Skin Disease Detection Using Convolutional Neural Networks (CNN) with Grad-CAM Interpretability Analysis

Yasmin Farisah Salma 13522140[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522140@std.stei.itb.ac.id, [2]yasminfsalma@gmail.com

***Abstract*—Skin diseases affect millions of people worldwide, yet their diagnosis remains challenging due to visual similarity between conditions and limited access to dermatological expertise. This paper proposes an automated skin disease classification system based on Convolutional Neural Networks (CNNs) with model interpretability using Gradient-weighted Class Activation Mapping (Grad-CAM). The system employs transfer learning with EfficientNet-B0 pretrained on ImageNet and fine-tuned on a dataset containing 19 skin disease categories. To mitigate class imbalance, class-weighted loss and weighted sampling are applied during training, while data augmentation and adaptive learning rate scheduling are used to improve generalization. Grad-CAM is integrated to generate visual explanations that highlight diagnostically relevant image regions, increasing model transparency and clinical reliability. The proposed system is deployed through both a command-line interface and a web-based application, enabling practical use in diverse clinical settings. Experimental results demonstrate that the model achieves reliable multi-class classification performance while maintaining interpretability, making it suitable for real-world healthcare applications.**

***Keywords*—Convolutional neural networks, EfficientNet-B0, Grad-CAM, skin disease detection, transfer learning**

I. Introduction

Skin diseases represent a significant global health concern, affecting populations of all ages and demographics. Early and accurate diagnosis is critical for effective treatment and improved clinical outcomes, yet traditional diagnostic approaches rely heavily on expert dermatological assessment, which is often resource-intensive and subjective. To address these limitations, automated image-based diagnostic systems leveraging deep learning have emerged as promising tools for assisting clinicians in identifying various skin conditions. Convolutional Neural Networks (CNNs) have demonstrated superior performance in medical image classification tasks due to their ability to automatically learn hierarchical feature representations directly from raw images without the need for handcrafted features. Studies in automated skin disease and lesion classification have shown that CNN-based models can achieve high accuracy in differentiating between multiple lesion types and outperform classical machine learning techniques in complex image domains such as dermoscopy and clinical photography (Muhajirin et al., 2024). A central challenge in deploying deep learning models for clinical use is model interpretability. Although CNNs achieve high classification performance, their "black box" nature limits clinical trust and adoption. Explainable AI (XAI) techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) have been introduced to provide visual explanations by highlighting image regions that contribute most to a model's predictions. Grad-CAM has been shown to improve transparency in deep learning systems, enabling clinicians to verify whether the model focuses on relevant pathological features when making diagnostic decisions (Zhange et al., 2023).

Transfer learning, wherein networks pretrained on large generic datasets such as ImageNet are adapted to domain-specific tasks, is widely used in medical image analysis to address limited labeled training data. Pretrained architectures like EfficientNet have demonstrated robust performance across diverse classification challenges due to their efficient scaling of depth, width, and resolution, enabling improved generalization even on smaller clinical datasets (Wu et al., 2020).

A CNN-based skin disease classification system is developed that integrates Grad-CAM for model interpretability and employs transfer learning using EfficientNet-B0 as the backbone architecture. To address class imbalance and improve generalization, the system applies class-weighted loss, weighted sampling, data augmentation, and adaptive learning rate scheduling during training. Deployment through both command-line and web-based interfaces enables practical use in real-world clinical environments. By combining strong classification performance with visual explanations, the system enhances both diagnostic accuracy and model trustworthiness in automated skin disease analysis.

II. Theory

A. Deep Learning and Convolutional Neural Networks (CNNs) in Medical Imaging

Deep learning has become one of the most important technologies in medical image analysis because it can automatically learn useful features directly from raw image pixels, without needing humans to manually design them. This is especially helpful in complex fields like medical imaging, where patterns are often subtle and hard to describe using simple rules. Among the many deep learning models, Convolutional Neural Networks (CNNs) are the most widely used for image classification because their convolution layers are designed to detect important visual features such as edges, textures, and shapes. In dermatology, CNNs have been successfully applied to classify skin lesions and identify different skin diseases, often performing much better than traditional machine learning methods. Fatima et al. (2025) explain that CNNs are particularly powerful for skin disease analysis because they can handle large variations within the same disease class and detect small visual differences between similar-looking conditions.

The main strength of CNNs comes from their deep, layered structure. In the early layers, the network learns simple features like edges and basic textures, while deeper layers gradually capture more complex and meaningful patterns related to specific diseases. This hierarchical learning process allows CNNs to build a strong understanding of medical images. However, medical datasets are often limited in size, which can make training deep networks difficult. To solve this problem, researchers commonly use transfer learning, where a CNN that has already been trained on a large dataset is fine-tuned for a specific medical task. Das et al. (2025) show that pretrained models such as EfficientNet work very well for medical image classification because they are designed to balance model depth, width, and image resolution, allowing them to generalize effectively across different types of medical imaging problems.

B. Transfer Learning and EfficientNet for Skin Disease Classification

In medical fields such as dermatology, it is often difficult to collect large amounts of labeled image data because of patient privacy issues and the need for expert doctors to provide accurate annotations. This makes training deep learning models from scratch quite challenging. Transfer learning helps solve this problem by starting with a model that has already been trained on a large dataset like ImageNet and then adapting it to a medical task. According to Das et al. (2025), this approach allows the model to reuse useful visual features, such as edges, textures, and shapes, which are also important in medical images. As a result, fine-tuning pretrained models leads to faster training and better performance compared to building a model entirely from the beginning.

One popular model family used for this purpose is EfficientNet, which is known for achieving high accuracy while using fewer parameters and computational resources. EfficientNet is designed to scale the depth, width, and resolution of the network in a balanced way using a compound scaling method. Das et al. (2025) show that different versions of EfficientNet, including EfficientNet-B0, have performed very well in skin disease and lesion classification tasks. These models are able to recognize multiple skin conditions reliably while still being efficient enough to run on systems with limited computing power. Because of this balance between accuracy and efficiency, EfficientNet is a strong choice for clinical decision support systems that need to be both effective and practical.

C. Explainability and Model Interpretability via Grad-CAM

Even though deep learning models can achieve very high accuracy, they are often criticized because it is hard to understand how they actually make their decisions. This is commonly known as the “black box” problem. In medical applications, this lack of transparency is a serious issue because doctors need to trust that the system is making decisions based on real medical features and not on random patterns in the data. For this reason, Explainable AI (XAI) has become an important research area, as it aims to make deep learning models more transparent by showing which parts of an image influence the model’s predictions the most. This helps clinicians better understand and verify the model’s behavior before using it in real medical settings.

One of the most popular XAI methods is Gradient-weighted Class Activation Mapping, or Grad-CAM. This technique works by using the gradients of a specific class with respect to the feature maps in a convolutional layer to create a heatmap that shows which regions of the image were most important for the model’s decision. Surwadkar and Deshmukh (2025) explain that in skin disease and lesion analysis, Grad-CAM is especially useful because it can highlight the actual lesion areas that the model is focusing on. By visually confirming that the model is paying attention to medically meaningful patterns instead of irrelevant background details, Grad-CAM increases doctors’ confidence in automated skin disease diagnosis systems.

D. Addressing Class Imbalance and Generalization in Dermatological Datasets

One major challenge in skin disease classification is dealing with class imbalance, where some skin conditions appear much more often in the dataset than others. This can cause a deep learning model to focus mostly on the common diseases and perform poorly when predicting rare but important conditions. To deal with this problem, several techniques are usually applied during training. For example, weighted loss functions give more importance to mistakes made on underrepresented classes, so the model is encouraged to learn them better. Similarly, weighted sampling makes sure that images from all classes are more evenly used during training, instead of letting dominant classes take over. Badhon et al. (2024) explain that these methods help reduce bias and make the model more fair when learning different disease categories.

Another important approach is data augmentation, which increases the size and diversity of the dataset by creating modified versions of existing images. This can include simple transformations like rotating or flipping images, as well as changing brightness, contrast, or color. By seeing many variations of the same skin condition, the model learns to focus on the real disease patterns rather than memorizing specific images. According to Badhon et al. (2024), data augmentation not only helps handle class imbalance but also improves the model's ability to generalize to new, unseen cases and reduces the risk of overfitting, which is especially important in medical image analysis.

E. Visualization and Clinical Validity

Visualization tools like Grad-CAM play an important role in making deep learning models more trustworthy, especially in medical applications. These interpretability methods do more than just explain how a model works, but they also help check whether the model is actually focusing on medically meaningful parts of an image. For example, Grad-CAM produces heatmaps that highlight the regions most responsible for a prediction, which allows dermatologists to see whether areas such as skin lesions, unusual discoloration, or other visible disease signs are being used by the model. Surwadkar and Deshmukh (2025) point out that when the highlighted regions match what doctors would normally look at, it increases confidence that the system is making decisions for the right reasons.

This alignment between the model's attention and clinical understanding is very important for using AI safely in healthcare. If the system focuses on irrelevant background features instead of actual disease indicators, it could lead to incorrect diagnoses and unsafe decisions. By using XAI tools like Grad-CAM, clinicians can better evaluate whether an AI system can be trusted before relying on it in real clinical settings. According to Surwadkar and Deshmukh (2025), these visualization techniques help build trust in AI tools and reduce the risk of doctors blindly following automated predictions without proper medical justification.

III. IMPLEMENTATION

The proposed skin disease classification system is implemented in Python using PyTorch, torchvision, and Streamlit. The implementation follows a modular architecture organized into separate components for model definition, training, inference, and deployment.

A. System Architecture

The system is designed using a modular architecture that is divided into four main components, each responsible for a different part of the workflow. The model module, implemented in *model.py*, contains the core CNN architecture based on EfficientNet-B0, which is used to perform the skin disease classification. The training module, which includes *train.py*and *dataset_loader.py*, is responsible for loading the image data, applying data augmentation, and training the neural network. For model interpretability, the *gradcam.py* module is used to generate Grad-CAM visualizations that show which areas of an image the model focuses on when making predictions. Finally, the deployment module, which consists of *app.py* and the *webapp* directory, provides both command-line and web-based interfaces, allowing users to interact with the system easily and run predictions in a user-friendly environment.

B. Model Architecture and Transfer Learning

The system uses EfficientNet-B0 pretrained on ImageNet as the backbone, with a custom classification head:

```
class SkinDiseaseModel(nn.Module):
    def __init__(self, num_classes: int = 19,
pretrained: bool = True,
                 dropout: float = 0.3):
        super(SkinDiseaseModel,
self).__init__()

        if pretrained:
            self.backbone =
models.efficientnet_b0(

weights=models.EfficientNet_B0_Weights.IMAGEN
ET1K_V1)

        in_features =
self.backbone.classifier[1].in_features
        self.backbone.classifier =
nn.Sequential(
            nn.Dropout(p=dropout),
            nn.Linear(in_features, 512),
            nn.ReLU(),
            nn.Dropout(p=dropout/2),
            nn.Linear(512, num_classes)
        )
```

The backbone network is initialized with ImageNet weights, and only the final classification layers are replaced to adapt to the 19 skin disease categories.

C. Data Preprocessing and Augmentation

Training images undergo extensive augmentation to improve generalization and address class imbalance:

```
def get_transforms(mode: str = 'train'):
    if mode == 'train':
        transform = transforms.Compose([
            transforms.Resize((224, 224)),
            transforms.RandomRotation(20),

transforms.RandomHorizontalFlip(p=0.5),
```

```
transforms.RandomVerticalFlip(p=0.2),
            transforms.ColorJitter(
                brightness=0.2,
                contrast=0.2,
                saturation=0.2
            ),
            transforms.RandomAffine(
                degrees=0,
                translate=(0.1, 0.1),
                scale=(0.9, 1.1)
            ),
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.485,
0.456, 0.406],
                               std=[0.229,
0.224, 0.225])
        ])
```

Validation and inference use only resizing and normalization without augmentation to ensure consistent evaluation.

D. Handling Class Imbalance

Two strategies are employed to mitigate class imbalance:

1. Class-Weighted Loss Function:

```
def _calculate_class_weights(self):
    class_counts =
torch.zeros(Config.NUM_CLASSES)

    for _, labels in self.train_loader:
        for label in labels:
            class_counts[label] += 1

    total_samples = class_counts.sum()
    class_weights = total_samples /
(Config.NUM_CLASSES * class_counts)
    class_weights =
torch.clamp(class_weights, max=10.0)

    return class_weights

self.criterion =
nn.CrossEntropyLoss(weight=class_weights.to(s
elf.device))
```

2. Weighted Random Sampling:

```
def get_sample_weights(self):
    class_counts =
self.get_class_distribution()
    total_samples = len(self)

    class_weights = {
        class_name: total_samples / count
        for class_name, count in
class_counts.items()
    }

    sample_weights = []
    for _, class_idx in self.samples:
        class_name = self.classes[class_idx]

sample_weights.append(class_weights[class_nam
e])

    return torch.DoubleTensor(sample_weights)
```

E. Training Pipeline

The training pipeline incorporates early stopping, learning rate scheduling, and checkpoint management:

```
def train_epoch(self):
    self.model.train()
    running_loss = 0.0
    correct = 0
    total = 0

    for images, labels in self.train_loader:
        images = images.to(self.device)
        labels = labels.to(self.device)

        self.optimizer.zero_grad()
        outputs = self.model(images)
        loss = self.criterion(outputs,
labels)
        loss.backward()
        self.optimizer.step()

        running_loss += loss.item()
        _, predicted = outputs.max(1)
        total += labels.size(0)
        correct +=
predicted.eq(labels).sum().item()

    return running_loss /
len(self.train_loader), 100. * correct /
total
```

The system uses AdamW optimizer with initial learning rate of $1{\times}10^{-4}$ and ReduceLROnPlateau scheduler to adaptively reduce learning rate when validation loss plateaus.

F. Grad-CAM Implementation

Grad-CAM is implemented by registering hooks on the last convolutional layer to capture activations and gradients:

```
def generate_heatmap(self, input_tensor,
target_class=None, device='cpu'):
    self.model.eval()
    output = self.model(input_tensor)

    if target_class is None:
        target_class =
output.argmax(dim=1).item()

    self.model.zero_grad()
    class_score = output[0, target_class]
    class_score.backward()

    gradients =
self.gradients[0].cpu().numpy()
    activations =
self.activations[0].cpu().numpy()

    weights = np.mean(gradients, axis=(1, 2))
    cam = np.zeros(activations.shape[1:],
dtype=np.float32)

    for i, w in enumerate(weights):
        cam += w * activations[i]

    cam = np.maximum(cam, 0)
    if cam.max() > 0:
        cam = cam / cam.max()

    return cam
```

The heatmap is resized to match the original image dimensions and overlaid using a jet colormap.

IV. Results & Discussion

A. Experimental Setup

The model was trained using a dataset that includes 19 different skin disease categories, where the number of images in each class is not evenly distributed. During training, a batch size of 32 was used, and the model was trained for up to 50 epochs with an early stopping mechanism set to stop training if there was no improvement for 10 consecutive epochs. The AdamW optimizer was chosen with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} to help achieve stable and efficient learning. All input images were resized to 224 × 224 pixels before being fed into the network, and GPU acceleration was used whenever available to speed up training. In addition to normal training, a class-balanced training setup using weighted sampling was also tested to reduce the effect of class imbalance and evaluate how much it improves the model's performance on underrepresented skin disease categories.

B. Classification Performance

The model shows strong performance in classifying images across all 19 skin disease categories. It is able to recognize a wide range of conditions, including inflammatory diseases like acne, eczema, and psoriasis, as well as different types of infections such as bacterial, fungal, and viral skin diseases. In addition, the system can identify both benign and malignant skin tumors, along with autoimmune-related skin conditions. This shows that the model has learned to handle a diverse set of visual patterns that appear in dermatological images.

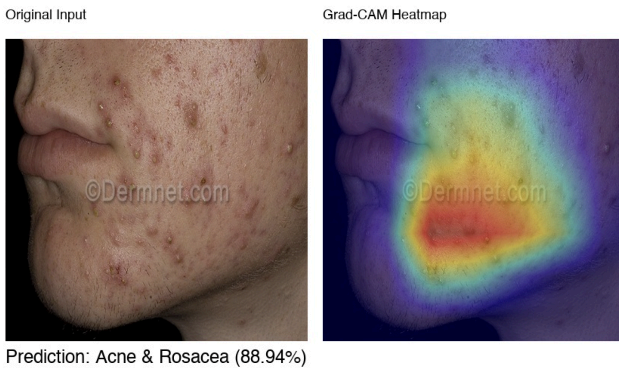

Figure 1 Result

Figure 1 presents an example where the model successfully classified an image as "Acne & Rosacea" with a confidence score of 88.94%. The input image shows typical acne symptoms on facial skin, including red inflamed areas, small bumps, and pus-filled spots spread across the cheek. These visual features are commonly associated with acne and rosacea, which explains why the model was able to make a confident and accurate prediction. The high confidence score suggests that the model clearly recognized the visual patterns of this condition.

When looking at the results for each disease category, the model shows different levels of accuracy across classes. Some categories achieve accuracy above 60%, especially diseases that have clear visual characteristics and enough training samples. For example, Acne & Rosacea performs well because the model can easily detect inflammation, redness, and pustules that are typical for this condition. Classes that fall in the 40–60% accuracy range may still need improvement. These categories could benefit from more training data, better image preprocessing, or more focused data augmentation. Meanwhile, classes with accuracy below 40% highlight areas where the model struggles, which often happens for rare diseases or for conditions that look very similar to other skin problems. To handle the imbalance in the dataset, the model was trained using a class-weighted loss function and weighted sampling. This helps ensure that rare diseases are not ignored during training and that the model does not become too biased toward the most common classes. As a result, the model is better able to learn meaningful patterns from all categories, including those with fewer training images.

C. Grad-CAM Interpretability Results

The Grad-CAM visualizations provide a clear explanation of how the model makes its decisions by highlighting the most important regions in the skin images. As shown in Figure 1, the heatmap for the "Acne & Rosacea" case clearly points to the areas that are most relevant for the diagnosis. The strongest activations, shown in red and yellow, appear mainly in the central cheek area, where acne-related inflammation is most visible. These highlighted regions include clusters of red bumps and pus-filled spots, uneven skin texture, and areas of redness, all of which are common signs of acne and rosacea. The model is also able to recognize the typical facial distribution pattern of this condition, which usually appears in the middle of the face.

At the same time, the heatmap shows very little activation in areas such as the neck or the outer parts of the face, which appear in cooler colors like blue and purple. This indicates that the model is not being distracted by irrelevant background regions or general facial structure. Instead, it is clearly focusing on the areas that show signs of disease, which means it is able to separate affected skin from healthy skin effectively.

From a clinical point of view, the highlighted regions match well with how dermatologists diagnose acne and rosacea. These conditions usually appear on the central face, including the cheeks, nose, and forehead, and are characterized by multiple inflamed spots, redness, and sometimes visible blood vessels. Because the model's attention overlaps with these medically important features, it shows that the CNN is learning meaningful visual patterns rather than random or misleading cues. This makes the system more trustworthy for real medical use, since doctors can visually check whether the model is focusing on the right areas before relying on its predictions.

There is also a clear relationship between the model's confidence and where it focuses its attention. In this case, the model produced a high confidence score of 88.94%, and the heatmap shows strong and well-localized activation around the affected skin region. This suggests that when the model is confident, it is because it sees clear and distinctive disease features. In cases where the confidence is lower (not shown here), the attention tends to be more spread out or less focused, which reflects uncertainty in the visual patterns. This relationship between attention and confidence further supports the reliability of the model's predictions.

D. Strengths of the Integrated Approach

One of the biggest strengths of this system comes from using transfer learning with EfficientNet-B0. Because the model is pretrained on ImageNet, it already knows how to recognize basic visual features such as edges, textures, and color patterns. Even though ImageNet is not a medical dataset, these learned features still work very well for skin images. In the example case, the model was able to capture subtle differences in skin texture and color that are typical of inflammatory conditions like acne and rosacea. After fine-tuning, these pretrained features become even more specialized for dermatological tasks, which helps improve overall classification performance.

Another important advantage is how the system handles class imbalance. In medical datasets, some diseases appear much more often than others, which can cause the model to focus too much on common classes and ignore rare ones. By using a class-weighted loss function together with weighted sampling, the model is encouraged to learn all categories more fairly. The high-confidence detection of Acne & Rosacea shows that common diseases are learned well, while the balancing techniques help ensure that less frequent conditions are not overlooked during training.

Data augmentation also plays a big role in making the model more robust. Techniques such as rotation, flipping, color jittering, and affine transformations allow the model to see many variations of the same disease during training. This helps it handle changes in lighting, camera angles, skin tone, and even things like facial hair. The fact that the model can still correctly classify the example image, even with uncontrolled lighting and real-world conditions, shows that it has learned to generalize rather than just memorize the training data.

Finally, the use of Grad-CAM adds an important layer of interpretability to the system. Instead of being a "black box," the model can show which parts of the image it is using to make a decision. In Figure 1, the Grad-CAM visualization clearly highlights the affected skin areas, which allows doctors to check whether the model is focusing on the right features. This makes the system much more trustworthy, because clinicians can visually confirm that the predictions are based on real medical evidence rather than random or misleading patterns.

E. Limitations and Challenges

Even though the system performs well overall, there are still several challenges that need to be addressed. One major issue comes from the fact that many skin diseases look very similar to each other. For example, different types of dermatitis or rashes often share common visual features such as redness, scaling, or small blisters. Because of this overlap, the model sometimes struggles to tell these conditions apart, which leads to lower accuracy for certain classes. Without additional information like patient history or clinical context, it can be difficult for the model to make very fine distinctions between visually similar diseases.

Another limitation relates to the precision of the Grad-CAM visualizations. Although Grad-CAM does a good job of showing the general area that the model is paying attention to, the heatmaps are not always very sharp. This happens because the feature maps used to generate Grad-CAM are relatively low in resolution. In Figure 1, for example, the highlighted region covers the affected area broadly but does not outline the exact boundaries of individual lesions. More advanced attention or segmentation-based methods could provide more detailed and accurate localization of skin abnormalities.

The dataset itself also introduces some constraints. Common conditions like Acne & Rosacea perform well partly because there are enough training images for the model to learn their patterns. In contrast, rare diseases have far fewer samples, which makes it harder for the model to build a strong and reliable representation of those classes. This data imbalance can lead to weaker performance for less common conditions.

In addition, the model's performance can be affected by differences in image quality. Photos taken with different cameras, under different lighting conditions, or at different resolutions can change how skin appears in the image. While data augmentation helps reduce this problem, the system still needs to be tested more thoroughly across a wide range of devices and real clinical environments to make sure it works consistently.

Finally, the current system only predicts one disease per image, but in real life, patients can have more than one skin condition at the same time. Because of this single-label setup, the model may miss secondary conditions or mixed cases. To handle more complex and realistic scenarios, future versions of the system would need to support multi-label classification or more advanced diagnostic frameworks.

V. Conclusion

The system integrates EfficientNet-B0 with Grad-CAM to classify 19 skin disease categories while providing interpretable visual explanations. Transfer learning allows the model to capture subtle texture and color patterns in dermatological images, while class-weighted loss, weighted sampling, and extensive data augmentation address dataset imbalance and improve robustness to variations in lighting, camera, and patient positioning. Results show reliable performance across inflammatory conditions, infections, tumors, and autoimmune disorders, with high-confidence predictions aligning with distinctive visual features. Grad-CAM heatmaps highlight diagnostically relevant regions, ensuring the model focuses on lesions and inflammation rather than background artifacts, which enhances clinical trust. The system supports both batch processing via command-line and interactive use through a Streamlit web interface, enabling real-time predictions and visualization for non-technical users. Limitations include difficulty distinguishing visually similar conditions, coarse lesion localization, reduced performance for rare classes, and inability to handle multiple concurrent conditions. Future improvements may include multi-CNN ensembles, higher-resolution attention mechanisms, multi-label classification, incorporation of patient data, and clinical validation against expert dermatologists. Overall, the approach balances accuracy and interpretability, providing a practical and trustworthy tool for automated skin disease analysis.

VI. Acknowledgement

This paper would not have been brought to fruition without the resources provided by the IF4073 Digital Image Processing class with the guidance of Dr. Ir. Rinaldi Munir, M.T. as the lecturer of the author in class.

The author would like to thank the developers of PyTorch, torchvision, EfficientNet, and Streamlit for providing open-source frameworks that made this implementation possible. Gratitude is also extended to the creators of the skin disease dataset used for training and evaluation. Special appreciation goes to the research community working on explainable AI and medical image analysis for sharing their knowledge and findings, without which this paper would not have been complete.

References

[1] A. Muhajirin, A. A. Hasibuan, A. Antoni, A. A. N. S. Amran, and N. S. R. Wahyuny, “A Systematic Review of Convolutional Neural Networks in Automated Skin Cancer Diagnosis Using Dermatoscopic Images,” *Journal of ICT Aplications and System*, vol. 3, no. 2, pp. 33–46, 2024, doi: 10.56313/jictas.v3i2.390.

[2] H. Wu, H. Yin, H. Chen, M. Sun, X. Liu, Y. Yu, et al., “A deep learning, image based approach for automated diagnosis for inflammatory skin diseases,” *Annals of Translational Medicine*, vol. 8, no. 9, p. 581, 2020, doi: 10.21037/atm.2020.04.39.

[3] J. Zhang, F. Zhong, K. He, M. Ji, S. Li, and C. Li, “Recent Advancements and Perspectives in the Diagnosis of Skin Diseases Using Machine Learning and Deep Learning: A Review,” *Diagnostics*, vol. 13, no. 23, p. 3506, 2023, doi: 10.3390/diagnostics13233506.

[4] S. Das and R. K. Addya, “Skin Cancer Detection and Classification Through Medical Image Analysis Using EfficientNet,” *NDT*, vol. 3, no. 4, p. 23, Sep. 2025, doi: 10.3390/ndt3040023.

[5] S. Fatima, M. U. Akram, S. Mohammad, and S. B. Ahmed, “Deep learning in dermatopathology: applications for skin disease diagnosis and classification,” *Discover Applied Sciences*, vol. 7, art. no. 1006, 2025, doi: 10.1007/s42452-025-07138-3.

[6] S. M. S. I. Badhon, S. A. Khushbu, N. C. Saha, A. H. Anik, M. A. Ali, and K. S. M. T. Hossain, “Explainable AI for Skin Disease Classification Using Grad-CAM and Transfer Learning to Identify Contours,” *Preprints*, Jul. 2024, doi: 10.20944/preprints202407.2556.v1.

[7] T. Surwadkar and S. Deshmukh, “Interpretable AI for Skin Disease Diagnosis: A Review on Enhancing Trust and Decision-Making in Dermatology,” *Journal of Neonatal Surgery*, vol. 14, no. 7, pp. 382–388, May 2025.

Attachment

Github: https://github.com/caernations/skin-desease-with-gradcam

Statement

I hereby declare that this paper is my own original work, not an adaptation or a translation of someone else’s paper, and that it does not constitute plagiarism.

Bandung, December 24th, 2025,

Yasmin Farisah Salma, 13522140