ISSN 2085-4552

Implementation of Deep Learning Model for Identification of Skin Diseases by Utilizing Convolutional Neural Network

Lysa Apriani [1], Muhamad Bahrul Ulum [2]
[1,2]Informatics, Esa Unggul University, Jakarta, Indonesia
lysapriani@gmail.com[1], m.bahrul_ulum@esaunggul.ac.id[2]

Accepted 29 October 2024
Approved 15 November 2024

***Abstract*— Skin diseases are health problems that affect many individuals worldwide. Rapid and accurate diagnosis of skin diseases is essential for effective treatment. In an effort to improve diagnosis, information technology and artificial intelligence have taken on increasingly significant roles. This study focuses on the implementation of deep learning models for skin disease identification using CNN architectures EfficientNetB0, Xception and VGG16. The models were trained and tested on a dataset of 1800 images with 5 dermatitis classes and 1 normal class. Confusion matrices were used to assess the performance of the three deep learning models on the components of accuracy, recall, precision, and F1-score. The results of the deep learning model that can classify dermatitis skin diseases with a performance of more than 90% for each evaluation matrix are deep learning models utilizing EfficientNetB0 transfer learning with an accuracy of 93%. In contrast, the Xception model indicates overfitting with a training accuracy of 99.96% and a validation accuracy of 86.38%. The VGG16 model indicates underfitting with a training accuracy of 69.71% and a validation accuracy of 46.79%.**

Index Terms*— Skin Disease Classification; CNN; Transfer Learning; EfficientNetB0; Xception; VGG16.

I. INTRODUCTION

The integumentary system serves as the exterior barrier that safeguards the human body against external environmental factors. The skin, being the outermost epithelial organ in the human body, is very vulnerable to diseases due to its direct exposure to the external environment, which harbors a significant amount of pollutants, germs, and viruses. Epidermal illness, often known as dermatosis, is a pathological disorder that impacts the human skin. These skin diseases can arise due to a multitude of circumstances, encompassing infections, allergies, autoimmune disorders, genetic predisposition, environmental influences, and other contributing variables. Contact dermatitis, actinic curatory, neoplasms, dermatophytosis, acne and granulomas are various skin diseases that surrender certain workers to almost 70-95% [1]. Dermatitis with a prevalence of 10% of cases in the world is an inflammation or skin disease that causes clinical abnormalities in the form of polymorphic collation and itching complaints [2]. Dermatitis, particularly contact dermatitis, constitutes a substantial proportion of dermatologist consultations, accounting for around 4-7% of cases. The prevalence of hand dermatitis is high, with a reported incidence rate of 2% among the general population. Indeed, a 20% prevalence rate of hand dermatitis is observed among women, occurring at least once during their lifetime. Furthermore, the findings from the patch test indicated that 30% of children diagnosed with dermatitis exhibited possible allergens[3]. The data obtained shows a significant increase in dermatitis cases in Indonesia from year to year. In 2019, the percentage of dermatitis incidence reached 60.79%[4].

People without a medical background tend to ignore the early manifestations of skin diseases because of the difficulty in distinguishing one type of skin disease from another. The provision of information that is common on the internet is often inadequate in providing adequate guidance for patients trying to understand the condition of their skin disease. Even though this type of skin disease can be cured today, these diseases have indeed brought problems to the patient's life[5]. Indonesia's Health Profile 2022 shows that restrictions on community activities due to the COVID-19 pandemic in 2022 have hampered efforts to detect early cases of skin diseases [6]. Proper identification of skin diseases is important in order to provide appropriate treatment. At this time, technological developments have taken a big role in the health sector, including the use of deep learning in classifying skin disease types based solely on images.

Previous research comparing CNN, RFC, SVM and KNN algorithms resulted in the CNN algorithm having the highest accuracy of 97.89%, followed by RFC with 87.43% accuracy, SVM with 78.61% accuracy and KNN with 76.96% accuracy[7], conducting research on the development of a CNN model in detecting kulti diseases without utilizing transfer learning resulting in an accuracy of 73%[8], research on large class classifications will cause the model to have many class options so that to correctly classify the class is smaller[9], comparing EfficientNet, ResNet, and VGG

in classifying skin diseases with the result that EfficientNet is the model with the best performance reaching an accuracy of 87.31%[10], using the transfer learning models ResNet50, Inception-V3, Inception-ResNet, DenseNet, MobileNet, and Xception to classify skin diseases with Xception results showing the best accuracy at 97% followed by MobileNet (96%), Inception-ResNet (5%), Inception-V3 (95%), DenseNet (93%) and ResNet50 (87%)[11].

Research reports that the accuracy of skin disease detection using image processing and CNN ranges from 70% to 95% [12]. This study aims to identify skin diseases using a convolutional neural network algorithm by utilizing CNN architectures, namely EfficientNet, Xception and VGG16. This study will compare the performance of each CNN architecture using components in the confusion matrix so that it can provide the best performance architectural results. By incorporating deep learning technology, it is hoped that the results of this study can help in making more accurate and efficient identification of skin diseases, as well as making an important contribution to better diagnosis efforts and more efficient management of skin diseases overall

II. METHOD

The research method in this study consists of dataset collection, data pre-processing, CNN modeling, training, and performance evaluation. The flow of the research methodology can be seen in figure 1.

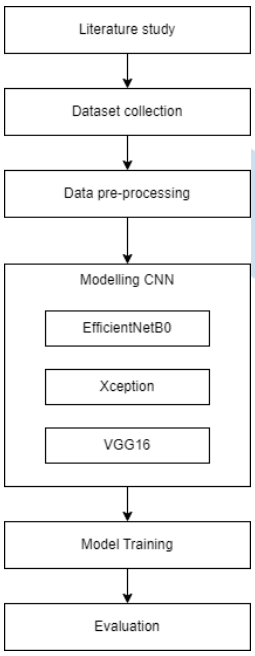

Fig. 1. Research Methods

A. Literature Studies

The literature study stage includes a comprehensive review of relevant scientific literature obtained from websites, articles and journals with the science of skin disease classification.

B. Dataset Collection

At this stage, data collection from other sources was carried out, datasets of allergic contact dermatitis, atopic dermatitis, perioral dermatitis, seborrheic dermatitis, and neurodermatitis used from [13] and the normal class dataset used comes from[14].

The dataset used has the format of *Joint Photographic Experts Group (jpeg/jpg).* This dataset consists of 1800 images with 300 images each for each class. Table I shows the distribution of datasets and classes that will be used in this study.

Table 1. Dataset Distribution

No	Class	Sum
1	Allergic Contact Dermatitis	300
2	Atopic Dermatitis	300
3	Seborrheic Dermatitis	300
4	Dermatitis Perioral	300
5	Neurodermatitis	300
6	Normal	300

C. Data Pre-processing

In the processing process, several things will be done, including changing the size of the image, augmenting data, and distributing the dataset. The image resizing is carried out because the dataset has a different size, so the image size is changed before the process so that the data has the same size so that the model can receive appropriate input. In this study, the image size was changed to 224 x 224 pixels.

Fig. 2 Data augmentation

Data augmentation is a way to multiply the number of images[15]. Augmentation is a technique that aims to enhance the quantity and variety of training data by strategically applying specific alterations to pre-existing data. To enhance the generalization of the model, several augmentations will be implemented. These include rotation with a maximum value of 0.7 for both right and left rotation, flip right and left with a value of 0.5, flip up and down with a value of 0.5, zoom random with a value of 0.5, crop random with a value of 0.5, random brightness with a value of 0.5, and random contrast with a value of 0.5.

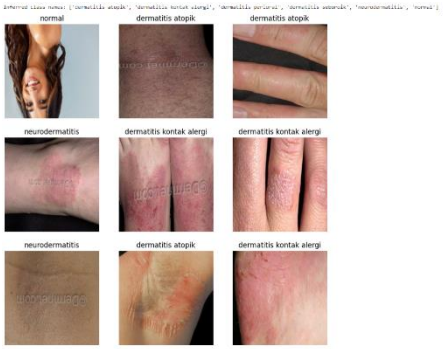

Fig. 3 Examples of images that have been augmented

The process of distributing the training, validation, and test datasets involves a method designed to carefully divide the dataset into distinct segments for training, testing, and validation, while ensuring that the proportion of each class is consistently maintained across all sections. This approach is crucial for preserving the integrity and balance of the dataset, allowing the model to learn effectively from the training data and be accurately evaluated on the test and validation sets. In the context of this study, the dataset has been allocated in a manner where 80% of the data is used for training the model, 10% is reserved for testing its performance, and the remaining 10% is dedicated to validating the model's accuracy and generalization capabilities. This proportional distribution is intended to optimize the model's learning process and ensure robust and reliable results.

D. Modelling CNN

Following the pre-processing of the dataset, the subsequent step involves the development of a high-performance deep learning model. The modeling procedure involves conducting multiple experimental iterations using various hyperparameters in order to identify the model that yields the highest level of accuracy. By iteratively optimizing accuracy through trial and error to identify the hyperparameters that yield optimal outcomes.

Figure 4 illustrates the CNN architecture employed in the present investigation. The employed architectural design has five distinct layers, specifically the layer base model, batch normalizing step, dense layer, dropout layer, and fully linked layer.

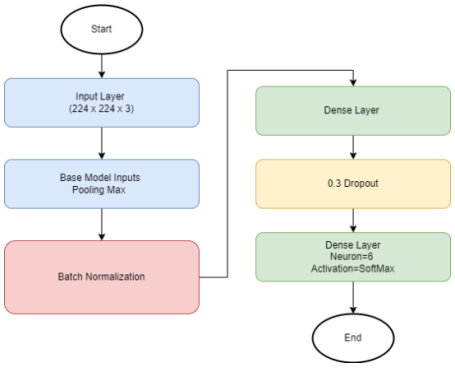

Fig. 4 Applied CNN architecture

The hyperparameters used in this study are shown in table II. EfficientNetB0, VGG16 and Xception were chosen as CNN architectures for detecting dermatitis skin disease classes. The performance of the three architectures will be compared to find the best architecture. The data is split into 80% for training, 10% for validation, and 10% for testing. With a batch size of 64, this means that if there are 4800 training samples, the CNN algorithm will take 64 data samples from the total 4800 available. These samples will then be trained by the neural network until complete, after which it will take the next 64 samples, continuing this process until all data is processed. While 5 layers means the architecture consists of 5 layers that have been explained in figure 4.1. Epoch 25 means the training process is carried out as many as 25 iterations or repetitions which are then applied earlystopping callbacks that can stop training if according to the defined metrics, then there is adamax optimizer which is used to iteratively improve weights based on training data, and the use of learning rate of 0.001 where the smaller the learning rate, the model will learn the training data in more detail, the learning rate also applies reduce_lr callbacks which will reduce the learning rate if the defined metrics are met.

Table 2. Hyperparameters

No	Hyperparameter	
1	CNN Architecture	EfficientNetB0, Xception, VGG16
2	Batch Size	64
3	Epoch	25
4	Optimizer	Adamax
5	Learning rate	0.001

E. Training

At this stage, the training stage is carried out with the dataset owned on the selected architecture. The training process is used using training data and updating the model using optimization algorithms. In the training model, callbacks such as

'ReduceLROnPlateau', 'ModelCheckpoint' and 'EarlyStopping' are applied which are used to maximize the model training process. ReduceLROnPlateau is used to reduce the learning rate when the monitored metric, namely 'val_loss', does not improve. ModelCheckpoint is used to store the best model based on the best 'val_accuracy' metric or validation accuracy, ensuring the most optimal model is not lost during training. EarlyStopping is used to stop training or training if the monitored metric 'val_accuracy' stops increasing, this aims to prevent overfitting and save computing resources. By implementing these callbacks, the training process will become more efficient and adaptive to validation performance.

```
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=1e-5)
model_checkpoint = ModelCheckpoint('model-Model-New.h5', monitor='val_accuracy', mode='max', verbose=1, save_best_only=True)
early_stopping = EarlyStopping(monitor='val_accuracy', patience=5, restore_best_weights=True)

history = model.fit(train_ds, validation_data=val_ds, epochs=initial_epochs, callbacks=[model_checkpoint, reduce_lr])
```

Fig. 5 Application of callbacks in model training

F. Evaluation

This study will analyze the work of the classification model by using the confusiation matrix as the core evaluation instrument. The confusion matrix is an evaluation tool that provides a detailed overview of the performance of a classification model that shows how correct a model is in grouping data [16]. The confusion matrix is shown in table 3.

Table 3 Confusion Matrix

Data Classes	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

True Positive (TP) indicates a correct prediction when the model accurately classifies data as positive, and the data is actually positive. True Negative (TN) represents a correct prediction when the model correctly identifies data as negative, and the data is indeed negative. On the other hand, False Positive (FP) occurs when the model incorrectly classifies negative data as positive, while False Negative (FN) happens when the model incorrectly labels positive data as negative. These components of the confusion matrix are essential for calculating key metrics such as accuracy, precision, recall, and F1-Score, which are used to assess and compare the performance of the model.

Equation (1) shows the formula for calculating accuracy.

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

Equation (2) shows the precision calculation formula.

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

Equation (3) shows the recall calculation formula.

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

Equation (4) shows the formula for calculating the F1-score.

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

III. Results and Discussions

In this section, it will be explained in detail how the results have been obtained, starting from the results of model training, and the results of the confusion matrix test. Table 4 shows the results of the training process of each archivist.

Table 4 Model Training Results

No	Architecture	Acc	Val Acc	Loss	Val Loss
1	EfficientNetB0	0.99	0.93	0.03	0.24
2	Xception	0.99	0.86	0.54	0.84
3	VGG16	0.69	0.46	0.86	1.31

EfficietNetB0 performs very well in testing and training. With a validation accuracy of 0.9987 and a training accuracy of 0.9343, this model shows that it can effectively learn patterns from training data and generalize them to new data (data validation).

Xception shows signs of overfitting. Given the very high validation accuracy (0.9996) and lower training accuracy (0.8638), it seems that this model may be overly dependent on the training set and unsuitable for generalization. The much larger loss value in the validation data (0.8412) compared to the training data (0.5400) further supports this.

VGG16 shows overfitting and gives substandard results. The model's poor training accuracy (0.4679) and low validation accuracy (0.6971) indicate that the model has difficulty generalizing previously unknown data and learning patterns from the training set. It can be seen from the high loss value in the validation data (1.3071) and training data (0.8697) that this model is not able to maximize learning.

Macro averages consider precision, recall, and F1-score for each class individually, without applying any weights based on the class distribution. On the other hand, the weighted average also takes into account precision, recall, and F1-score for each class but incorporates weights that correspond to the size or importance of each class. This means that in a weighted average, classes with more samples or higher relevance have a greater influence on the overall metric.[17]. Table 5 presents the confusion matrix outcomes derived from the calculations of three models.

Table 5 Confusion matrix calculation results

No	Architecture	Acc	Macro Avg			Weighted Avg		
			Precision	Recall	F1-score	Precision	Recall	F1-Score
1	EfficientNetB0	0.93	0.93	0.93	0.92	0.93	0.93	0.92
2	Xception	0.86	0.84	0.84	0.83	0.84	0.85	0.84
3	VGG16	0.46	0.62	0.49	0.47	0.49	0.47	0.47

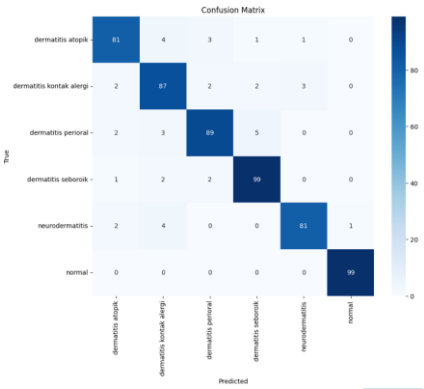

Fig. 6 EfficientNetB0 confusion matrix results

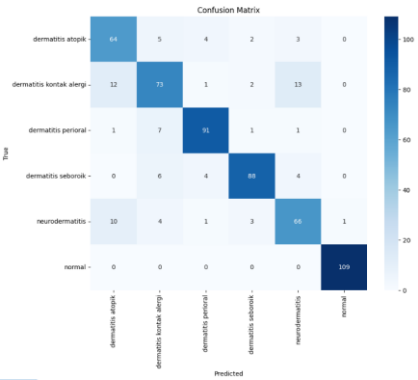

Fig. 7 Xception confusion matrix results

Figure 6 show the confusion matrix of EfficientNetB0 model. EfficientNetB0 model performs very well in both testing and training. This model is not overfitting, as evidenced by the low loss values on training and validation data. The precision, recall, and F1-Score values for Macro Average are 0.93, 0.93, and 0.92 respectively, then for the precision, recall, and F1-Score values for Weighted Average are 0.93, 0.93, and 0.92 respectively, which are consistently the same between the two. This approach not only concentrates on the majority class but also effectively manages the minority class, as seen from the high macro and weighted average performance that is constant across all classes. This shows that the model is reliable and has strong generalization across various types of data in the dataset.

Figure 7 show the confusion matrix of Xception model. The Xception model shows signs of overfitting. The macro average values for precision, recall and F1-score are 0.84, 0.84, and 0.83 respectively and the precision, recall and f1-score values for the weighted average are 0.84, 0.85, and 0.84.

The decrease in the weighted average and macro values indicates that the performance of this model is not evenly distributed across all classes. The reason why the performance of this model is inconsistent across datasets may be that this model over-learns the majority class and ignores the minority class.

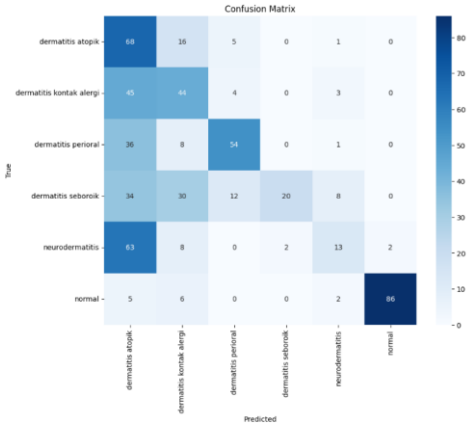

Fig. 8 VGG16 confusion matrix results

Figure 8 show the confusion matrix of VGG16 model. The VGG16 model shows overfitting and produces subpar results. The precision, recall, and F1-score values for the macro average are 0.62, 0.49, and 0.47, respectively, which are lower than the weighted average values of 0.49, 0.47, and 0.47, respectively. The very low weighted average and macro values indicate that the model's performance is not balanced across all classes and cannot manage classes that are in the majority or not.

IV. Conclusion

Based on the findings from the study on the implementation of Deep Learning models for skin disease identification using Convolutional Neural Networks (CNNs), it can be concluded that the process involves several key stages. These stages include data collection, image pre-processing, integrating transfer learning models, batch normalization, adding dense layers, incorporating dropout layers, and final dense layers. The study applied three transfer learning models—EfficientNetB0, Xception, and VGG16—utilizing 1800 images, which were augmented to expand the training dataset.

The performance of the skin disease classification model was evaluated using a confusion matrix, where the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values played a crucial role in calculating metrics such as recall, precision, and F1-Score. These metrics were essential in determining the model's accuracy. Among the various experiments conducted (A to C), the best result was achieved in experiment A using the EfficientNetB0 model, which obtained a training accuracy of 93%. Validation using the confusion matrix confirmed this performance, yielding an accuracy of 93%, precision of 93%, recall of 93%, and an F1-Score of 92%.

For further development, it is imperative to conduct research employing diverse transfer learning architectures. Additionally, expanding the dataset size to encompass a higher number of instances can yield improved accuracy outcomes. Furthermore, the utilization of convolutional neural network (CNN) architectures with varying hyperparameters can effectively mitigate the issues of overfitting and underfitting

References

[1] J. S. Park, E. K. Park, H. K. Kim, and G. S. Choi, "Prevalence and risk factors of occupational skin disease in Korean workers from the 2014 Korean working conditions survey," *Yonsei Med. J.*, vol. 61, no. 1, pp. 64–72, 2020, doi: 10.3349/ymj.2020.61.1.64.

[2] A. S. Maudani, M. Ikhtiar, and A. Baharuddin, "Analisis Spasial Penyakit Dermatitis di Puskesmas Labakkang Kabupaten Pangkep," *Ikesma*, vol. 16, no. 1, p. 51, 2020, doi: 10.19184/ikesma.v16i1.16998.

[3] R. Apriliani, S. Suherman, E. Ernyasih, N. Romdhona, and M. Fauziah, "Hubungan Personal Hygiene Dengan Kejadian Dermatitis Kontak Iritan Pada Pemulung Di Tpa Bantargebang," *Environ. Occup. Heal. Saf. J.*, vol. 2, no. 2, p. 221, 2022, doi: 10.24853/eohjs.2.2.221-234.

[4] G. Soegiarto, M. S. Abdullah, L. A. Damayanti, A. Suseno, and C. Effendi, "The prevalence of allergic diseases in school children of metropolitan city in Indonesia shows a similar pattern to that of developed countries," *Asia Pac. Allergy*, vol. 9, no. 2, p. e17, 2019, doi: 10.5415/apallergy.2019.9.e17.

[5] L. S. Wei, Q. Gan, and T. Ji, "Skin Disease Recognition Method Based on Image Color and Texture Features," *Comput. Math. Methods Med.*, vol. 2018, 2018, doi: 10.1155/2018/8145713.

[6] Kemenkes RI, *Profil Kesehatan Indo-nesia*. 2022. [Online]. Available: https://www.kemkes.go.id/downloads/resources/download/pusdatin/profil-kesehatan-indonesia/Profil-Kesehatan-2021.pdf

[7] T. A. Rimi, N. Sultana, and M. F. Ahmed Foysal, "Derm-NN: Skin Diseases Detection Using Convolutional Neural Network," *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. Iciccs, pp. 1205–1209, 2020, doi: 10.1109/ICICCS48265.2020.9120925.

[8] J. Boman and A. Volminger, "Evaluating a deep convolutional neural network for classification of skin cancer Evaluating a deep convolutional neural network for classification of skin cancer," 2018.

[9] A. Rafay and W. Hussain, "EfficientSkinDis: An EfficientNet-based classification model for a large manually curated dataset of 31 skin diseases," *Biomed. Signal Process. Control*, vol. 85, no. March, p. 104869, 2023, doi: 10.1016/j.bspc.2023.104869.

[10] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and M. M. Rahman, "An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis," *Healthc. Anal.*, vol. 3, no. January, 2023, doi: 10.1016/j.health.2023.100143.

[11] Bergi Veeresha Gowda, Likith Kumar S, Gadilingappa G V, K. Likhith, and G Sai Ranga, "Skin Disease Detection using Image Processing and CNN," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 286–295, 2023, doi: 10.48175/ijarsct-9751.

[12] M. Groh *et al.*, "Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1820–1828, 2021, doi: 10.1109/CVPRW53098.2021.00201.

[13] D. Bala *et al.*, "MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification," *Neural Networks*, vol. 161, pp. 757–775, 2023, doi: 10.1016/j.neunet.2023.02.022.

[14] D. Putri Ayuni, Jasril, M. Irsyad, F. Yanto, and S. Sanjaya,

"Augmentasi Data Pada Implementasi Convolutional Neural Network Arsitektur Efficientnet-B3 Untuk Klasifikasi Penyakit Daun Padi," *Zo. J. Sist. Inf.*, vol. 5, no. 2, pp. 239–249, 2023, doi: 10.31849/zn.v5i2.13874.

[15] L. Qadrini, A. Sepperwali, and A. Aina, "Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial," *J. Inov. Penelit.*, vol. 2, no. 7, pp. 1959–1966, 2021.

[16] D. Morales, E. Talavera, and B. Remeseiro, "Playing to distraction: towards a robust training of CNN classifiers through visual explanation techniques," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 16937–16949, 2021, doi: 10.1007/s00521-021-06282-2.