Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20503-5

Custom CNN architectures for skin disease classification: binary and multi-class performance

Pragya Gupta[1] · **Jagannath Nirmal**[1] · **Ninad Mehendale**[1]

Received: 23 July 2024 / Revised: 22 November 2024 / Accepted: 9 December 2024

Abstract
This study explores the performance of custom Convolutional Neural Network (CNN) architectures for both binary and multiclass skin disease classification, utilizing datasets sourced from Kaggle and Google. Images of ringworm and healthy skin were used, resized to 224×224 pixels, and augmented with techniques such as flipping and rotation to address data limitations. We experimented with two dataset splits (80:20 and 70:30) and compared our custom CNN's performance against State-of-the-Art (SOTA) models and Vision Transformers (ViTs). For binary classification, our custom CNN architecture included four convolutional layers (32, 64, and 128 filters in successive blocks) with ReLU activation after each convolutional layer, followed by max-pooling layers and dense layers, and a final softmax output. This model achieved 98.9% accuracy, demonstrating strong performance in distinguishing ringworm from healthy skin. For the multiclass (23-class) classification task, we adapted the CNN architecture with added class frequency-based weights and achieved 35.23% accuracy, illustrating the challenges in multi-class dermatological classification. Our results indicate the practical applicability of a custom CNN, modified to incorporate class frequency balancing, for dermatological contexts, while highlighting the potential and limitations compared to SOTA architectures and ViTs in skin disease classification.

Keywords Skin infections · Dermatology · Deep learning · Convolutional neural network · Classification

1 Introduction

Skin diseases are a significant global health concern, affecting millions of people worldwide [1]. Among these conditions, ringworm, a common fungal infection, poses a particular challenge due to its contagious nature and potential for misdiagnosis [2]. Ringworm is caused

✉ Ninad Mehendale
ninad@somaiya.edu

Pragya Gupta
pragya.g@somaiya.edu

Jagannath Nirmal
jhnirmal@somaiya.edu

[1] Department of Electronics Engineering, K.J.Somaiya College of Engineering, Somaiya Vidyavihar University, Vidyavihar, Mumbai 400077, Maharashtra, India

by dermatophytes, leading to ring-shaped, scaly, and itchy lesions [3]. Although not lethal, its highly contagious nature makes it a public health concern, particularly in close-contact settings like schools and gyms. While treatable with antifungal medications, delayed or inaccurate diagnosis can lead to complications, such as secondary infections [4], especially in immunocompromised individuals [5].

The adoption of AI and ML has revolutionized dermatological diagnostics by significantly enhancing accuracy and efficiency in detecting skin conditions [6]. These technologies, especially through the application of deep learning and Convolutional Neural Networks (CNNs) [7], have outperformed traditional diagnostic methods in image analysis, providing dermatologist-level accuracy in identifying various skin diseases and supporting clinical decision-making processes [8],[9]. In recent years, deep learning has revolutionized the field of artificial intelligence, particularly in image analysis tasks. Unlike traditional machine learning techniques, which often require manual feature extraction, deep learning models and CNNs automatically learn and extract complex patterns and features directly from the raw image data [10]. This capability is crucial in dermatological diagnostics, where subtle differences in texture, color, and shape play a significant role in distinguishing between various skin conditions [11]. CNNs are specifically designed to handle visual data, making them ideally suited for the intricate task of skin disease classification [12].

By leveraging their hierarchical feature extraction capabilities, our study aims to demonstrate how CNNs can achieve a higher level of accuracy and robustness compared to conventional machine learning methods, ultimately leading to more reliable and efficient diagnostic tools in dermatology. This study focuses on the application of CNNs in the classification of ringworm and healthy skin images, as well as the broader task of multi-class skin disease classification.

The main contributions of this study are:

1. **Development of a Custom CNN for Skin Disease Classification:** This study presents a custom CNN architecture designed to classify skin conditions in both binary (ringworm vs. healthy skin) and multi-class settings, with the primary objective of creating a robust model that can assist healthcare professionals in early detection and diagnosis. By leveraging deep learning techniques, the model aims to reduce diagnostic errors, improve patient outcomes, and enhance resource efficiency in dermatology.
2. **Analysis of Multi-Class Classification Challenges:** A comprehensive analysis of challenges specific to multi-class skin disease classification is provided, focusing on class imbalance, interclass visual similarities, and intraclass variability. These insights are valuable for addressing limitations in automated dermatological diagnosis.
3. **Benchmarking Against State-of-the-Art Models:** The custom CNN model is compared with state-of-the-art approaches, highlighting its strengths and areas for further improvement. This comparison establishes a baseline for future research in automated skin disease diagnosis using deep learning techniques.

Our study utilizes a dataset comprising ringworm images sourced from Kaggle [13], [14] and healthy skin images obtained from Google [15]. To address the limited size of the dataset, we employ data augmentation techniques to enhance the model's generalization capabilities. The custom CNN architecture is designed to extract relevant features from skin images and learn complex patterns that distinguish between different skin conditions. By comparing the performance of our model in binary and multi-class classification tasks, we aim to gain insights into the scalability and adaptability of the approach for real-world clinical applications. This research contributes to the growing body of work on AI-assisted dermatology and explores the potential of deep learning in revolutionizing skin disease diagnosis.

The paper is organized as follows: Section II offers a literature review on the automated classification of skin diseases. Section III explains the methodology used in this study. Section IV outlines the results obtained. Section V discusses the findings, and Section VI concludes the study.

2 Literature review

Effectively diagnosing skin infections is crucial in dermatology, as accurate and prompt identification leads to improved patient outcomes [16]. Researchers have explored various feature extraction techniques to enhance the accuracy of skin disease classification.

Sreedhar et al. [17] conducted a comparative study of traditional and modern image processing techniques for skin cancer detection, highlighting the importance of feature extraction and image segmentation. Similarly, Wei et al. [18] proposed an automated skin cancer detection model based on ensemble lightweight deep learning networks, where feature extraction modules are essential for lesion classification. Maniraj et al. [19] employed a combination of Genetic Algorithm and Deep Learning Neural Network for dermoscopic image classification, achieving high accuracy across different datasets. These studies emphasize the effectiveness of feature extraction techniques, especially when combined with diverse classification algorithms.

Recent research has increasingly focused on leveraging deep learning algorithms for skin disease classification. Alam et al. [20] highlighted the efficiency of deep learning in skin disease detection. Obayya et al. [21] introduced a deep learning technique utilizing multi-attention fusion for skin cancer diagnosis, demonstrating the growing application of Artificial Intelligence in dermatology. Anggriandi et al. [22] compared CNNs with MobileNet architecture and CNN-SVM methods for classifying human skin diseases, suggesting that integrating CNNs for feature extraction with SVM for classification can be an effective strategy. Deep learning, particularly CNNs, have revolutionized the field of skin disease classification. While pre-trained models offer a starting point, custom CNN architectures can be tailored to achieve superior performance in both binary (healthy vs. disease) and multi-class (multiple diseases) classification tasks.

Custom CNN architectures offer flexibility in addressing the challenges posed by imbalanced datasets. A recent study by Allugunti et al.[23] proposed a Machine Learning Model for Skin Disease Classification using a custom CNN that achieved high accuracy. Another approach by Wei et al. [24] combined DenseNet and ConvNeXt architectures, demonstrating promising results for multi-class classification.

The design choices within these architectures significantly impact performance. Factors like network depth, activation functions, and the use of techniques like data augmentation and dropout layers all influence the model's ability to learn discriminative features from skin lesion images. Several studies explore these design considerations, such as CNN-based approaches by Sazzadul et al. [25].

Biasi et al. [26] conducted a detailed analysis of key CNN architectures for detecting melanoma in clinical images. Emphasizing the need to minimize the False Negative Rate (FNR) in CAD systems, they adapted popular models like VGG16, AlexNet, DenseNet, InceptionV3, and others for melanoma classification using the MED-NODE dataset, which contains 170 clinical images. VGG16 and AlexNet emerged as the top performers in reducing FNR, with 0.07 and 0.13, respectively. AlexNet achieved an accuracy of 89%, sensitivity of 87%, and specificity of 90%, while VGG16 showed lower specificity (59%) but reasonable sensitivity (82%). These results underscore VGG16 and AlexNet's suitability for developing

CAD systems for melanoma diagnosis. Yadav et al. [27] reviewed 45 studies on skin disease detection from 2021 to 2023, highlighting the increasing role of ML and DL in healthcare. They found that 32 studies focused on deep learning, 11 on traditional machine learning, and 2 used hybrid approaches. The review covered key models, datasets, and metrics, noting challenges like handling noisy data and accurately capturing symptoms. This survey provides researchers with valuable insights into current trends and challenges in using ML and DL for skin disease detection. Biasi et al. [28] proposed a novel CNN architecture design using genetic algorithms for melanoma detection. By allowing network configurations to evolve over generations, they optimized the CNN for the ISIC dataset, achieving 94% accuracy, 90% sensitivity, 97% specificity, and 98% precision. This hybrid approach demonstrated the effectiveness of genetic algorithms in automating CNN design without manual intervention, paving the way for more efficient and accurate melanoma classification.

Sadik et al. [29]address challenges in skin disease recognition due to low c ontrast and high similarity between conditions. They use CNN-based architectures, specifically MobileNet and Xception, in a computer vision system for improved disease recognition, achieving 96% and 97% accuracy, respectively. The study benchmarks against CNNs like ResNet50 and InceptionV3, demonstrating the effectiveness of transfer learning and data augmentation. They also propose a web-based framework for real-time diagnostics. Schielein et al. [30] explore CNNs for outlier detection in non-melanoma dermatological conditions, evaluating models like InceptionV3, Xception, and ResNet50 on datasets curated by dermatologists. Testing on 4,051 clinical images, they achieved high accuracies, notably 100% for onychomycosis. The study highlights that expert data selection enhances model accuracy, offering insights for training dermatology AI systems. Hossain et al. use an ensemble approach combining deep learning models (e.g., MobileNetV2, ResNet50) for skin cancer detection. Using the Max Voting Ensemble Technique, they achieved 93.18% accuracy on the ISIC 2018 dataset, improving upon individual model performance. This method supports healthcare professionals in precise skin cancer diagnosis, validated on HAM10000 for robustness. Perez et al. [31] propose a CNN model for melanoma diagnosis optimized by genetic algorithms to select ensemble members. This approach combines model features, improving prediction accuracy and generalization, with an 11-13% performance boost across sixteen datasets. The method leverages transfer learning, data augmentation, and CPU-efficient segmentation, presenting a resource-friendly diagnostic tool.

Abbas et al. [32] introduce Assist-Dermo, a lightweight SVT-based system for classifying nine skin lesion types. Utilizing depthwise separable CNN layers, it achieves 95.6% accuracy with enhanced efficiency. Tested on datasets like HAM10000, the model outperforms other methods and provides a practical tool for dermatologists, addressing class imbalance with data augmentation and image preprocessing.

Yang et al. [33] propose a novel ViT model specifically designed for skin cancer classification. Recognizing the difficulty in distinguishing types of skin cancer due to visual similarities, especially in early stages, they present a four-block approach aimed at enhancing classification accuracy in clinical skin images. The model leverages transfer learning by pretraining on the ImageNet dataset and fine-tuning on the HAM10000 dataset. Experimental results demonstrate that the model achieves a high classification accuracy of 94.1%, outperforming the current state-of-the-art model, Inception-ResNet-V2 with soft attention, on the same dataset. The model also performs better on the Edinburgh DERMOFIT dataset than baseline models, indicating its effectiveness in skin cancer classification.

Evaluating these architectures requires careful consideration of appropriate metrics. Accuracy, precision, recall, F1-score, and AUC-ROC curves are all commonly used to assess the model's effectiveness in identifying skin diseases.

Current research suggests that custom CNN architectures can outperform pre-trained models in certain scenarios. However, there's still room for improvement. Future directions include addressing limited datasets, enhancing model interpretability, and developing architectures optimized for mobile or resource-constrained environments.

3 Methodology

This section describes the methodology for this study.

Figure 1 shows the block diagram that illustrates the sequential steps involved in our deep learning model for skin disease classification. First, the Dataset is sourced and organized into labeled categories. The Data Pre-processing stage includes image resizing, normalization, and augmentation to improve the model's generalizability by introducing a variety of transformations. Finally, the processed images are fed into a CNN Model, which automatically extracts features and learns patterns through a series of convolutional, pooling, and fully connected layers to classify images into respective disease categories. For binary classification, the model distinguishes between ringworm and healthy skin, while in multiclass classification, it categorizes images into one of the 23 identified skin disease classes.

3.1 Dataset

In our study, to ensure reproducibility and consistency of results across different runs, we set a seed value of 123 for all random operations. By fixing this seed value, we aimed to standardize the behavior of operations involving randomness, such as data splitting, data augmentation, and model initialization. This practice helps in maintaining the same experimental conditions, enabling other researchers to reproduce our findings reliably and verify the outcomes using the same settings. For the binary classification task, two sets of images were utilized:

1. images of Ringworm (tinea corporis) were obtained from a dataset available on Kaggle [14], which has been accessed and utilized by a few researchers [34], [35] and
2. images of healthy skin were sourced from Google. These images were compiled and organized into a new dataset hosted on Kaggle [15], ensuring consistency and availability for future experiments.

For the multiclass classification task, a broader dataset was sourced from Dermnet, also available on Kaggle [13]. This dataset, widely used by several researchers for skin disease classification tasks, [7], [36] contains a diverse range of skin conditions. It enabled the development of a robust multiclass classification model. The dataset comprises images of various dermatological conditions, providing a comprehensive basis for classifying multiple skin

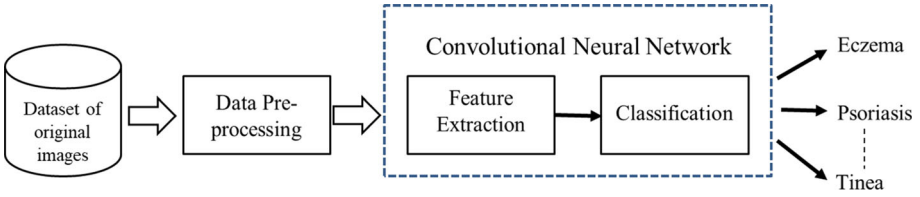

Fig. 1 Block diagram representing the end-to-end workflow: Dataset acquisition, Data Pre-processing for enhanced feature learning, and CNN-based classification model

diseases, including tinea corporis, psoriasis, eczema, and other visually similar conditions. Table 1 presents the distribution of the images in the Dermnet data sourced from Kaggle.

3.2 Dataset expansion

Initially, model training and evaluation were performed using a subset of the Dermnet dataset, consisting of approximately 4,590 images across various skin conditions. This subset allowed for preliminary model validation and provided initial insights into classification performance. However, recognizing the limitations posed by a smaller dataset in terms of representational diversity, we subsequently expanded the experiments to utilize the entire Dermnet dataset, encompassing over 24,829 images. This expansion provided a broader range of examples per class, enhancing the model's exposure to intra-class variability and enabling a more robust training process. Consequently, the model was better equipped to generalize across the diverse cases within the full dataset, leading to improved accuracy and performance stability across different skin conditions. For reproducibility, the experimentation details are provided in the Colab Notebook, and the full dataset analysis can be accessed in the GitHub Repository.

3.3 Data pre-processing

To prepare the dataset for training, we applied a series of pre-processing steps as follows:

1. Image Resizing: All input images were resized to a uniform dimension of 224 × 224 pixels to ensure consistency across the dataset. This resizing was necessary for several reasons:
   - to align with standard input dimensions used by state-of-the-art pre-trained models such as VGG19, ResNet, and Inception, which are designed to work with images of this size;
   - to maintain consistency and reduce variability during training and evaluation; and
   - to optimize computational efficiency, as using a fixed image size of 224 × 224 strikes a balance between preserving sufficient image detail and minimizing memory usage during training. This transformation helps the models extract relevant features while keeping the computational requirements within practical limits.
2. Normalization: Pixel values were scaled to the range [0, 1] by dividing by 255. This normalization step was essential for enhancing model convergence and overall performance.
3. Data Augmentation: To increase data variability and reduce overfitting, a range of augmentation techniques was employed, including random rotations, width and height shifts, and horizontal flips. These augmentations improved the model's ability to generalize effectively to new data.
4. Data Splitting: The dataset was divided into training, validation, and test sets. The training set was augmented in real-time, while the validation and test sets were used to monitor the model's performance without augmentation. During training, real-time data augmentation was employed, where augmentation transformations were applied dynamically to each batch of images, enhancing variability without increasing dataset storage requirements.

Figure 2 depicts the data pre-processing pipeline consisting of several critical steps aimed at optimizing image input for skin disease classification. Image Resizing is performed to standardize all images to a dimension of 224 × 224, aligning with the input requirements

Table 1 Class distribution in dermnet dataset

Class	Train	Valid	Test
Acne and Rosacea Photos	831	328	312
Actinic Keratosis Basal Cell Carcinoma and other Tumors	1140	425	288
Atopic Dermatitis Photos	489	147	120
Bullous Disease Photos	441	190	112
Cellulitis Impetigo and other Bacterial Infections	287	122	72
Eczema Photos	1235	371	309
Exanthems and Drug Eruptions	403	136	101
Hair Loss Photos Alopecia and other Hair Diseases	231	84	60
Herpes HPV and other STDs Photos	404	131	102
Light Diseases and Disorders of Pigmentation	566	237	143
Lupus and other Connective Tissue diseases	416	162	102
Melanoma Skin Cancer Nevi and Moles	463	139	116
Nail Fungus and other Nail Disease	1040	312	261
Poison Ivy Photos and other Contact Dermatitis	253	100	61
Psoriasis pictures Lichen Planus and related diseases	1383	531	347
Scabies Lyme Disease and other Infestations and Bites	398	169	102
Seborrheic Keratoses and other Benign Tumors	1362	530	343
Systemic Disease	596	217	152
Tinea Ringworm Candidiasis and other Fungal Infections	1300	390	325
Urticaria Hives	212	64	53
Vascular Tumors	482	145	121
Vasculitis Photos	410	166	105
Warts Molluscum and other Viral Infections	1086	326	272
Total	15428	5422	3979

of advanced CNN models like VGG19, ResNet, and Inception. This step is followed by normalization which scales pixel values to the [0, 1] range. Various Data Augmentation techniques, including random rotations, shifts, and horizontal flips, are applied for Data Augmentation. Finally, the dataset is systematically split into training, validation, and test sets.

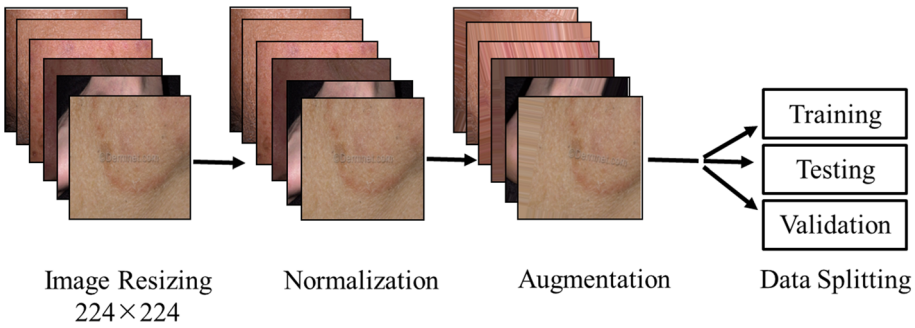

Fig. 2 Data Pre-processing Pipeline for Skin Disease Classification: This pipeline encompasses image resizing, normalization, data augmentation, and dataset splitting

Table 2 summarizes the Data Augmentation techniques used in this study.

3.4 Model architecture

We designed a custom CNN architecture for both binary and 23-ary classification tasks. Our custom CNN architecture was designed to extract complex patterns from input images for accurate classification. All images were resized to 224 × 224 pixels, a standard size widely used by ImageNet-based CNNs like VGG19, ResNet, and EfficientNet. This choice ensures compatibility with pre-trained models and allows a fair comparison with state-of-the-art architectures. While built from scratch to address the specific needs of skin disease classification, our custom CNN leverages these established practices for consistent feature extraction and optimal performance.

The architecture begins with an initial convolutional layer that serves as a basic feature extractor, followed by progressively deeper layers that capture increasingly complex features. Each convolutional layer is followed by an activation function and a max-pooling layer to enhance computational efficiency and focus on prominent attributes. Dropout layers are incorporated to prevent overfitting, and the network culminates in a dense softmax layer for multi-class classification.

3.4.1 Detailed layer-wise architecture

1. Initial Convolutional Layer:
The network starts with a convolutional layer containing 32 filters of size 3 × 3, which is applied to the input image $\mathbf{I}$ to capture low-level features like edges and textures. The convolution operation is defined as:

$$\mathbf{O}_1(i, j) = \sum_{m=0}^{2} \sum_{n=0}^{2} \mathbf{I}(i + m, j + n) \cdot \mathbf{K}_1(m, n) \tag{1}$$

where $\mathbf{K}_1$ represents the kernel for this layer. The ReLU activation function is then applied to introduce non-linearity:

$$\text{ReLU}(x) = \max(0, x) \tag{2}$$

A 2 × 2 max-pooling layer follows, defined as:

$$\mathbf{P}_1(i, j) = \max_{0 \le m < 2, 0 \le n < 2} \mathbf{O}_1(i \cdot 2 + m, j \cdot 2 + n) \tag{3}$$

2. Intermediate Convolutional Layers:

Table 2 Configuration of data augmentation methods used in the image classification pipeline

Parameter	Value
Rotation Range	20 degrees
Width Shift Range	0.2 (20% of width)
Height Shift Range	0.2 (20% of height)
Shear Range	0.2 (20%)
Zoom Range	0.2 (20%)
Horizontal Flip	True
Fill Mode	‘nearest’

The network progresses with a second convolutional layer of 64 filters (size 3×3) to detect intricate patterns, such as lesion contours. This operation is defined as:

$$\mathbf{O}_2(i, j) = \sum_{m=0}^{2} \sum_{n=0}^{2} \mathbf{P}_1(i + m, j + n) \cdot \mathbf{K}_2(m, n) \tag{4}$$

Followed by ReLU activation and max-pooling:

$$\mathbf{P}_2(i, j) = \max_{0 \le m < 2, 0 \le n < 2} \mathbf{O}_2(i \cdot 2 + m, j \cdot 2 + n) \tag{5}$$

3. Deeper Convolutional Layers:
To capture high-level, abstract features such as lesion shapes and pigmentation variations, two additional convolutional layers with 128 filters (size 3×3) are added:

$$\mathbf{O}_3(i, j) = \sum_{m=0}^{2} \sum_{n=0}^{2} \mathbf{P}_2(i + m, j + n) \cdot \mathbf{K}_3(m, n) \tag{6}$$

$$\mathbf{P}_3(i, j) = \max_{0 \le m < 2, 0 \le n < 2} \mathbf{O}_3(i \cdot 2 + m, j \cdot 2 + n) \tag{7}$$

$$\mathbf{O}_4(i, j) = \sum_{m=0}^{2} \sum_{n=0}^{2} \mathbf{P}_3(i + m, j + n) \cdot \mathbf{K}_4(m, n) \tag{8}$$

$$\mathbf{P}_4(i, j) = \max_{0 \le m < 2, 0 \le n < 2} \mathbf{O}_4(i \cdot 2 + m, j \cdot 2 + n) \tag{9}$$

4. Fully Connected Layers:
The output from the final convolutional layer is flattened:

$$\mathbf{v} = \text{Flatten}(\mathbf{P}_4) \tag{10}$$

This vector is then passed through a dense layer of 512 units with ReLU activation:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_h \mathbf{v} + \mathbf{b}_h) \tag{11}$$

where $\mathbf{W}_h$ and $\mathbf{b}_h$ are the weights and biases, respectively. A dropout layer with a rate of 0.5 is applied to prevent overfitting.
5. Output Layer:
For final classification, the dense output is passed through a softmax layer:

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_o \mathbf{h} + \mathbf{b}_o) \tag{12}$$

3.4.2 Hierarchical learning for feature extraction

The custom CNN architecture is designed to hierarchically extract features, progressing from basic to complex patterns critical for accurate skin condition classification. Each layer plays a specific role in learning visual attributes at varying levels of abstraction:

1. Low-Level Features: Initial layers (Conv2D with 32 filters) focus on detecting basic structures such as edges and textures, which form the foundational patterns. Applying ReLU activation introduces non-linearity, enabling the network to capture essential variations in surface textures across different skin types.

2. Intermediate-Level Features: Middle layers (Conv2D with 64 and 128 filters) extract more detailed structures like lesion borders, color transitions, and texture gradients. Max-pooling layers reduce spatial dimensions while preserving key patterns, improving computational efficiency and enhancing focus on distinctive lesion attributes.
3. High-Level Features: Deeper layers (Conv2D with 128 filters) capture complex, disease-specific patterns such as morphology, pigmentation, and intricate texture variations, facilitating differentiation between visually similar conditions. These abstract features provide a comprehensive understanding of each image.
4. Feature Integration via Fully Connected Layers: A dense layer with 512 neurons integrates spatial and feature information from prior layers into a unified feature vector. A dropout layer (rate of 0.5) prevents overfitting, supporting robust generalization.
5. Classification: The final softmax layer outputs a probability distribution, indicating classification confidence across skin condition classes. This hierarchical learning enables the CNN to progressively refine feature maps, capturing nuanced visual cues vital for diagnosis.

3.4.3 Key features extracted by the architecture

The architecture is specifically tailored to extract:

- Low-level features: Edges, textures, gradients, and basic shapes from the initial convolutional layers.
- Intermediate features: Patterns, lesion borders, and regional texture variations indicative of skin conditions in the subsequent layers.
- High-level features: Complex structures and color variations that differentiate among multiple skin diseases, extracted by the deeper layers.

This hierarchical learning approach enables the model to capture a comprehensive range of visual features, from basic structures to complex patterns that are critical for distinguishing between different skin conditions, thereby enhancing its diagnostic accuracy.

3.4.4 CNN model parameter summary

The total number of learnable parameters has been optimized to ensure the model captures complex patterns while maintaining computational efficiency. This design balances model complexity and parameter count to prevent overfitting, providing sufficient capacity for effective learning in both binary and multi-class classification tasks. Table 3 summarizes the learnable parameters of the custom CNN.
Total parameters: 29,070,983
Trainable parameters: 9,690,327
Non-trainable parameters: 0
Optimizer parameters: 19,380,656

Figure 3 shows the visual representation of the custom CNN architecture for both binary and 23-class classification. For binary classification, the model distinguishes between two classes: ringworm and healthy skin. The architecture includes multiple convolutional layers with ReLU activation functions, interspersed with max-pooling layers to reduce spatial dimensions while retaining key features. Following the convolutional layers, a flattening layer converts the 2D feature maps into a 1D vector, which is then passed through a dense layer with 512 units and a dropout layer for regularization. The final output layer utilizes a softmax activation function to provide probabilities for the two classes.

Table 3 Learnable Parameters in the Custom CNN Architecture

Layer type	Description	Output shape	Number of parameters
Conv2D	3×3 kernel, 32 filters	(None, 222, 222, 32)	896
MaxPooling2D	MaxPooling Layer	(None, 111, 111, 32)	0
Conv2D	3×3 kernel, 64 filters	(None, 109, 109, 64)	18,496
MaxPooling2D		(None, 54, 54, 64)	0
Conv2D	3×3 kernel, 128 filters	(None, 52, 52, 128)	73,856
MaxPooling2D		(None, 26, 26, 128)	0
Conv2D	3×3 kernel, 128 filters	(None, 24, 24, 128)	147,584
MaxPooling2D		(None, 12, 12, 128)	0
Flatten	Reshaping Layer	(None, 18432)	0
Dense	Fully connected layer, 512 units	(None, 512)	9,437,696
Dropout	Regularization Layer	(None, 512)	0
Dense (Output)	NUM_CLASSES (Softmax)	(None, NUM_CLASSES)	(512 + 1) × NUM_CLASSES

The same custom CNN architecture is adapted for 23-class classification, where the model is trained to identify 23 different skin diseases. The structure remains consistent, with convolutional layers for feature extraction, max-pooling layers for dimension reduction, and a dense layer for complex pattern learning. The key difference lies in the final output layer, which produces probabilities across the 23 classes, enabling the model to classify a broad range of skin diseases accurately.

The detailed architecture of the custom CNN is shown in Fig. 4. This figure exhibits the type of each layer and its placement within the network. It begins with the input layer, which accepts preprocessed images of specified dimensions. Following the input layer, the network includes multiple convolutional layers (Conv2D) with ReLU activation functions, each responsible for detecting various features from the input images, paired with max-pooling layers (MaxPooling2D) to reduce spatial dimensions and computational complexity while preserving essential information.

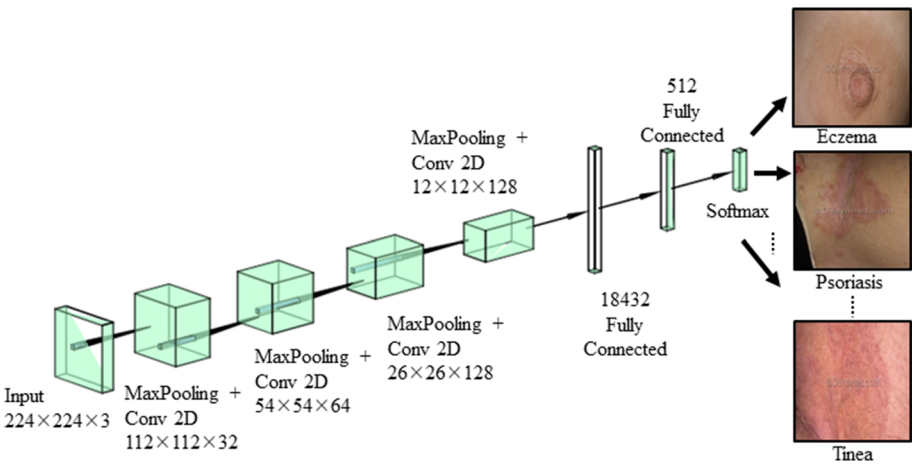

Fig. 3 Custom CNN architecture for 23-class skin disease classification, featuring Conv2D layers, max-pooling, a dense layer with dropout, and a softmax output layer for class probabilities

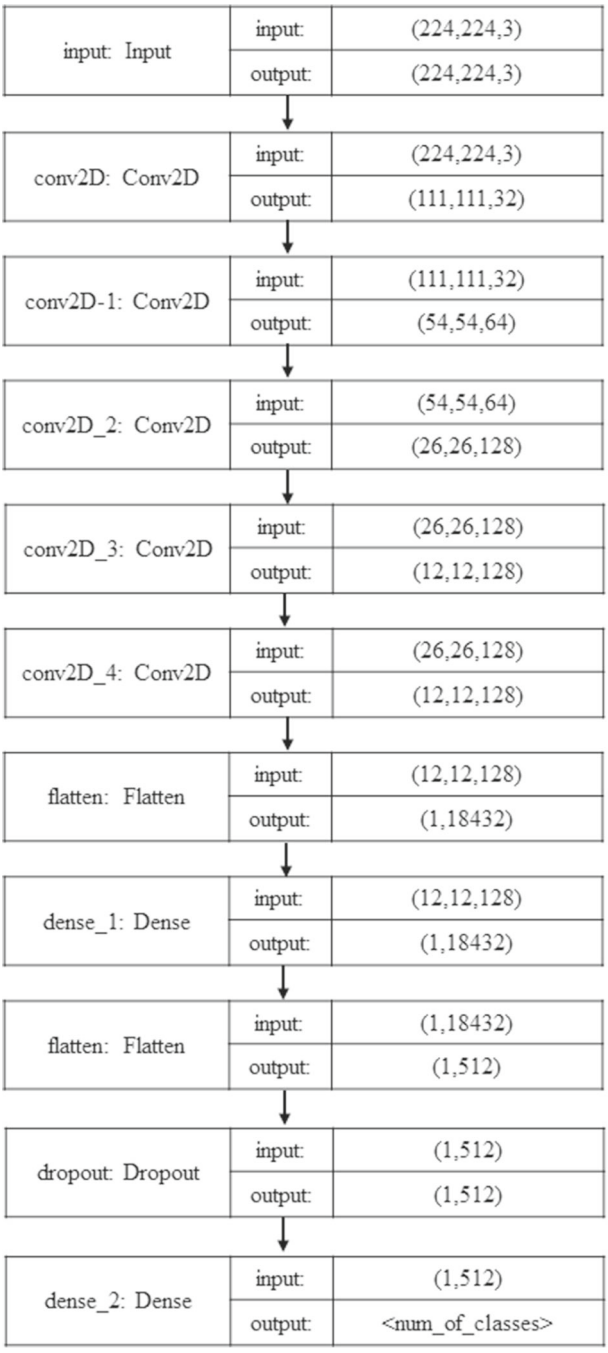

Fig. 4 This sequence of layers diagram shows the sequence of layers in the custom CNN architecture for skin disease classification, featuring four convolutional layers with ReLU activation and concluding with a softmax layer for class probabilities

The architecture progresses through a series of convolutional and max-pooling layers, progressively increasing the number of filters to capture more complex features. The output from the last convolutional layer is flattened into a 1D vector, preparing it for the fully connected dense layers. A dense layer with 512 units and ReLU activation follows, helping to learn high-level representations of the features, while a dropout layer with a 0.5 dropout rate is included to prevent overfitting.

The final layer of the network is a dense layer with a softmax activation function, which outputs the classification probabilities. The figure also details the input and output feature sizes for each layer, providing a comprehensive view of the data transformation process through the network. This architecture highlights the flow of data and the hierarchical feature extraction process in the custom CNN, ensuring clarity in understanding each layer's role and contribution to the overall model.

3.5 Modified CNN architecture

The modified CNN architecture incorporates several enhancements aimed at improving feature extraction, regularization, and model generalization. Key modifications include the addition of batch normalization, increased convolutional depth, distributed dropout layers, and global average pooling. This revised architecture, designed to balance computational efficiency with predictive performance, is described below.

1. Initial Convolutional Layers with Batch Normalization:
    - The network begins with a sequence of convolutional layers of increasing filter sizes: 32, 64, 128, and 256. Each layer utilizes a (3, 3) kernel, ReLU activation, and is followed by batch normalization. Batch normalization normalizes the activations across the batch, stabilizing training, allowing higher learning rates, and helping mitigate the problem of internal covariate shift.
2. Max-Pooling Layers and Distributed Dropout:
    - Each convolutional layer is followed by a max-pooling layer with a (2, 2) pool size, reducing spatial dimensions and retaining essential features. To further regularize the network, a dropout layer (rate = 0.25) is applied after each pooling layer.
    - By distributing dropout throughout the network, we effectively reduce overfitting and improve generalization, especially given the limited size of available training data. This approach allows the model to avoid dependency on specific neurons, thus encouraging robust feature learning across all layers.
3. Increased Convolutional Depth:
    - The model's final convolutional layer increases the filter depth to 256, allowing the network to capture more complex patterns within the data. This additional layer provides a deeper, more abstracted representation of the features, which is especially beneficial for handling variations in class features in large, multi-class datasets.
4. Global Average Pooling:
    - The fully connected layers traditionally used in CNN architectures are replaced with a Global Average Pooling (GAP) layer. GAP reduces each feature map to a single value, capturing the spatially averaged information across each filter and drastically reducing the number of trainable parameters.

5. Fully Connected Layer and Final Classification:
    - Following GAP, a fully connected dense layer with 512 units and ReLU activation aggregates the learned global features. A dropout layer (rate = 0.4) further regularizes the network, and the model's final dense layer utilizes softmax activation with 23 output units, aligning with the number of target classes in the dataset.

This modified CNN design prioritizes stability, feature abstraction, and overfitting reduction through batch normalization, dropout, and global average pooling.

Table 4 summarizes the sequence of layers of the Modified CNN architecture.

3.6 Class-weighted cnn architecture

We implemented the modified CNN trained with class weights to address the imbalanced distribution of skin disease classes. This approach ensures that the model places greater emphasis on minority classes, thereby enhancing its ability to learn meaningful features across all categories. To mitigate the class imbalance, class weights were computed and applied to the loss function during training. This technique assigns higher weights to classes with fewer samples, ensuring that these classes contribute proportionally to the total loss. Specifically, the class weights were calculated as follows:

$$\text{class\_weight}[i] = \frac{\text{total\_samples}}{\text{num\_classes} \times \text{samples\_per\_class}[i]} \tag{13}$$

Table 4 Layer-wise architecture of the Modified custom CNN model with input and output shapes at each layer

Layer	Input Shape	Output Shape	Details
Input Layer	(224, 224, 3)	(224, 224, 3)	RGB image
Conv2D (32 filters)	(224, 224, 3)	(222, 222, 32)	3 × 3 filters, ReLU activation
BatchNormalization	(222, 222, 32)	(222, 222, 32)	Normalizes feature maps
MaxPooling2D	(222, 222, 32)	(111, 111, 32)	2 × 2 pooling
Dropout (0.25)	(111, 111, 32)	(111, 111, 32)	Regularization
Conv2D (64 filters)	(111, 111, 32)	(109, 109, 64)	3 × 3 filters, ReLU activation
BatchNormalization	(109, 109, 64)	(109, 109, 64)	Normalizes feature maps
MaxPooling2D	(109, 109, 64)	(54, 54, 64)	2 × 2 pooling
Dropout (0.25)	(54, 54, 64)	(54, 54, 64)	Regularization
Conv2D (128 filters)	(54, 54, 64)	(52, 52, 128)	3 × 3 filters, ReLU activation
BatchNormalization	(52, 52, 128)	(52, 52, 128)	Normalizes feature maps
MaxPooling2D	(52, 52, 128)	(26, 26, 128)	2 × 2 pooling
Dropout (0.25)	(26, 26, 128)	(26, 26, 128)	Regularization
Conv2D (256 filters)	(26, 26, 128)	(24, 24, 256)	3 × 3 filters, ReLU activation
BatchNormalization	(24, 24, 256)	(24, 24, 256)	Normalizes feature maps
MaxPooling2D	(24, 24, 256)	(12, 12, 256)	2 × 2 pooling
Dropout (0.25)	(12, 12, 256)	(12, 12, 256)	Regularization
GlobalAveragePooling2D	(12, 12, 256)	(256,)	Averages across each channel

where i represents each class in the dataset. These weights were then used in the categorical cross-entropy loss function, guiding the model to focus more on the underrepresented classes during the training process.

3.7 Transfer learning with SOTA models

In this study, we employed transfer learning with SOTA deep learning models, specifically VGG19, InceptionV3, DenseNet121, ResNet50, and EfficientNetB0. Each of these models has demonstrated strong performance in image classification tasks and provides robust feature extraction capabilities when used as a base model. Leveraging their pre-trained weights from ImageNet, we aimed to adapt their knowledge to classify our dataset effectively.

Each SOTA model was modified to tailor it to the specific requirements of our dataset:

1. Loading Pre-trained Weights: Each model was initialized with pre-trained weights from ImageNet to benefit from prior learning on a large, diverse dataset. This initialization aids in transferring general visual features such as edges, shapes, and textures, which are useful for classifying images in our dataset.
2. Freezing Base Layers: To preserve the learned features and prevent overfitting on our relatively smaller dataset, the convolutional layers of each base model were frozen, meaning their weights were not updated during training. Only the final classification layers were trained on the new dataset. This approach allows the model to leverage high-quality feature maps from the lower layers while focusing on adapting to the specific classes in our dataset at the higher layers.
3. Adding Custom Classification Layers: On top of each base model, custom fully connected layers were added to transform the output of the frozen layers into predictions for our dataset. The following modifications were made to the top of each base model:
    - A GAP layer was applied to reduce the feature maps from the base model. GAP replaces the need for fully connected layers, reducing the risk of overfitting and the model's parameter count.
    - A dense layer with 512 units and ReLU activation was added as a high-level feature extractor.
    - A Dropout layer with a rate of 0.5 was included to mitigate overfitting by randomly deactivating neurons during training. The final output layer is a softmax layer with neurons equal to the number of classes in our dataset, allowing for multi-class classification.

3.8 Transfer learning with vision transformer

This work incorporates a custom VViT model as part of our transfer learning approach for classifying skin disease images. The model architecture draws from the principles established in Dosovitskiy et al.'s [37] where the authors demonstrate that transformer-based models, typically applied in natural language processing, can be adapted effectively for visual recognition tasks. The ViT model utilizes a transformer-based approach for image classification by processing 16×16 patches instead of individual pixels, allowing for efficient pattern recognition compared to traditional CNNs.

- Image Patching: Images are divided into 196 non-overlapping patches (for a 224×224 image), each projected into a 64-dimensional vector through a convolutional layer, which reduces dimensionality for encoding.

- Patch Encoding: The Patch Encoder flattens and encodes the patches with positional embeddings to retain spatial information. A dense layer projects them into a fixed-dimensional space.
- Transformer Blocks: The model features 8 transformer layers with 4 attention heads each, incorporating:
  - Multi-Head Self-Attention for learning relationships between distant patches.
  - Feed-Forward Network (FFN) with a two-layer MLP, GELU activation, and Dropout to prevent overfitting.
  - Residual Connections to enhance gradient flow.

  These components convert patch data into high-level features.
- Classification Head: A GAP layer aggregates features, followed by a dense layer with 512 units and GELU activation, and a final dense layer with 23 units (for class probabilities) activated by softmax.

3.9 Training and evaluation

The model was trained using a batch size of 32 and run for 50 epochs. The Adam optimizer was employed to minimize the categorical cross-entropy loss function. During training, we monitored the accuracy and loss on both the training and validation sets. The performance of the trained model was then evaluated on the held-out test set to assess its generalization capabilities.

3.10 Experimental setup

The initial experiments were conducted on Google Colab, where we used an NVIDIA T4 GPU to train the model on a subset of the Dermnet dataset, focusing on rapid prototyping and preliminary analysis. For the main experiments, including model architecture modifications, SOTA comparisons, and custom weight adjustments, we utilized a Jupyter Notebook environment on a laptop with an NVIDIA RTX 4050 GPU. This setup allowed us to train on the full dataset, perform comprehensive evaluations, and refine model parameters for optimal performance.

3.10.1 Training configuration

The model was trained with a batch size of 32 for 50 epochs. We used the Adam optimizer for training, which adjusts the learning rate dynamically during the training process. The Adam optimizer is defined as:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{14}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the estimates of the first and second moments of the gradients, respectively.

3.10.2 Train-test split

The dataset was split into training and testing sets with a ratio of 80:20. This ensures that the model is evaluated on unseen data, providing a reliable estimate of its performance. For comparison, the model was also implemented using a 70:30 split.

3.10.3 Evaluation metric

To thoroughly assess our model's performance in multiclass skin disease classification, we used multiple metrics beyond accuracy, including precision, recall, F1-score, and AUC-ROC. These metrics help balance true and false predictions, minimizing both false positives and false negatives, essential in medical diagnostics.

- **Accuracy**: Measures overall correct predictions:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Samples}} \tag{15}$$

  While accuracy gives an overall snapshot, it may be limited for imbalanced data.
- **Precision**: Indicates reliability in predicting positive cases:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{16}$$

  Precision is critical to minimizing unnecessary treatments.
- **Recall (Sensitivity)**: Reflects the model's ability to identify true positives:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{17}$$

  High recall reduces missed cases, essential in medical contexts.
- **F1-Score**: Balances precision and recall, especially useful for imbalanced datasets:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

  This metric is valuable when dealing with imbalanced datasets, as it considers both false positives and false negatives in a single metric, providing a more balanced view of model performance.
- **False Negative Rate (FNR)**: Measures the proportion of actual positives incorrectly classified as negatives, indicating missed detections:

$$\text{FNR} = \frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}} \tag{19}$$

- **AUC-ROC**: Assesses the model's ability to distinguish between classes across thresholds. An AUC-ROC close to 1 indicates strong discriminatory power, crucial for robust decision-making in medical applications.

4 Results

The custom-designed CNN demonstrated remarkable performance in the binary classification task, achieving an accuracy of 98.9% in distinguishing between ringworm and healthy skin images. This high accuracy suggests that the model has successfully learned to identify the distinctive features of ringworm infections, potentially offering a valuable tool for rapid screening and diagnosis in clinical settings. The strong performance in binary classification can be attributed to several factors, including the effective data augmentation techniques employed to expand the limited dataset, the carefully designed CNN architecture that captures relevant features at multiple scales, and the clear visual distinctions between ringworm-affected and healthy skin. These visual distinctions include characteristic ring-shaped lesions

with raised, scaly borders contrasting with the uniform texture of healthy skin, reddish or discolored patches in infected regions compared to normal skin tone, clear demarcation between affected and unaffected areas with distinctive circular patterns, and the presence of scaling, flaking, or crusting in ringworm lesions versus the smooth appearance of healthy skin. These pronounced morphological differences provided strong discriminative features for the CNN to learn from, contributing to its high classification accuracy.

However, when the same CNN architecture was applied to the more complex task of 23-ary classification, encompassing a diverse range of skin conditions, the model's performance decreased significantly, achieving an accuracy of 35.23%. While this accuracy is substantially higher than random chance (which would be approximately 4.35% for 23 classes), it indicates the increased difficulty of differentiating among a larger number of skin diseases, many of which may share similar visual characteristics. This drop in performance from binary to multi-class classification highlights the challenges inherent in developing a single model capable of accurately diagnosing a wide spectrum of dermatological conditions.

4.1 Binary classification

In the binary classification task, models were assessed based on their ability to distinguish between fungal and non-fungal skin conditions. Key performance metrics-Accuracy, Precision, Recall, F1-Score, and False Negative Rate (FNR)-are presented in Table 5.

Confusion matrices Figure 5 presents the confusion matrices for the binary classification models: Custom CNN and ViT. These matrices display the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each model. The Custom CNN matrix demonstrates a strong concentration of TPs and TNs, highlighting its accuracy with minimal misclassifications. In contrast, the ViT confusion matrix shows a higher number of FPs and a complete absence of TNs, indicating challenges in distinguishing between classes and a tendency to misclassify certain samples.

Training and validation accuracy and loss curves Figure 6 illustrates the training and validation accuracy and loss curves for Custom CNN and ViT.

Summary of results

- For binary classification of skin fungal infections, Custom CNN is the best-performing model, offering the highest accuracy, precision, recall, and F1-score. DenseNet121 also delivers robust performance and is a strong alternative for this task.
- InceptionV3 provides a balanced approach, with slightly lower performance compared to the top models but still offering a good compromise between precision and recall.

Table 5 Performance metrics for different models in binary classification

Model	Accuracy	Precision	Recall	F1-Score	FNR
VGG19	96.11%	96.15%	96.11%	95.70%	3.89%
EfficientNetB0	90.52%	81.95%	90.52%	86.02%	9.48%
InceptionV3	97.63%	97.59%	97.63%	97.53%	2.37%
ResNet50	90.52%	81.95%	90.52%	86.02%	9.48%
DenseNet121	98.14%	98.18%	98.14%	98.05%	1.86%
Custom CNN	**98.82%**	**98.90%**	**98.82%**	**98.84%**	**1.18%**
Vision Transformer	90.52%	81.95%	90.52%	86.02%	9.48%

Bold signify our results of Custom and Improvised CNN

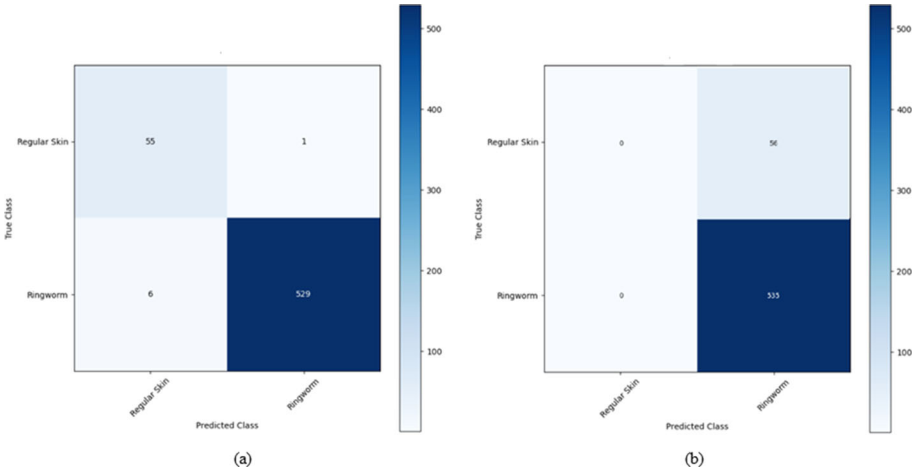

Fig. 5 Confusion Matrices for Binary Classification. (a): Using Custom CNN, (b): Using Vision Transformer

- EfficientNetB0, ResNet50, and Vision Transformer show potential in terms of recall, but their lower precision requires optimization to make them more reliable for clinical or practical applications, where minimizing false positives is crucial. These results demonstrate the importance of selecting a model that not only maximizes accuracy but also strikes an appropriate balance between precision and recall, depending on the specific requirements of the task at hand.

4.2 Multiclass classification

For multiclass classification, models were evaluated across several skin disease categories. Table 6 presents Accuracy, Precision, Recall, F1-Score, and FNR as percentages, highlighting model performance across all classes.

Confusion matrices Figures 7, 8, 9 and 10 present the confusion matrices for the multiclass classification model using Custom CNN, Modified CNN, Class Weighted CNN, and ViT respectively.

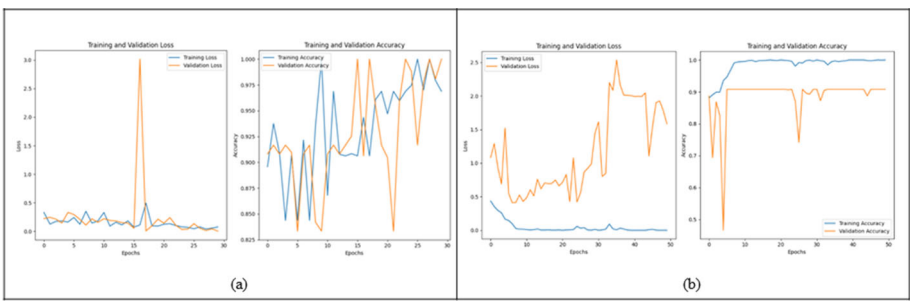

Fig. 6 Training and Validation Loss and Accuracy for Binary classification. (a): Using Custom CNN, (b): Using Vision Transformer

Table 6 Performance Metrics for Different Models in Multiclass Classification

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FNR (%)
VGG19	17.59	9.75	17.59	10.51	82.41
EfficientNetB0	8.98	1.00	9.00	1.00	91.00
InceptionV3	13.19	7.98	13.19	6.74	86.81
ResNet50	8.72	0.76	8.72	1.40	91.28
DenseNet121	18.04	10.84	18.04	9.91	81.96
Custom CNN	**33.88**	**35.20**	**33.88**	**31.58**	**66.12**
Improvised CNN	**35.23**	**41.22**	**35.23**	**35.81**	**64.77**
Weighted CNN	20.36	34.50	20.36	20.74	79.64
Vision Transformer	7.16	4.00	7.00	2.00	93.00
70:30 Split	*31.11*	*33.27*	*31.11*	*28.57*	*68.89*

Bold signify our results of Custom and Improvised CNN

Training and validation accuracy and loss curves Figure 11 illustrates the training and validation accuracy and loss curves for Custom CNN, Modified CNN, Class Weighted CNN, and ViT.

AUC-ROC curves The ROC for the Modified CNN presented in Figure 12 demonstrates the model's performance in distinguishing between classes, with the area under the curve (AUC)

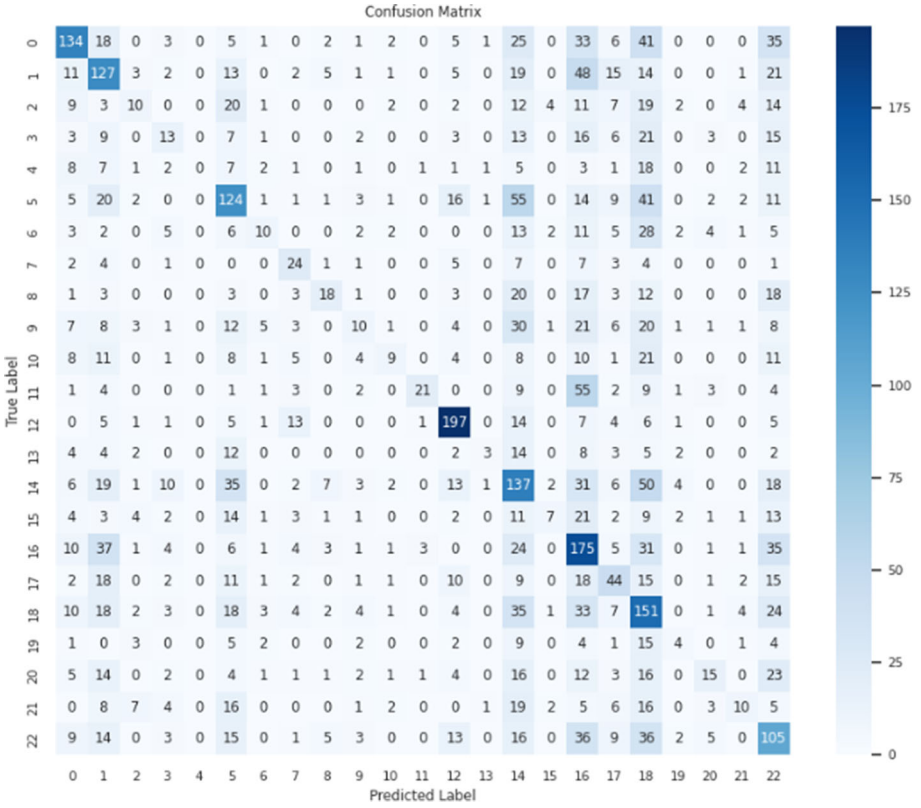

Fig. 7 Confusion Matrix for Multiclass classification using custom CNN

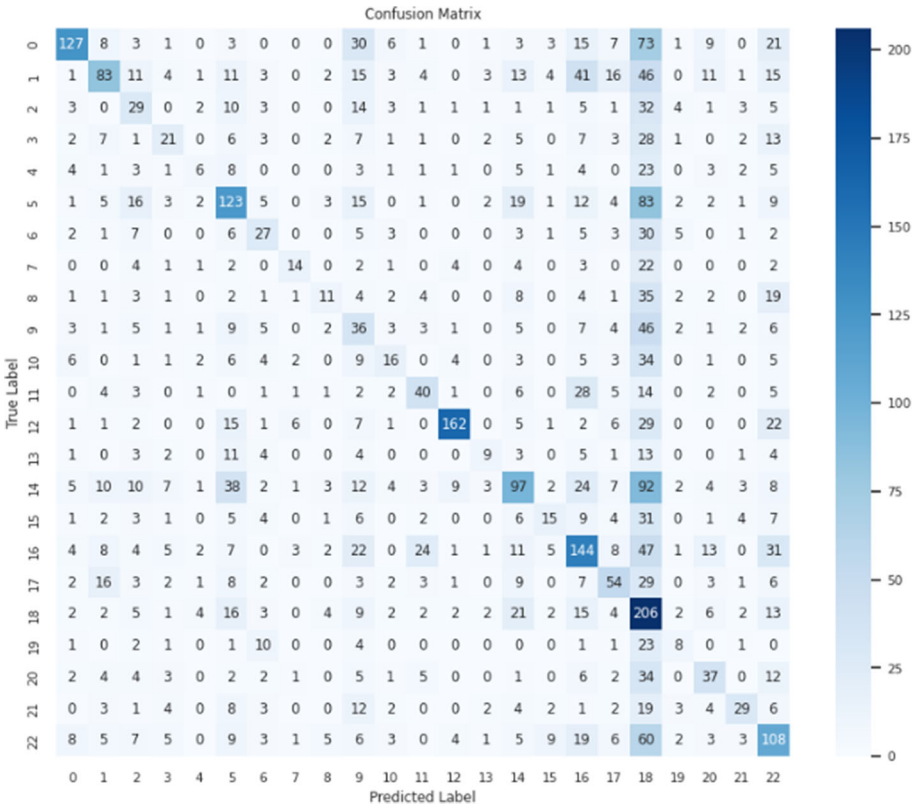

Fig. 8 Confusion Matrix for Multiclass classification using Modified CNN

providing a measure of classification accuracy. A higher AUC indicates strong predictive capability, with the curve approaching the top-left corner reflecting the model's effectiveness in minimizing false positives and false negatives. This ROC analysis highlights the Modified CNN's ability to achieve a balance between sensitivity and specificity in classification tasks.

Analysis of results

- Custom CNN and Improvised CNN: The Custom CNN achieved a notable accuracy of 33.88%, while the Improvised CNN model performed slightly better, with a 35.23% accuracy and a higher F1-score, making it the most effective model in this multiclass task. The reduced false negative rate in the Improvised CNN suggests better class differentiation, contributing to overall performance gains.
- Class-Weighted CNN: With a 20.36% accuracy and relatively high precision, the Class-Weighted CNN showed improved performance in minority classes due to the applied weighting, though its recall and F1-score remain lower than the Improvised CNN.
- Other Models (VGG19, InceptionV3, DenseNet121, Vision Transformer): These models exhibited lower accuracy and recall rates, reflecting challenges in distinguishing between classes in this multiclass setup. Notably, ViT had a high false negative rate, indicating its difficulty in handling multiclass distinctions within this dataset.

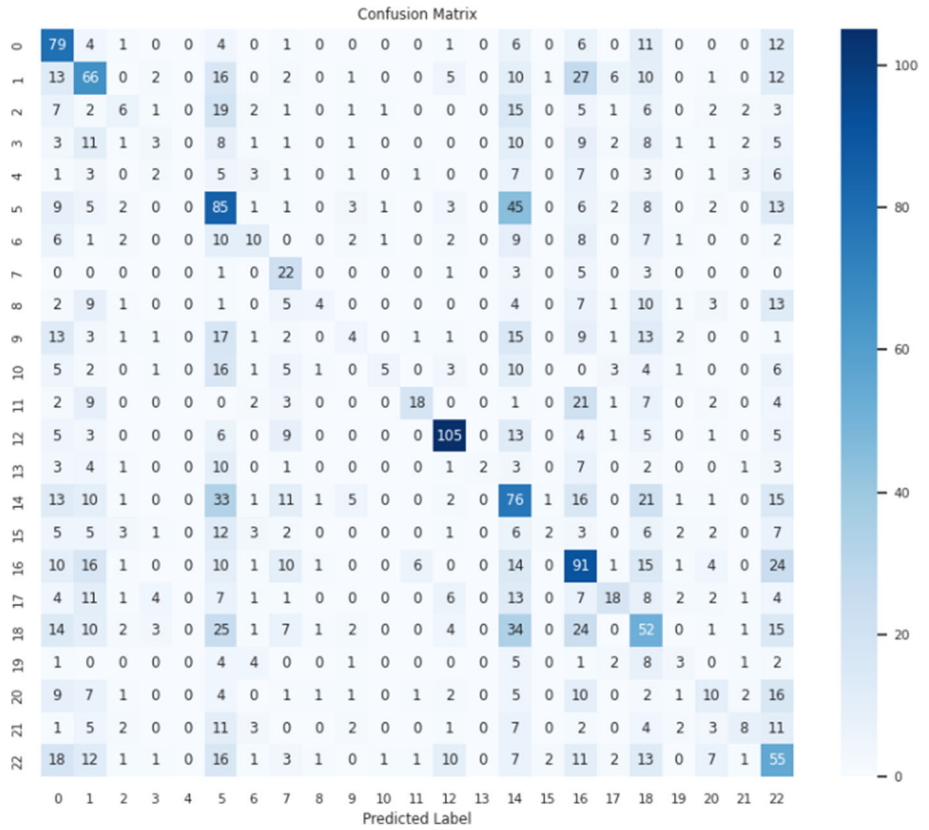

Fig. 9 Confusion Matrix for Multiclass classification using Class-Weighted CNN

4.3 Model interpretability

To enhance the interpretability of our custom CNN model, we incorporated Grad-CAM analysis, which provides visual explanations of the model's predictions by highlighting the specific regions of the input images that influence the decision-making process. This technique helps in visualizing the attention areas where the model focuses while classifying different skin conditions. The Grad-CAM visualizations in our study revealed that the network predominantly concentrates on clinically relevant areas in the images, thereby increasing the confidence in the model's predictions. These insights demonstrate the potential of Grad-CAM to support dermatologists in verifying the model's decisions, offering an additional layer of transparency to the automated classification process. Figure 13 presents the heatmaps across various models.

5 Discussion

This study highlights both the potential and limitations of custom CNNs for skin disease classification in dermatology. The model's high accuracy in binary classification (98.9%) sharply contrasts with its performance in 23-class classification (35.23%), emphasizing the challenges of distinguishing diverse skin conditions. The Dermnet dataset poses further issues, as class

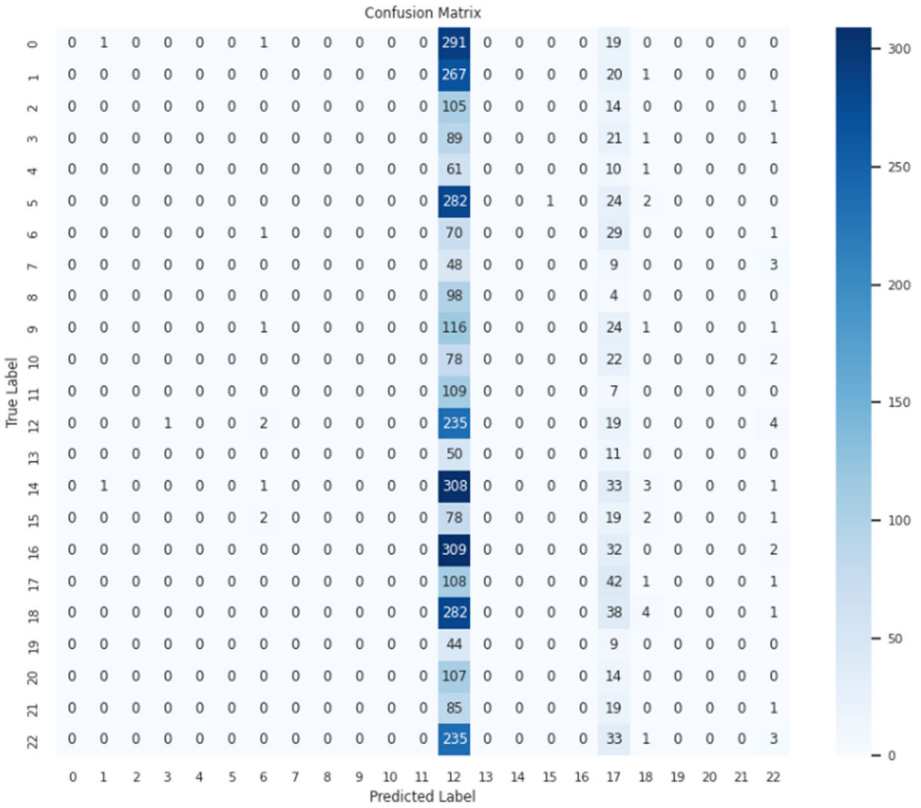

Fig. 10 Confusion Matrix for Multiclass classification using ViT

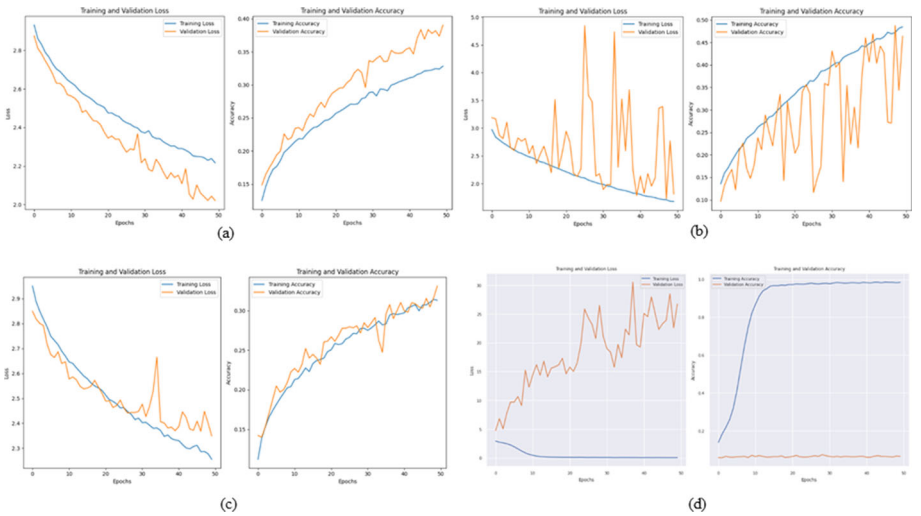

Fig. 11 Training and Validation Loss and Accuracy. (a) Custom CNN (b) Modified CNN (c) Class Weighted CNN (d) ViT

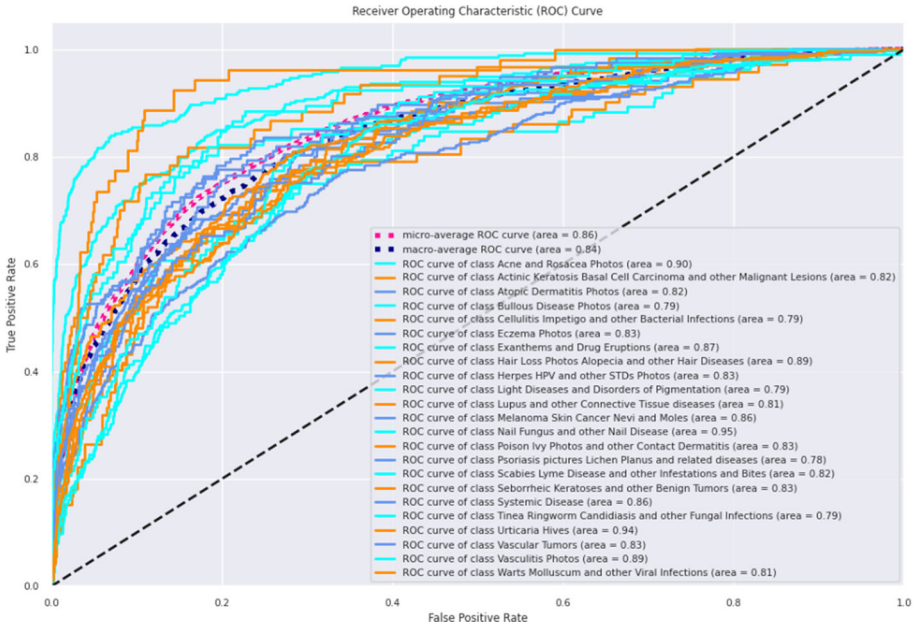

Fig. 12 AUC-ROC of Modified CNN

imbalances (e.g., many eczema vs. few rarer conditions) can bias predictions, and inter-class similarities (e.g., ringworm vs. eczema) make classification challenging. Additionally, high intraclass variance-due to factors like skin tone and condition severity-adds complexity, requiring models that generalize well across diverse cases.

5.1 Binary classification success and clinical potential

In binary classification, the custom CNN performed excellently, accurately distinguishing between ringworm and healthy skin (98.9%), supported by effective data augmentation and architecture design, echoing findings in [8]. This success suggests AI's promise for dermatology in well-defined tasks, aiding initial diagnosis for visually distinct conditions.

5.2 Challenges in multi-class classification

The lower accuracy in the 23-class task (35.23%) shows the difficulty in differentiating similar conditions, a challenge noted by prior studies in medical imaging [38–40]. Real-

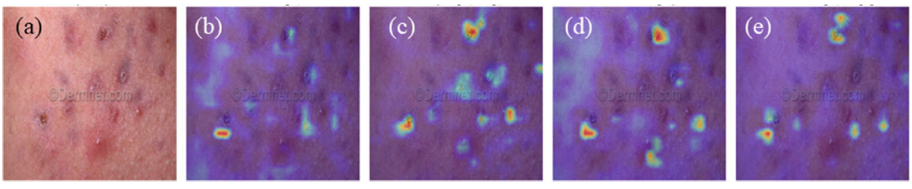

Fig. 13 Grad-CAM Visualizations of a Correctly Classified Image Across Multiple Models. (a) Original Image (b) Using Custom CNN (c) Using Modified Custom CNN (d) Using Class Weighted CNN (e) Using Model trained on 70:30 Split

world dermatological diagnosis relies on patient history and other contextual data, which AI currently lacks, as discussed in [41].

5.3 Dataset limitations and advanced techniques

Dataset limitations are a factor here; a larger, diverse dataset could help the model learn subtle features essential for multiclass classification, as [42] suggests. Future research could explore ensemble methods and hybrid approaches (e.g., CNNs with SVMs or decision trees) for improved classification, as highlighted in [22].

5.4 Custom CNN vs. SOTA models

The custom CNN design aimed to capture dermatology-specific features like lesion shape and texture. While ViTs are powerful for capturing long-range dependencies, they may overlook local patterns key to dermatology. Additionally, the custom CNN's simpler architecture reduced computational load, suited to clinical settings with limited resources. SOTA models, like ViTs, often require high computational power, limiting their practicality in low-resource environments. Moreover, our custom CNN showed less overfitting on a moderate dataset, enhancing generalizability.

5.5 Clinical integration and deployment challenges

Deploying AI models in clinical settings involves regulatory, ethical, and interpretability considerations [38]. Grad-CAM, which provides visual explanations of CNN predictions, enhances model transparency, crucial for clinical acceptance [43]. Variations in patient demographics and clinical protocols further complicate model deployment in dermatology. Integrating patient metadata, as suggested by [38], could improve generalizability across diverse populations.

5.6 Future directions

Future research should focus on expanding datasets, refining CNN architectures, and potentially incorporating multi-modal inputs (e.g., patient metadata) to enhance model accuracy in multiclass settings. Balancing model complexity with clinical applicability is essential, as [44] emphasizes. Hierarchical classification, grouping similar diseases before further categorization, may help with visually similar conditions in multi-class diagnosis.

6 Conclusion

This investigation into custom Convolutional Neural Network (CNN) architectures for skin disease classification sheds light on the potential and limitations of deep learning for dermatological diagnosis. The impressive accuracy achieved in binary classification (98.9%) showcases the effectiveness of CNNs in differentiating between specific skin conditions, particularly for well-defined problems. This success suggests that AI-powered tools could be valuable for rapid screening and initial diagnosis of certain skin diseases.

However, the challenges encountered in the 23-class classification task, resulting in a significant drop to 35.23% accuracy, highlight the complexities involved in developing comprehensive skin disease diagnostic systems. This performance gap underscores the need for more advanced approaches to handle a broader spectrum of dermatological conditions.

Despite these obstacles, the study provides a strong foundation for future research in AI-assisted dermatology. The results indicate that CNNs are capable of learning meaningful features from skin images, even in intricate multi-class scenarios. Moving forward, researchers should focus on expanding datasets, exploring more sophisticated model architectures, and potentially incorporating additional contextual information to enhance multi-class classification performance.

In conclusion, this research adds valuable insights to the growing field of AI-assisted dermatology, demonstrating both the promise and challenges of applying deep learning to skin disease diagnosis. As the field progresses, AI tools are likely to play an increasingly significant role in supporting dermatologists, potentially leading to more accurate and efficient diagnoses in clinical practice.

Author Contributions Conceptualization was done by Pragya Gupta (PG), Jagannath Nirmal (JN), and Ninad Mehendale (NM). The experimentation work was done by PG. All the summarization was performed by PG, JN, and NM. The manuscript draft was prepared by PG, and corrections were made by JN and NM. Data analysis and graphics designing were done by PG.

Funding NA

Data Availability NA

Materials Availability NA

Code Availability Will be provided upon request.

Declarations

Conflict of interest/Competing interests: The Authors declare that there is no conflict of interest.

Ethics approval and consent to participate: NA

References

1. Seth D, Cheldize K, Brown D, Freeman EE (2017) Global burden of skin disease: inequities and innovations. Current Dermatology Reports 6:204–210
2. Maji HS, Chatterjee R, Das D, Maji S (2023) Fungal infection: An unrecognized threat. In: Viral, parasitic, bacterial, and fungal infections, Elsevier, pp 625–644
3. Mayser P (2022) Fungal infections. In: Braun-Falco′ S dermatology, Springer, pp 249–284
4. Fuller L, Barton R, Mohd Mustapa M, Proudfoot L, Punjabi S, Higgins E, Hughes J, Sahota A, Griffiths M, McDonagh A et al (2014) British association of dermatologists' guidelines for the management of tinea capitis 2014. Br J Dermatol 171(3):454–463
5. Reich A, Schwartz RA, Szepietowski JC (2009) Complications of superficial mycoses. Sequelae and Long-Term Consequences of Infectious Diseases:407–413
6. Gupta P, Nirmal J, Mehendale N (2024) A survey of recent advances in analysis of skin images. Evol Intell:1–24
7. Bajwa MN, Muta K, Malik MI, Siddiqui SA, Braun SA, Homey B, Dengel A, Ahmed S (2020) Computer-aided diagnosis of skin diseases using deep neural networks. Appl Sci 10(7):2488
8. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118

9. Maron RC, Weichenthal M, Utikal JS, Hekler A, Berking C, Hauschild A, Enk AH, Haferkamp S, Klode J, Schadendorf D et al (2019) Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks. Eur J Cancer 119:57–65
10. Myslicka M, Kawala-Sterniuk A, Bryniarska A, Sudol A, Podpora M, Gasz R, Martinek R, Kahankova Vilimkova R, Vilimek D, Pelc M et al (2024) Review of the application of the most current sophisticated image processing methods for the skin cancer diagnostics purposes. Arch Dermatol Res 316(4):99
11. Javed R, Rahim MSM, Saba T, Rehman A (2020) A comparative study of features selection for skin lesion detection from dermoscopic images. Network Model Anal Health Inf Bioinformatics 9(1):4
12. Li Z, Koban KC, Schenck TL, Giunta RE, Li Q, Sun Y (2022) Artificial intelligence in dermatology image analysis: current developments and future trends. J Clin Med 11(22):6826
13. Goel S (2020) Dermnet. Accessed: 4-Nov-2024. https://www.kaggle.com/datasets/shubhamgoel27/dermnet
14. Biswas S (2023) Skin-Disease-Dataset. Kaggle. Accessed: 4-Nov-2024. https://doi.org/10.34740/KAGGLE/DSV/6695743. https://www.kaggle.com/dsv/6695743
15. Shah S (2023) Ringworm. Accessed: 4-Nov-2024. https://www.kaggle.com/datasets/shubhamtheshah/ringworm-temp
16. Gupta P, Nirmal J, Mehendale N (2024) A survey on computer vision approaches for automated classification of skin diseases. Multimed Tool Appl:1–33
17. Sreedhar B, BE, MS, Kumar MS (2020) A comparative study of melanoma skin cancer detection in traditional and current image processing techniques. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE, pp 654–658
18. Wei L, Ding K, Hu H (2020) Automatic skin cancer detection in dermoscopy images based on ensemble lightweight deep learning network. IEEE Access 8:99633–99647
19. Maniraj S, Sardarmaran P (2021) Classification of dermoscopic images using soft computing techniques. Neural Comput Appl 33(19):13015–13026
20. Alam J (2021) An efficient approach for skin disease detection using deep learning. In: 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, pp 1–8
21. Obayya M, Alhebri A, Maashi M, Salama SA, Mustafa Hilal A, Alsaid MI, Osman AE, Alneil AA (2023) Henry gas solubility optimization algorithm based feature extraction in dermoscopic images analysis of skin cancer. Cancers 15(7):2146
22. Anggriandi D, Utami E, Ariatmanto D (2023) Comparative analysis of cnn and cnn-svm methods for classification types of human skin disease. Sinkron: jurnal dan penelitian teknik informatika 8(4):2168–2178
23. Allugunti VR (2022) A machine learning model for skin disease classification using convolution neural network. Int J Comput Program Database Manag 3(1):141–147
24. Wei M, Wu Q, Ji H, Wang J, Lyu T, Liu J, Zhao L (2023) A skin disease classification model based on densenet and convnext fusion. Electronics 12(2):438
25. Prottasha Sazzadul Islam M, Mahjabin Farin S, Bulbul Ahmed M, Zihadur Rahman M, Kabir Hossain A, Shamim Kaiser M (2023) Deep learning-based skin disease detection using convolutional neural networks (cnn). In: The fourth industrial revolution and beyond: select proceedings of IC4IR+, Springer, pp 551–564
26. Di Biasi L, De Marco F, Auriemma Citarella A, Castrillón-Santana M, Barra P, Tortora G (2023) Refactoring and performance analysis of the main cnn architectures: using false negative rate minimization to solve the clinical images melanoma detection problem. BMC Bioinformatics 24(1):386
27. Yadav R, Bhat A (2024) A systematic literature survey on skin disease detection and classification using machine learning and deep learning. Multimed Tool Appl:1–32
28. Di Biasi L, De Marco F, Auriemma Citarella A, Barra P, Piotto Piotto S, Tortora G (2022) Hybrid approach for the design of cnns using genetic algorithms for melanoma classification. In: International conference on pattern recognition, Springer, pp 514–528
29. Sadik R, Majumder A, Biswas AA, Ahammad B, Rahman MM (2023) An in-depth analysis of convolutional neural network architectures with transfer learning for skin disease diagnosis. Healthcare Analytics 3:100143
30. Schielein MC, Christl J, Sitaru S, Pilz AC, Kaczmarczyk R, Biedermann T, Lasser T, Zink A (2023) Outlier detection in dermatology: performance of different convolutional neural networks for binary classification of inflammatory skin diseases. J Eur Acad Dermatol Venereol 37(5):1071–1079
31. Pérez E, Ventura S (2022) An ensemble-based convolutional neural network model powered by a genetic algorithm for melanoma diagnosis. Neural Comput Appl 34(13):10429–10448
32. Abbas Q, Daadaa Y, Rashid U, Ibrahim ME (2023) Assist-dermo: A lightweight separable vision transformer model for multiclass skin lesion classification. Diagnostics 13(15):2531
33. Yang G, Luo S, Greer P (2023) A novel vision transformer model for skin cancer classification. Neural Process Lett 55(7):9335–9351

34. Alyas T, Alissa K, Mohammad AS, Asif S, Faiz T, Ahmed G (2022) Innovative fungal disease diagnosis system using convolutional neural network. Comput Mater Continua 73(3)
35. Furqon A, Malik K, Fajri FN (2024) Detection of eight skin diseases using convolutional neural network with mobilenetv2 architecture for identification and treatment recommendation on android application. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) 10(2):373–384
36. Balasundaram A, Shaik A, Alroy BR, Singh A, Shivaprakash S (2024) Genetic algorithm optimized stacking approach to skin disease detection. IEEE Access
37. Dosovitskiy A (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv:2010.11929
38. Liu Y, Jain A, Eng C, Way DH, Lee K, Bui P, Kanada K, Oliveira Marinho G, Gallegos J, Gabriele S et al (2020) A deep learning system for differential diagnosis of skin diseases. Nat Med 26(6):900–908
39. Weng W-H, Deaton J, Natarajan V, Elsayed GF, Liu Y (2020) Addressing the real-world class imbalance problem in dermatology. In: Machine learning for health, PMLR, pp 415–429
40. Kumar Y, Koul A, Singla R, Ijaz MF (2023) Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. J Ambient Intell Humaniz Comput 14(7):8459–8486
41. Dulmage B, Tegtmeyer K, Zhang MZ, Colavincenzo M, Xu S (2021) A point-of-care, real-time artificial intelligence system to support clinician diagnosis of a wide range of skin diseases. J Investig Dermatol 141(5):1230–1235
42. Cheplygina V, De Bruijne M, Pluim JP (2019) Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal 54:280–296
43. Nunnari F, Kadir MA, Sonntag D (2021) On the overlap between grad-cam saliency maps and explainable visual features in skin cancer images. In: International cross-domain conference for machine learning and knowledge extraction, Springer, pp 241–253
44. Roy AG, Ren J, Azizi S, Loh A, Natarajan V, Mustafa B, Pawlowski N, Freyberg J, Liu Y, Beaver Z et al (2022) Does your dermatology classifier know what it doesn't know? Detecting the long-tail of unseen conditions. Med Image Anal 75:102274