

Application of EfficientNetV2-S Architecture with Focal Loss to Overcome Class Imbalances in Skin Cancer Classification

Marfungah Wati¹, Aris Thobirin², Sugiyarto Surono³
^{1,2,3} Department of Mathematics, Ahmad Dahlan University, Indonesia

Article Info

Article history:

Received Jan 29, 2026

Revised Mar 23, 2026

Accepted Apr 08, 2026

Keywords:

EfficientNetV2-S

Focal Loss

Class imbalance

Skin Cancer

dermoscopic images

ABSTRACT

Imbalanced class distributions in skin lesion image datasets can reduce the effectiveness of multiclass classification models. This research proposes a classification model based on the EfficientNetV2-S architecture with the application of two-stage training and loss functions that emphasize learning in classes with limited data. The models were trained using on-the-fly image augmentation and evaluated to assess generalization capabilities to the test data. In the initial stage, the model is trained by freezing the backbone and only updating the classifier layer. Next, fine-tuning was carried out on part of the backbone layer to adjust the representation of features to the image characteristics of the skin lesion. Evaluation is conducted through multiple training times with different random initializations to ensure consistency of results. The test results showed that the model experienced an improvement in performance after the fine-tuning process, with an accuracy of about 88% as well as an increase in F1-score values in some classes. Overall, the results indicate that the proposed approach may help improve classification performance when dealing with imbalanced skin cancer image data.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Marfungah Wati,

Mathematics,

Ahmad Dahlan University,

Jl. Ringroad Selatan, Kragilan, Tamanan, Banguntapan, Bantul, Special Region of Yogyakarta.

Email: 2200015013@webmail.uad.ac.id

1. INTRODUCTION

In recent years, capabilities and applications Artificial Intelligence (AI) has evolved very rapidly [1]. AI is the fastest growing technology, serving as a computing system designed to mimic human activities, thoughts, and perceptions through sophisticated algorithms [2]. Deep learning is a major AI approach and a subset of machine learning, where neural network architectures are used to automatically extract patterns from data autonomously [3]. Deep learning has a high learning ability, as it is able to extract and learn information from large, complex datasets, which are difficult to process manually [4].

In the domain of digital image processing, Convolutional Neural Networks (CNNs) extensively used as architectures for deep learning [5]. CNNs are designed to process pixel-based data through a series of mathematical operations, so that it is able to extract patterns and visual features gradually at each layer of the network [6]. With this approach, CNN can capture the spatial relationships between pixels and study the complex representations of an image effectively [7]. Over the past ten years, CNN has achieved significant results in pattern recognition, image processing, and voice recognition [8]. Its ability to detect visual features that are difficult for humans to identify makes CNN able to provide high accuracy and go beyond traditional approaches [9].

The CNN system also allows for standardization and improvement of the accuracy of medical image interpretation in various healthcare settings [10]. However, the application of this algorithm practically still faces a number of challenges[11], including the scarcity of annotated medical data, limited interpretability, and ethical considerations, which continue to hinder the broader implementation of CNN-based systems in clinical practice [12]. Class imbalance in the dataset is also a problem, which occurs when the sample distribution across classes is uneven, causing one or more classes to have significantly more data than others [13]. To overcome these challenges, a data augmentation method was applied, which added image variety and balanced the class distribution, thereby improving the model's capacity for generalization and enhancing training effectiveness. [14]. However, the increase in the number of image variations also causes the size of the dataset to be larger, making the computational process more cumbersome and training taking longer if done manually [15]. Therefore, the transfer learning It is used because it can make a major contribution to the analysis of medical images, by being able to overcome data limitations while saving time and computing resources[16]. Focal loss is also needed to enhance the model's capability in handling dataset complexity, as this method can reduce the influence of the majority class by down-weighting the loss contribution of the dominant class [17].

Gayatri and Aarth (2023) performed skin lesion classification on the ISIC-2019 dataset using a ResNet-50 architecture with static augmentation and focal loss to address class imbalance, reporting an accuracy of 98.85%. However, the study focused on a single architecture and augmentation strategy, leaving room for exploration of more recent models and diverse augmentation techniques [18]. Another study by X et al. (2020) used the HAM10000 dataset to develop a modified ResNet-50 with focal loss and class-weighted loss, obtaining an average accuracy of 93%. The study also compared several architectures, including VGG16, MobileNet, and EfficientNet-B1, with ResNet-50 showing the best performance. These works highlight the relevance of transfer learning and specialized loss functions for multiclass skin lesion classification [19].

Focal loss has been shown to help improve model performance in the face of class imbalances. However, previous research generally still used simpler augmentation architectures and techniques. This research presents a new contribution by applying a more modern EfficientNetV2-S architecture and a more diverse on-the-fly augmentation strategy, combined with focal loss to increase the model's in recognizing minority classes in the HAM10000 dataset. This approach is expected to result in a more stable model and be able to make better generalizations in the classification of multiclass skin lesions.

2. RESEARCH METHOD

2.1 Research Flow

Figure 1 illustrates the overall research workflow.

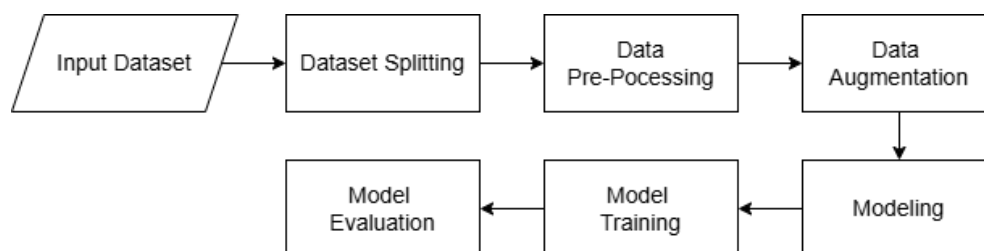


Figure 1. Research Process Flow

2.2 Dataset

This study uses the HAM10000 dataset (Human Against Machine with 10,000 training images) introduced by Tschandl et al. (2018) [20]. The dataset includes 10,015 dermoscopic images grouped into seven diagnostic categories: melanocytic nevi (NV), melanoma (MEL), benign keratosis-like lesions (BKL), basal cell carcinoma (BCC), actinic keratosis (AKIEC), dermatofibroma (DF), and vascular lesions (VASC).

In this research, the dataset was accessed through the Kaggle platform via a redistributed version uploaded by Teguh Budi Setya Prasetya ([datasets/teguhbudi/cancer-skin](https://www.kaggle.com/datasets/teguhbudi/cancer-skin)). The researchers verified that the total number of images (10,015), the number of classes (7 classes), and the distribution of each class corresponded to the official release, without any modification of the label or image content. Example images from the dataset are presented in Figure 2.

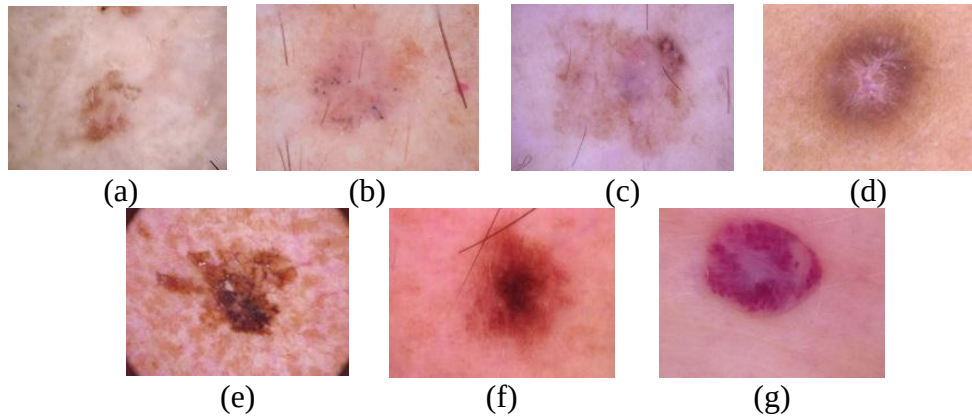


Figure 2. Skin Cancer Image Data Sample
(a) AKIEC, (b) BCC, (c) BKL, (d) DF, (e) MEL, (f) NV, and (g) VASC

2.3 Dataset Splitting

The dataset included seven classes of dermoscopic images and was divided into training, validation, and testing subsets with a 70:15:15 ratio using stratified sampling. The dataset was split based solely on class labels without considering patient identities, which constitutes a limitation of this study. In the HAM10000 dataset, multiple images may originate from the same patient or lesion, and this splitting strategy may introduce data leakage. As a result, the model may inadvertently learn patient-specific features rather than generalizable disease patterns, potentially leading to overoptimistic performance estimates. Therefore, the reported results should be interpreted with caution, and future work should consider patient-level data splitting to provide a more reliable evaluation. The distribution of images across subsets is provided in Table 1.

Table 1. Dataset Distribution per Class After Splitting Process

Class	Training	Validation	Test
AKIEC	229	49	49
BCC	360	77	77
BKL	769	165	165
DF	81	17	17
MEL	779	167	167
NV	4.693	1.006	1.006
VASC	99	21	22
Total	7.010	1.502	1.503

2.4 Data Pre - processing

Optimal data pre-processing can improve model accuracy, reduce the risk of overfitting, and speed up training time [21]. Preprocessing was applied to prepare dermoscopic images for training, including resizing to 384×384 pixels, conversion to tensor format, and normalization to match the input requirements of EfficientNetV2-S. Next, a pixel normalization is carried out which is calculated using the equation [22]:

$$I_{norm} = \frac{I_{cropped} - \mu}{\sigma} \quad (1)$$

Where $I_{cropped}$ is the cropped image, I_{norm} is the normalized image, μ and σ are the mean and standard deviation of the pixel values, respectively. The values of μ and σ follow the ImageNet standard, where the mean values are [0.485, 0.456, 0.406], and the corresponding standard deviations are [0.229, 0.224, 0.225].

2.5 Data Augmentation

Image transformations were applied using torchvision to normalize images according to ImageNet standards and to perform real-time augmentation aimed at improving generalization and reducing overfitting [23]. During training, on-the-fly augmentation was applied to each batch to increase data variability and enhance model robustness. The augmentation techniques included RandomResizedCrop, GaussianBlur ($p = 0.2$), Random Flip, Random Rotation ($\pm 30^\circ$), and ColorJitter with brightness, contrast, and saturation set to 0.1 to increase visual diversity while preserving the essential characteristics of skin lesions.

2.6 Modeling EfficientNetV2-S

EfficientNetV2 is a CNN introduced by Google in 2021 as an improved version of the EfficientNet family, offering better computational efficiency and training speed [24][25]. The architecture uses an efficient combination of MBConv and Fused-MBConv blocks to process features [24]. here MBConv (inverted residual block) is designed to enhance computational efficiency [26].

According to [27], the MBConv architecture combines pointwise convolutions for channel transformation and depthwise convolutions (kernel size 3 or 5) for spatial feature extraction, along with batch normalization, Swish activation, squeeze and excitation (SE) attention, and dropout. Furthermore, a SE block adaptively reweights channels features to emphasize important information [28][29], followed by another 1×1 pointwise convolution to restore the channel dimensions (with batch normalization), and finalized with a dropout layer [27]. The pointwise convolution can be expressed as follows [30].

$$PWconv(W, y)_{(i,j)} = \sum_m^M W_m \cdot y_{(i,j,m)} \quad (2)$$

Where $y_{(i,j,m)}$ is the feature's activation at position (i,j) in the m channel, W_m denotes the weight corresponding to the m channel, and $PWconv(W, y)_{(i,j)}$ represents the output at position (i,j) obtained by combining all input channels.

Depthwise separable convolution is an alternative form of conventional convolution that can be used without changing the structure of the network [31]. In the depthwise stage, each input channel is processed independently using its own filter. Mathematically, the process depthwise convolution can be written as follows [30].

$$DWconv(W, y)_{(i,j)} = \sum_{k,l}^{K,L} W_{(k,l)} \cdot y_{(i+k,j+l)} \quad (3)$$

Where $y_{(i+k,j+l)}$ is the input feature value at spatial position $(i+k, j+l)$, $W_{(k,l)}$ denotes the convolution kernel of size $K \times L$, and $DWConv(W, y)_{(i,j)}$ represents the output at position (i,j) . For simplicity, the channel dimension is omitted, but the convolution is applied independently to each channel.

In Fused-MBConv, the two main blocks of MBConv, namely depthwise convolution 3×3 and pointwise expansion 1×1 , are replaced by one standard 3×3 convolution operation. This

approach simplifies the computational process so that training becomes faster without compromising feature extraction capabilities [25][31].

In this study, EfficientNetV2-S pretrained on ImageNet was used as the backbone model. Its original classification component was substituted with a task-specific head that includes a 256-unit dense layer activated by ReLU, along with dropout regularization applied before and after the dense layer ($p = 0.3$ and $p = 0.2$, respectively), and a output layer tuned to the target number of class. The classifier was trained for 20 epochs in Phase 1, followed by fine-tuning of the last three backbone blocks for up to 30 epochs with validation-based early stopping.

2.7 Model Training

2.7.1 Loss Function

2.7.1.1 Weighted Cross Entropy Loss

As a baseline approach, Weighted Cross Entropy (WCE) is used to mitigate class imbalance. Giving greater importance to minority classes helps prevent the model from being overly influenced by majority classes during training. Weighted Cross Entropy is defined in Equation (4) as follows [32].

$$\mathcal{L}_{CWCE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C \beta_c \cdot y_{i,c} \log(p_{i,c}) \quad (4)$$

where N is the total number of samples, C denotes the number of classes, $y_{i,c}$ is the ground truth label for sample i and class c , $p_{i,c}$ is the predicted probability for class c , and β_c represents the weight assigned to class c . The class weights β_c are computed using the same strategy as in focal loss to ensure a fair and consistent comparison.

2.7.1.2 Focal loss

Focal Loss is formulated to mitigate class imbalance issues, especially when the distributions of positive and negative samples [32]. This approach makes the model more focused on difficult samples so that performance in minority classes improves. Mathematically, Focal Loss is defined in Equation (5) as follows [32].

$$\mathcal{L}_{focal} = -\frac{1}{N} \sum_{i=1}^N \beta_t (1 - p_t)^\gamma \log(p_t) \quad (5)$$

In the equation, p_t it states the probability for the correct class, while β_t balances the loss contribution across classes. For $\gamma \geq 0$, the loss function adjusts the weighting between easy and hard samples. When $\gamma = 0$, focal loss is equivalent to weighted cross-entropy [32]. As γ increases, the contribution of easy samples is reduced, causing the model to pay greater attention to samples that are difficult to classify [18]. Based on this principle, this study evaluated several values of γ , namely 1, 1.5, 2, and 5, to determine the most suitable focal loss configuration according to model performance.

The class weights β used in both Weighted Cross-Entropy and Focal Loss were computed based on inverse class frequency. Classes with fewer sample counts gain greater weight. The weight of each class is calculated through [33]:

$$\beta_c = \frac{(1 - f_c)}{\sum_{k=1}^C (1 - f_k)} \quad (6)$$

In this equation, β_c it is the weight of the c-class, f_c is the proportion of the sample of the c-class, and C expresses the number of classes. Adjusting weights by class proportion increases the contribution of minority classes while maintaining training stability through normalization.

2.7.2 Optimizer Adam

Adam is an adaptive optimization method that is widely used as an alternative to traditional stochastic gradient descent [34]. This algorithm combines RMSProp and momentum, with momentum from the first moment of the gradients biased, while the second moment is computed via an exponential moving average [35], as formulated in [36].

Counting the momentum

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (7)$$

Calculating the gradient square

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (8)$$

Since $m_0 = 0, v_0 = 0$, the initial value m_t and v_t at the beginning tends to be biased toward zero. Therefore, bias correction is applied:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (9)$$

Updated parameters using:

$$\hat{\theta}_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (10)$$

Where θ_t denotes the model parameters at iteration t , α is learning rate, g_t is the gradient of the loss function with respect to the parameters at iteration t , m_t and v_t are the first and second moment estimates, $\hat{m}_t$ provides a stable direction (*momentum*), $\hat{v}_t$ determines the magnitude of the step, β_1 and β_2 are the exponential decay rates, and ϵ is a small constant.

In this study, the Adam was used in Phase 1 with a learning rate and weight decay of 1×10^{-4} to train only the classifier. In Phase 2, fine-tuning used learning rates of 3×10^{-5} for the backbone and 3×10^{-4} for the classifier, with weight decay kept at 1×10^{-4} .

2.7.3 Training Strategy

To improve the reliability of the results and minimize the influence of random initialization, each model in the study was trained using three different random seeds. Both the Weighted Cross Entropy and Focal Loss models were run three times with the same experimental setup.

2.8 Evaluation Model

To evaluate the model, a confusion matrix was used to compare predicted labels with actual labels [37]. In presentation, the actual label is placed on the row section, while the prediction label is on the column section [38]. Based on this matrix, model performance was quantified using widely used classification indicators, which are mathematically defined as follows [39]:

$$Accuracy (\%) = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \times 100 \quad (11)$$

$$Precision (\%) = \frac{T_p}{T_p + F_p} \times 100 \quad (12)$$

$$Recall (\%) = \frac{T_p}{T_p + F_n} \times 100 \quad (13)$$

$$F1 - score (\%) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

In class-imbalanced scenarios, the Matthews Correlation Coefficient (MCC) was also used to provide a more comprehensive performance evaluation [18]. The MCC is mathematically defined as follows:

$$MCC = \frac{T_p \times T_n - F_p \times F_n}{\sqrt{(T_p + F_p)(T_p + F_n)(T_n + F_p)(T_n + F_n)}} \quad (15)$$

Here, T_p and T_n represent correct classification outcomes, while F_p and F_n correspond to misclassified samples.

3. RESULTS AND DISCUSSION

Model training is carried out in two stages: Phase 1 training *classifier* with the backbone frozen, and Phase 2 fine-tuned part of the EfficientNetV2-S backbone. Focal Loss is used to overcome class imbalances, with values $\gamma = 1, 1.5, 2$, and 5 evaluated to determine the best configuration shown in the table and used for final parameter selection.

Table 2. Model Performance Evaluation for Various γ Values on Focal Loss

γ	Accuracy (%)	Macro F1-Score	MCC
1	90.00	83.14	80.29
1.5	89.24	82.85	79.48
2	88.45	80.28	77.91
5	88.18	82.57	76.84

Based on Table 2, $\gamma = 1$ showed relatively better performance and was selected for the final evaluation. The value of β was determined adaptively using inverse class frequency in Equation (6), based on the class proportions in the training set so that minority classes received higher loss weights. The β values for each class are presented in the following table 3.

Table 3. Distribution of β Values for Each Class

Classes	Number of Train Samples	Weight (β)
AKIEC	229	0.1612
BCC	360	0.1581
BKL	769	0.1484
DF	81	0.1647
MEL	779	0.1481
NV	4.693	0.0551
VASC	99	0.1643

Weight is given to increase the contribution of the minority class. Focal Loss and Weighted Cross Entropy are trained on the same two-phase scheme as well as settings, and use identical weighting methods to maintain a fair comparison.

The final evaluation was carried out to compare the performance of Focal Loss and WCE. The values reported in Table 4. is an average along with standard deviations from several trainings, thus providing a more stable and consistent picture of model performance.

Table 4. Comparison of WCE and Focal Loss Performance

Loss Function	Accuracy	Macro-F1	Weighted-F1	MCC
Weighted Cross Entropy (WCE)	0.877 ± 0.005	0.798 ± 0.014	0.878 ± 0.006	0.768 ± 0.011
Focal Loss	0.879 ± 0.016	0.811 ± 0.026	0.881 ± 0.015	0.775 ± 0.028

Based on Table 4, Focal Loss shows a slight improvement compared to WCE, particularly in handling class imbalance, as reflected by the improvement in Macro-F1 and MCC. However, when considering the standard deviation values, the observed performance differences are relatively small and likely not statistically significant. This suggests that although Focal Loss provides a small improvement, its advantage over WCE in this study remains limited. These findings also indicate that both loss functions exhibit relatively comparable performance under the experimental conditions used.

3.1. Phase 1 (Classifier Training)

In the first stage, the entire EfficientNetV2-S backbone is frozen so that only the classifier layer is trained. To visualize the dynamics of learning in Phase 1, a graph of train-validation loss and train-validation accuracy is presented in Figure 3.

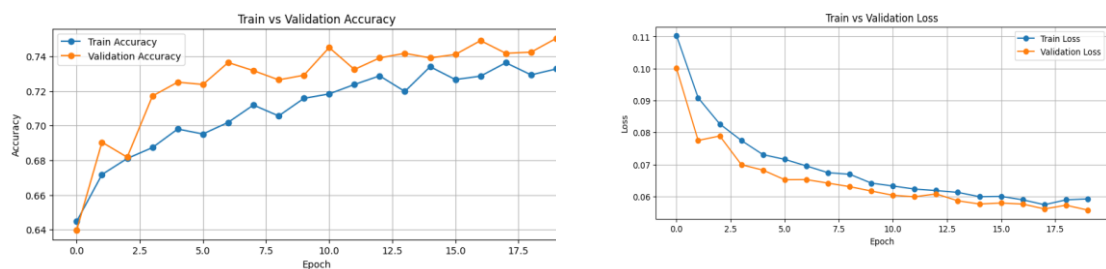


Figure 3. Train vs Validation Accuracy and Loss Chart in Phase 1

Based on Figure 3, in Phase 1 the accuracy increases gradually from about 64% to about 73% – 75% at the end of the epoch, with validation accuracy tending to be slightly higher than train accuracy. The loss value also decreased consistently, from about 0.11 to about 0.059 on the training data and from about 0.10 to about 0.056 on the validation data. This pattern shows a stable learning process with no indication of overfitting at the initial training stage before fine-tuning is performed.

3.2. Phase 2 (Fine Tuning)

Training continued in Phase 2 by fine-tuning the last three blocks of the EfficientNetV2-S backbone. The learning dynamics are shown in the train-validation loss and accuracy curves in Figure 4.

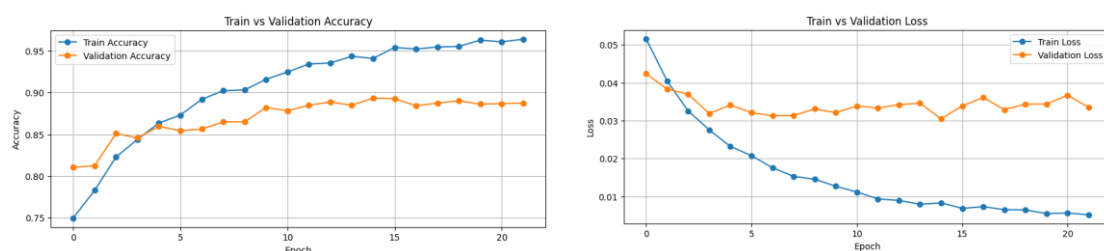


Figure 4. Train vs Validation Accuracy and Loss Chart in Phase 2

In Phase 2, there was a significant increase in train accuracy from around 75% to around 96% at the end of training, while validation accuracy increased and was relatively stable in the range of 88% – 89%. The train loss continues to decline to near 0.005, while the validation loss tends to be stable around 0.03 – 0.035 with slight fluctuations. The observed gap between training and validation performance indicates a tendency toward mild overfitting after fine-tuning. This behavior may be influenced by model complexity and class imbalance, despite the use of data augmentation.

3.3. Model Evaluation Results

The model was evaluated on the test set in two training phases. Each experiment was run multiple times with different random initializations, and the values in Table 4 are reported as the mean and standard deviation.

3.3.1 Evaluation Phase 1 (Classifier Training)

At this stage, the model only updates the classifier layer so that feature extraction capabilities still depend on the initial weight of the pretraining results. The evaluation outcomes are displayed in Table 5.

Table 5. Classification report Phase 1

Class	Precision	Recall	F1-Score	Support
AKIEC	0.52	0.29	0.37	49
BCC	0.62	0.42	0.59	77
BKL	0.46	0.68	0.55	165
DF	1.00	0.06	0.11	17
MEL	0.42	0.49	0.45	167
NV	0.91	0.86	0.88	1006
VASC	0.71	0.77	0.74	22
Accuracy			0.75	1503
Macro avg	0.66	0.51	0.51	1503
Weighted avg	0.77	0.75	0.75	1503

The results of the evaluation in Phase 1 showed that the model achieved an accuracy of 75%, with a weighted F1-score of 0.75 and a macro F1-score of 0.51. The difference between these metrics suggests that performance across classes remains uneven. Classes with larger sample sizes, such as NV, obtained higher F1-scores, while several minority classes showed lower values. This reflects the early training stage, where only the classifier is updated, so the capacity to distinguish classes in a balanced way remains limited. The distribution of predictions by class is presented in the matrix of confusion in Figure 5.

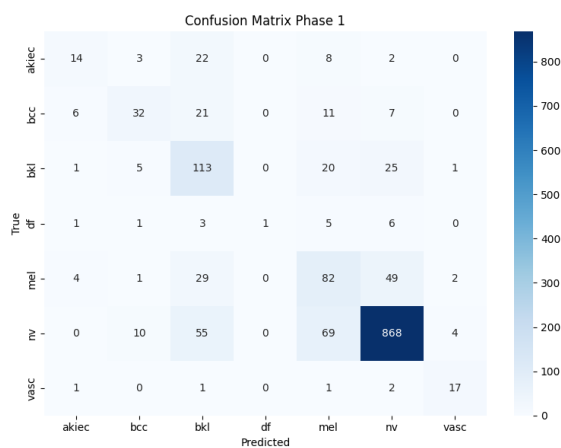


Figure 5. Confusion Matrix Phase 1

Based on the Phase 1 confusion matrix, most of the correct predictions are seen in classes with larger data volumes, such as NV, while some misclassification still occurs in other classes. Some samples in the minority class are also still confused with classes that have similar visual characteristics. This pattern is in line with the results of the classification report, which shows that the model's performance at this stage is not even across all classes.

3.3.2 Evaluasi Phase 2 (Fine Tuning)

Phase 2 involved fine-tuning the final three backbone blocks to better adapt feature representations to skin lesion images. The evaluation outcomes are displayed in Table 6.

Table 6. Classification report Phase 2

Class	Precision	Recall	F1-Score	Support
AKIEC	0.88	0.76	0.81	49
BCC	0.76	0.92	0.84	77
BKL	0.81	0.78	0.79	165
DF	1.00	0.65	0.79	17
MEL	0.64	0.80	0.71	167
NV	0.95	0.92	0.93	1006
VASC	0.91	0.91	0.91	22
Accuracy			0.88	1503
Macro avg	0.85	0.82	0.83	1503
Weighted avg	0.89	0.88	0.88	1503

Based on Table 6 in Phase 2, the model shows that the model achieved an accuracy of 88%, with a macro F1-score of 0.83 and a weighted F1-score of 0.88. Compared to Phase 1, F1-scores in some classes increased. These results show that the fine-tuning process provides changes to model performance, particularly in improving prediction consistency in some classes. Figure 6. presents a confusion matrix that illustrates the level of correct-false predictions in each class.

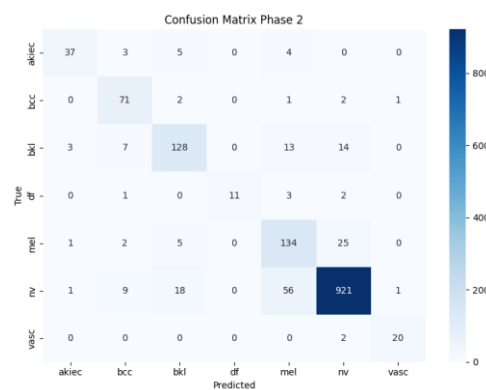


Figure 6. Confusion Matrix fine tuning

As shown in Figure 6, the matrix of confusion reflects the improved performance of the model following fine-tuning in Phase 2. Most samples were correctly classified, particularly in the NV class, while other classes also showed improved correct predictions. Although some misclassifications remain, the prediction pattern appears more concentrated on the correct classes compared to Phase 1.

3.3.3 Model Performance Evaluation Before and After Fine-Tuning

Comparison of F1-scores in both phases of training is shown in Figure 7. Describe changes in model performance more clearly.

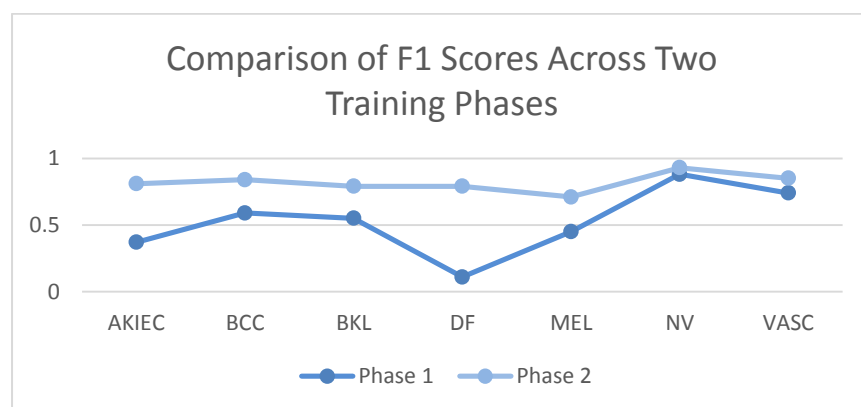


Figure 7. F1-Score Comparison

The F1-score comparison of the two training phases shown in Figure 7. shows changes in model performance before and after the fine-tuning process. In general, F1-score in Phase 2 appears to be higher in some classes than in Phase 1, although the improvement varies between classes.

4. CONCLUSION

This study aimed to develop a multiclass skin cancer classification model that is able to address class distribution imbalances through the application of EfficientNetV2-S and Focal Loss architectures. As expected in the introductory section, the two-stage training shows changes in model performance after the fine-tuning process. In Phase 1, the model reached an accuracy of 75% with a relatively low macro F1-score, while after fine-tuning in Phase 2 the accuracy increased to around 0.88 with an increase in F1-score values in some classes. These findings indicate that the proposed approach can enhance model performance for skin lesion classification under imbalanced data conditions. Future work may explore more advanced architectures, additional regularization strategies, and integrate interpretability approaches such as Grad-CAM to support further studies and real-world applications.

ACKNOWLEDGEMENTS

The authors sincerely thank the Faculty of Applied Sciences and Technology, Ahmad Dahlan University, for their support and the opportunity provided during this research. In addition, gratitude is also expressed to the Mathematics Laboratory for the facilities, resources, and conducive environment that supported the smooth completion of this research.

REFERENCES

- [1] A. Aldwean and D. Tenney, "Artificial Intelligence in Healthcare Sector: A Literature Review of the Adoption Challenges," pp. 129–147, 2024, doi: 10.4236/ojbm.2024.121009.
- [2] L. Choudhary and J. S. Choudhary, "Deep Learning Meets Machine Learning: A Synergistic Approach towards Artificial Intelligence," *J. Sci. Res. Reports*, vol. 30, no. 11, pp. 865–875, 2024, doi: 10.9734/jsrr/2024/v30i112614.
- [3] J. A. Alhijaj and R. S. Khudeyer, "Techniques and Applications for Deep Learning: A Review," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 15, no. 2, pp. 1–12, 2023, doi: 10.29304/jqcm.2023.15.2.1236.
- [4] K. Aggarwal *et al.*, "Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning," *Iraqi J. Comput. Sci. Math.*, vol. 3, no. 1, pp. 115–123, 2022, doi: 10.52866/ijcsm.2022.01.01.013.
- [5] F. Badri, M. Taqijuddin Alawiy, and E. M. Yuniarno, "Deep Learning Architecture Based on Convolutional Neural Network (Cnn) on Animal Image Classification," *J. Ilm. KURSUSOR*, vol. 12, no. 2, pp. 83–92, 2023.
- [6] B. Sistaninejhad, H. Rasi, and P. Nayeri, "A Review Paper about Deep Learning for Medical Image Analysis," *Comput. Math. Methods Med.*, vol. 2023, 2023, doi: 10.1155/2023/7091301.
- [7] M. M and S. P, "COVID-19 infection prediction from CT scan images of lungs using Iterative Convolution Neural Network model," *Adv. Eng. Softw.*, vol. 173, no. June, 2022, doi: 10.1016/j.advengsoft.2022.103214.
- [8] S. Surono, M. Yahya Firza Afitian, A. Setyawan, D. K. E. Arofah, and A. Thobirin, "Comparison of CNN Classification Model using Machine Learning with Bayesian Optimizer," *HighTech Innov. J.*, vol. 4, no. 3, pp. 531–542, 2023, doi: 10.28991/HIJ-2023-04-03-05.
- [9] H. Naseri and A. A. Safaei, "Diagnosis and prognosis of melanoma from dermoscopy images using machine learning and deep learning: a systematic literature review," *BMC Cancer*, vol. 25, no. 1, 2025, doi: 10.1186/s12885-024-13423-y.
- [10] S. G. Yogita K. Desai, Tushar Jadhav, Suvarna Nirmal, Ishita Dutt, Suchita A. Chavan, Avinash Thakur, "View of Diagnosis of Medical Images Using Convolutional Neural Networks.pdf," 2024.

- [11] B. Ji, "Optimizing Medical Imaging with AI: Advanced Cnn and Gan Techniques for Enhanced Diagnostic Efficiency," *ITM Web Conf.*, vol. 78, p. 02005, 2025, doi: 10.1051/itmconf/20257802005.
- [12] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," *Inf.*, vol. 16, no. 3, pp. 1–28, 2025, doi: 10.3390/info16030195.
- [13] I. Araf, A. Idri, and I. Chairi, *Cost - sensitive learning for imbalanced medical data : a review*, vol. 57, no. 4. Springer Netherlands, 2024. doi: 10.1007/s10462-023-10652-8.
- [14] E. Gocer, *Medical image data augmentation: techniques, comparisons and interpretations*, vol. 56, no. 11. Springer Netherlands, 2023. doi: 10.1007/s10462-023-10453-z.
- [15] A. Larasati, S. Surono, A. Thobirin, and D. A. Dewi, "Performance Analysis of Resampling Techniques for Overcoming Data Imbalance in Multiclass Classification," *JUITA J. Inform.*, vol. 13, no. 1, pp. 57–66, 2025, doi: 10.30595/juita.v13i1.25270.
- [16] H. E. Kim, A. C. Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification : a literature review," *BMC Med. Imaging*, pp. 1–13, 2022, doi: 10.1186/s12880-022-00793-7.
- [17] M. Liu, Y. Pei, M. Wu, and J. Wang, "Focal Cosine-Enhanced EfficientNetB0: A Novel Approach to Classifying Breast Histopathological Images," *Inf.*, vol. 16, no. 6, 2025, doi: 10.3390/info16060444.
- [18] E. Gayatri and S. L. Aarthy, "Reduction of overfitting on the highly imbalanced ISIC-2019 skin dataset using deep learning frameworks," *J. Xray. Sci. Technol.*, vol. 32, no. 1, pp. 53–68, 2024, doi: 10.3233/XST-230204.
- [19] D. N. T. Le, H. X. Le, L. T. Ngo, and H. T. Ngo, "Transfer learning with class-weighted and focal loss function for automatic skin cancer classification," pp. 1–7.
- [20] P. Tschandl, C. Rosendahl, and H. Kittler, "Data Descriptor : The HAM 10000 dataset , a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Nat. Publ. Gr.*, pp. 1–9, 2018, doi: 10.1038/sdata.2018.161.
- [21] A. A. A. El-aziz, M. A. Mahmood, and S. A. El-ghany, "A Robust EfficientNetV2-S Classifier for Predicting Acute Lymphoblastic Leukemia Based on Cross Validation," 2025.
- [22] M. Latha, P. S. Kumar, R. R. Chandrika, T. R. Mahesh, V. V. Kumar, and S. Guluwadi, "Revolutionizing breast ultrasound diagnostics with EfficientNet - B7 and Explainable," *BMC Med. Imaging*, 2024, doi: 10.1186/s12880-024-01404-3.
- [23] A. Razaq, S. Ramzan, S. Jabbar, M. M. Iqbal, M. A. Habib, and U. Raza, "A Deep Learning Framework for Early Parkinson's Disease Detection: Leveraging Spiral and Wave Handwriting Tasks with EfficientNetV2-S," *Diagnostics*, vol. 15, no. 21, pp. 1–25, 2025, doi: 10.3390/diagnostics15212795.
- [24] M. Tan and Q. V Le, "EfficientNetV2 : Smaller Models and Faster Training," 2021.
- [25] H. M. Hang Ma, Y.-H. Z. Hang Ma, B.-S. L. Yu-Hang Zhang, and W.-B. C. Bo-Si Liu, "DETRs with Dynamic Contrastive Denoising Training for Smartphone Assembly Parts," *電腦學刊*, vol. 35, no. 3, pp. 175–192, 2024, doi: 10.53106/199115992024063503013.
- [26] M. Motiejauskas and G. Dzemyda, "CCC Publications EfficientNet Convolutional Neural Network with Gram Matrices Modules for Predicting Sadness Emotion," 2024.
- [27] L. Deng, H. Suo, and D. Li, "Deepfake Video Detection Based on EfficientNet-V2 Network," vol. 2022, 2022.
- [28] S. Tummala, V. S. G. Thadikemalla, S. Kadry, M. Sharaf, and H. T. Rauf, "EfficientNetV2 Based Ensemble Model for Quality Estimation of Diabetic Retinopathy Images from DeepDRiD," *Diagnostics*, vol. 13, no. 4, 2023, doi: 10.3390/diagnostics13040622.
- [29] Z. Zhao, E. B. A. Bakar, N. B. A. Razak, and M. N. Akhtar, "Corrosion image classification method based on EfficientNetV2," *Heliyon*, vol. 10, no. 17, p. e36754, 2024, doi: 10.1016/j.heliyon.2024.e36754.
- [30] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, 2018.
- [31] M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, "Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.pdf," *arXiv*, pp. 4510–

- 4520, 2018.
- [32] C. Li, K. Liu, and S. Liu, “A Survey of Loss Functions in Deep Learning,” *Mathematics*, vol. 13, no. 15, p. 2417, 2025, doi: 10.3390/math13152417.
 - [33] J. T. D. C. J. Romero-, “A comprehensive survey of loss functions and metrics in deep learning,” 2025.
 - [34] M. Reyad and A. M. Sarhan, “A modified Adam algorithm for deep neural network optimization,” *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.
 - [35] R. V. Kashyap, “111 A survey of deep learning optimizers - first and second order methods,” 2023.
 - [36] Y. Wang, Z. Xiao, and G. Cao, “A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis,” pp. 666–678, 2022, doi: 10.21595/jve.2022.22271.
 - [37] G. Zeng, “Invariance Properties and Evaluation Metrics Derived from the Confusion Matrix in Multiclass Classification,” no. 1997, 2025.
 - [38] C. A. Sugianto, D. Rohmayani, J. Fredricka, and M. Doheir, “Interpretable Deep Learning Model for Grape Leaf Disease Classification Based on EfficientNet with Grad- CAM Visualization,” pp. 143–152, 2025.
 - [39] N. W. Utami, M. Adi, and P. Putra, “A CNN-Based Information System for Balinese Dance Classification with Hyperparameter Optimization,” vol. 6, no. 2, pp. 376–390, 2025, doi: 10.59395/ijadis.v6i2.1402.