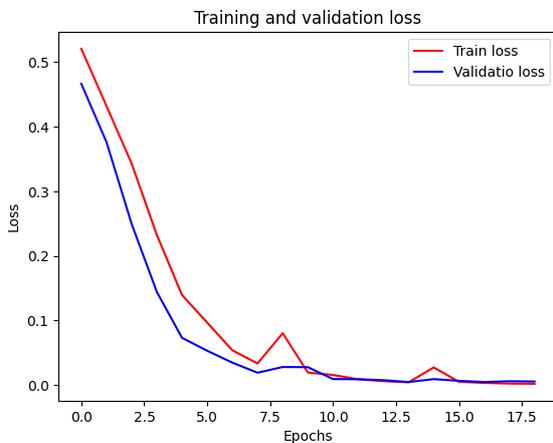
and 300 images were correctly classified as normal (True Negative/TN).



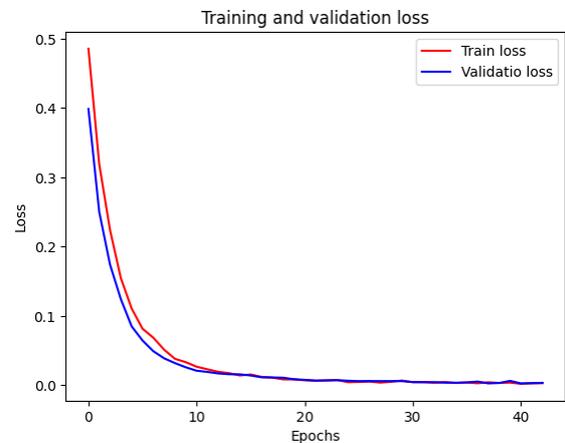
(a) MobileNetV2 Early Stopping Training Loss and Validation Graph



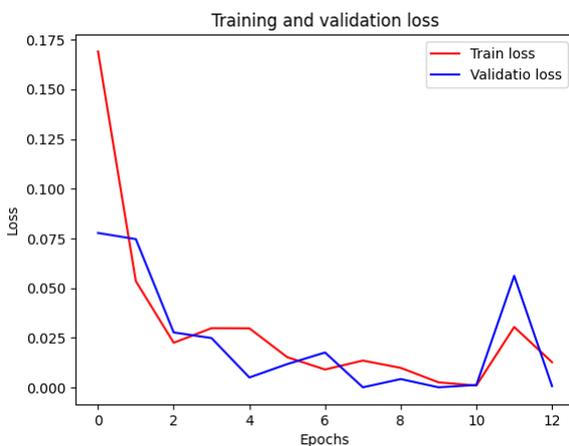
(b) InceptionV3 Early Stopping Training Loss and Validation Graph



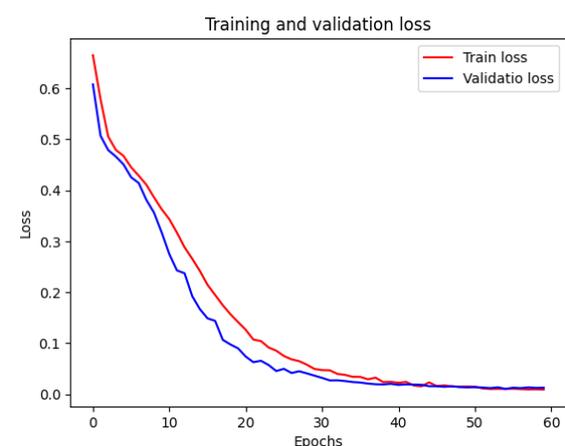
(c) VGG19 Early Stopping Training Loss and Validation Graph



(d) ResNet18 Early Stopping Training Loss and Validation Graph



(e) EfficientNetB0 Early Stopping Training Loss and Validation Graph



(f) AlexNet Early Stopping Training Loss and Validation Graph

Figure 4. Training and Validation Loss graphs for 100 Epochs with Early Stopping applied for all Architecture

In Figure 5(b), the InceptionV3 confusion matrix with early stopping shows the model's evaluation of the test data. It shows that 300 images were correctly classified as stunting (True Positive/TP), 3 images were incorrectly classified as normal (False Negative/FN), 5 images were incorrectly classified as stunting (False

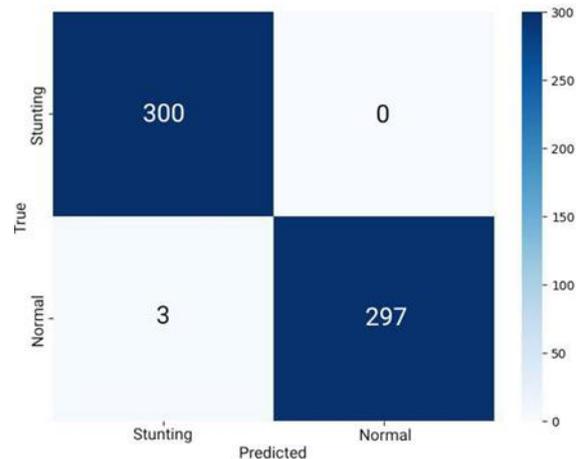
Positive/FP), and 297 images were correctly classified as normal (True Negative/TN). In Figure 5(c), the VGG19 confusion matrix presents the evaluation results from the test data. It shows that 299 images were correctly classified as stunting (True Positive/TP), 1 image was incorrectly classified as normal (False

Negative/FN), and 300 images were correctly classified as normal (True Negative/TN). In Figure 5(d), the ResNet18 confusion matrix displays the test data evaluation. The model correctly classified 299 images as stunting (True Positive/TP), incorrectly classified 1 image as normal (False Negative/FN), and correctly classified 300 images as normal (True Negative/TN). In Figure 5(e), the EfficientNetB0 confusion matrix presents the test data evaluation. It shows that 300 images were correctly classified as stunting (True

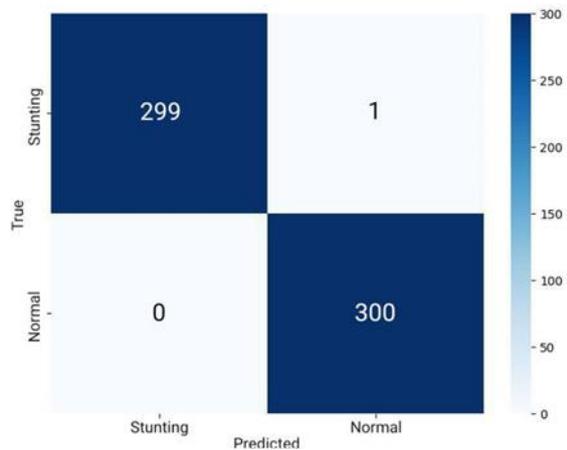
Positive/TP), and 300 were correctly classified as normal (True Negative/TN). Lastly, in Figure 5(f), the AlexNet confusion matrix with early stopping shows the evaluation of the test data. The model correctly classified 299 images as stunting (True Positive/TP), incorrectly classified 1 image as normal (False Negative/FN), incorrectly classified 1 image as stunting (False Positive/FP), and correctly classified 299 images as normal (True Negative/TN).



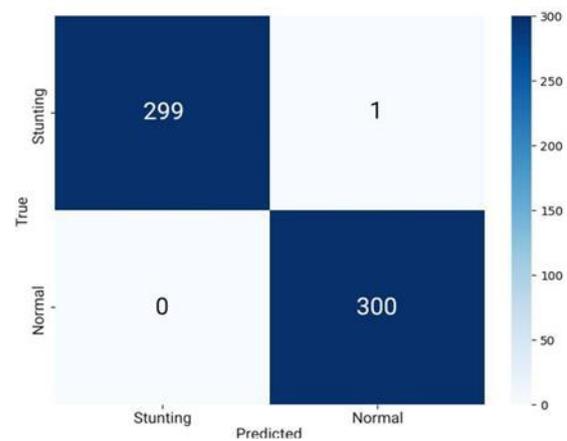
(a) Confusion Matrix MobileNetV2 Early Stopping



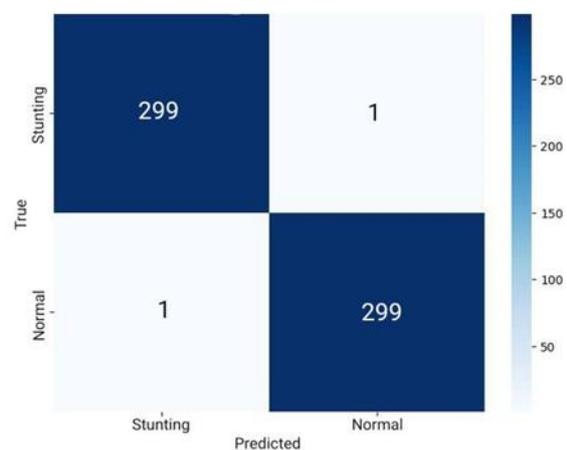
(b) Confusion Matrix InceptionV3 Early Stopping



(c) Confusion Matrix VGG19 Early Stopping



(d) Confusion Matrix ResNet18 Early Stopping



(e) Confusion Matrix EfficientNetB0 Early Stopp

(f) Confusion Matrix AlexNet Early Stopping

Figure 5. Confusion Matrix of All Architectures

Table 2 presents the results of using MobileNetV2, InceptionV3, VGG19, ResNet18, EfficientNetB0, and AlexNet without and with the implementation of early stopping. The EfficientNetB0 and EfficientNetB0 early stopping architectures produce superior accuracy compared to other architectures, with both accuracy values reaching 100%. Here, are the advantages of implementing early stopping; at epoch 13, the model training was stopped because the resulting accuracy is maximum. Meanwhile, those who do not apply early stopping will continue to be in the learning process so that the resulting graph experiences overfitting even though the accuracy reaches 100%. The superiority of EfficientNetB0 can be attributed to the use of a joint scaling technique, which optimizes the depth, width, and resolution of the model simultaneously, allowing it to capture relevant facial features at multiple scales very effectively.

Table 2. Performance of all models

Architecture	Accuracy	Precision	Recall	F1-score
MobileNetV2	96,83%	96%	97,62%	96,8%
MobileNetV2 Early Stopping	99,83%	99,67%	100%	99,83%
InceptionV3	98,83%	99,33%	98,34%	98,83%
InceptionV3 Early Stopping	99,5%	100%	99%	99,5%
VGG19	99,66%	99,66%	99,66%	99,66%
VGG19 Early Stopping	99,83%	99,67%	00%	99,83%
ResNet18	99,83%	100%	99,67%	99,83%
ResNet18 Early Stopping	99,83%	99,67%	100%	99,83%
EfficientNetB0	100%	100%	100%	100%
EfficientNetB0 Early Stopping	100%	100%	100%	100%
AlexNet	99,83%	99,67%	100%	99,83%
AlexNet Early Stopping	99,67%	99,67%	99,67%	99,67%

In addition, ResNet18 and AlexNet achieve the same accuracy of 99.83%. However, there are several advantages and several important differences between each model. ResNet18 has a major advantage in its use of Residual Learning, which allows the network to overcome the degradation problem as the network depth increases. This residual approach allows information to pass through several layers without affecting performance, allowing ResNet18 to be used on deeper networks without the risk of overfitting [15].

Meanwhile, AlexNet is one of the first convolutional neural networks to change the paradigm of computer vision by introducing several innovations, such as the ReLU activation function that speeds up the training process and the use of dropout to prevent overfitting. The advantages of AlexNet lie in the simplicity of its design and its ability to classify images with a relatively smaller number of parameters compared to other models at the time [16]. VGG produces an accuracy of 99.6%. The advantages of VGG19 lie in its

very deep architecture, which consists of many convolutional layers [17]. In addition, VGG19 has a large capacity, so it can learn complex and detailed feature representations from images of children's faces.

The InceptionV3 architecture produces an accuracy of 98.83%, showing good performance but could be more optimal in capturing specific stunting features. MobileNetV2 architecture produces the lowest accuracy compared to other architectures, namely 96.83%; MobileNet proves its ability to recognize facial stunting features even though MobileNet is designed for efficiency on mobile devices. With high accuracy, precision, recall, and F1 score, it shows good performance in classifying data related to stunting and normal conditions. However, it is important to note that our study has several limitations, such as the size and diversity of the dataset used, which may affect the results. It is crucial to acknowledge these limitations to ensure the transparency and reliability of our findings.

4. Conclusions

Based on the results of the study involving various popular CNN architectures such as MobileNetV2, InceptionV3, VGG19, ResNet18, EfficientNetB0, and AlexNet, as well as the implementation of early stopping in the classification of facial images of stunted and normal children, significant results were obtained. The learning hyperparameters used include epoch 100, the optimizer used is SGD, batch size 32, learning rate 0.001, and binary cross-entropy loss function. From the evaluation conducted, EfficientNetB0 and EfficientNetB0 with early stopping are proven to produce the best performance compared to other architectures. The evaluation metrics for both models show perfect accuracy, precision, recall, and F1 score, which is 100%. This shows that the model is able to classify images very well. The success of using early stopping in improving model performance also indicates that this method is effective in avoiding overfitting in the model. For further research, it is recommended to add more images and utilize facial landmark features to improve the performance of the classification algorithm. In addition, further exploration of more effective and efficient CNNs is expected to produce a model that is superior in facial image classification tasks.

Acknowledgements

The research is funded by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, under Penelitian Disertasi Doktor, with Grant No. 094/E5/PG.02.00.PL/2024

References

- [1] J. Liu, J. Sun, J. Huang, and J. Huo, "Prevalence of malnutrition and associated factors of stunting among 6–23-

- month-old infants in Central Rural China in 2019,” *International journal of environmental research and public health*, vol. 18, no. 15, p. 8165, 2021, Accessed: Jul. 16, 2024. [Online]. Available: <https://www.mdpi.com/1660-4601/18/15/8165>
- [2] P. Ghadekar, T. Adsare, N. Agrawal, T. Dharmik, A. Patil, and S. Zod, “Malnutrition Detection Analysis and Nutritional Treatment Using Ensemble Learning,” in *Artificial Intelligence of Things*, vol. 1929, R. K. Challa, G. S. Aujla, L. Mathew, A. Kumar, M. Kalra, S. L. Shimi, G. Saini, and K. Sharma, Eds., in Communications in Computer and Information Science, vol. 1929. Cham: Springer Nature Switzerland, 2024, pp. 296–310. doi: 10.1007/978-3-031-48774-3_21.
- [3] T. R. Gadekallu, C. Iwendu, C. Wei, and Q. Xin, “Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning,” *IET Image Process*, vol. 16, pp. 647–658, 2021, Accessed: Aug. 12, 2024. [Online]. Available: <https://pdfs.semanticscholar.org/1f59/e54001177de23c9306c4e194996825adc7be.pdf>
- [4] A. R. Lakshminarayanan, B. Pavani, V. Rajeswari, S. Parthasarathy, A. A. A. Khan, and K. J. Sathick, “Malnutrition detection using convolutional neural network,” in *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*, IEEE, 2021, pp. 1–5. Accessed: Jul. 16, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9445188/>
- [5] H. MohammedKhan, M. Balvert, C. Guven, and E. Postma, “Predicting Human Body Dimensions from Single Images: a first step in automatic malnutrition detection,” in *CAIP 2021: Proceedings of the 1st International Conference on AI for People: Towards Sustainable AI, CAIP 2021, 20-24 November 2021, Bologna, Italy*, European Alliance for Innovation, 2021, p. 48. Accessed: Jul. 17. <http://dx.doi.org/10.4108/eai.20-11-2021.2314166>
- [6] V. G. Biradar and K. K. Naik, “Detection of Malnutrition in Children Using Deep Learning Model,” in *Smart Trends in Computing and Communications*, T. Senjyu, C. So-In, and A. Joshi, Eds., Singapore: Springer Nature, 2024, pp. 35–45. doi: 10.1007/978-981-97-1323-3_4.
- [7] B. Jin, L. Cruz, and N. Gonçaves, “Deep facial diagnosis: deep transfer learning from face recognition to facial diagnosis,” *IEEE Access*, vol. 8, pp. 123649–123661, 2020, Accessed: Sep. 03, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9127907/>
- [8] T. R. Gadekallu, C. Iwendu, C. Wei, and Q. Xin, “Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning,” *IET Image Process*, vol. 16, pp. 647–658, 2021, Accessed: Sep. 03, 2024. [Online]. Available: <https://pdfs.semanticscholar.org/1f59/e54001177de23c9306c4e194996825adc7be.pdf>
- [9] Y. Yunidar, R. Roslidar, M. Oktiana, Y. Yusni, N. Nasaruddin, and F. Arnia, “Classification of stunted and normal children using novel facial image database and convolutional neural network,” *Radioelectronic and Computer Systems*, vol. 2024, no. 1, pp. 76–86, 2024, Accessed: Jul. 17, 2024. [Online]. Available: <http://nti.khai.edu/ojs/index.php/reks/article/view/reks.2024.1.07>
- [10] A. Singh, H. Herunde, and F. Furtado, “Modified Haar-cascade model for face detection issues,” *International journal of research in industrial engineering*, vol. 9, no. 2, pp. 143–171, 2020, Accessed: Sep. 02, 2024. [Online]. Available: https://www.riejournal.com/article_107190.html
- [11] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, “Facial recognition using Haar cascade and LBP classifiers,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330–335, Nov. 2021, doi: 10.1016/j.gltp.2021.08.044.
- [12] G. Y. Kimura, D. R. Lucio, A. S. Britto Jr., and D. Menotti, “CNN Hyperparameter tuning applied to Iris Liveness Detection.” arXiv, Feb. 12, 2020. Accessed: Sep. 03, 2024. [Online]. Available: <http://arxiv.org/abs/2003.00833>
- [13] A. Nurhopipah and N. A. Larasati, “CNN hyperparameter optimization using random grid coarse-to-fine search for face classification,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 19–26, 2021, Accessed: Sep. 03, 2024. [Online]. Available: <https://kinetik.umm.ac.id/index.php/kinetik/article/view/11855>
- [14] A. Mumuni and F. Mumuni, “Data augmentation: A comprehensive survey of modern approaches,” *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.
- [15] E. Goceri, “Medical image data augmentation: techniques, comparisons and interpretations,” *Artif Intell Rev*, vol. 56, no. 11, pp. 12561–12605, Nov. 2023, doi: 10.1007/s10462-023-10453-z.
- [16] G. Hu *et al.*, “When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 142–150. Accessed: Sep. 03, 2024. [Online]. Available: https://www.cv-foundation.org/openaccess/content_iccv_2015_workshops/w11/html/Hu_When_Face_Recognition_ICCV_2015_paper.html
- [17] “Mathematics | Free Full-Text | An Efficient Optimization Technique for Training Deep Neural Networks.” Accessed: Sep. 03, 2024. [Online]. Available: <https://www.mdpi.com/2227-7390/11/6/1360>
- [18] M. Moreira and E. Fiesler, “Neural networks with adaptive learning rate and momentum terms,” 1995, Accessed: Sep. 03, 2024. [Online]. Available: <https://infoscience.epfl.ch/record/82307/files/95-04.pdf>
- [19] T. S. Akheel, V. U. Shree, and S. A. Mastani, “Stochastic gradient descent linear collaborative discriminant regression classification based face recognition,” *Evol. Intel.*, vol. 15, no. 3, pp. 1729–1743, Sep. 2022, doi: 10.1007/s12065-021-00585-y.
- [20] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.
- [21] H. Choi, D. Choi, and H. Lee, “Early Stopping Based on Unlabeled Samples in Text Classification,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 708–718. doi: 10.18653/v1/2022.acl-long.52.