

Classification of Skin Disease from Skin Images Using Transfer Learning Technique

Pavithra Devi P. M.E. Students

Department of Computer Science and Engineering, MRK Institute of Technology, Kattumannarkoil, India

Bavithra S, Assistant Professor

Department of Computer Science and Engineering, MRK Institute of Technology, Kattumannarkoil, India

Senthamarai Sevi M, HOD

Department of Computer Science and Engineering, MRK Institute of Technology, Kattumannarkoil, India

Abstract-- Skin diseases are among the most common medical conditions and often require early identification to prevent serious complications. In many rural and resource-constrained areas, access to dermatologists is limited, which leads to delayed diagnosis and self-medication. This article presents an automated skin disease classification framework based on transfer learning. The proposed system employs deep convolutional neural networks, with emphasis on ResNet50, to classify skin disease images into multiple categories using dermoscopic and clinical image data. The framework contains image acquisition, preprocessing, data augmentation, deep feature extraction, fine-tuning, classification, visualization, and web-based deployment modules. A Python Flask backend, Kafka-based streaming layer, and HBase storage component support a scalable real-time prediction workflow. Experimental results from the thesis show that the Phase II ResNet50 model obtains 95.8% accuracy, 95.2% precision, 94.9% recall, and 95.0% F1-score, outperforming VGG16 and MobileNetV2 baselines. The system can serve as a decision-support tool for preliminary screening, telemedicine, and AI-assisted dermatology education.

Index Terms-- Skin disease classification, transfer learning, ResNet50, convolutional neural network, medical image analysis, Flask, Kafka, HBase, teledermatology.

I. INTRODUCTION

A. Background

The rapid growth of artificial intelligence in healthcare has opened new opportunities for automated disease screening and clinical decision support. Among visible medical disorders, skin diseases are highly suitable for image-based analysis because many symptoms appear on the surface of the body as rashes, lesions, pigmentation changes, swelling, scaling, or texture variations. Early recognition of these visual signs is important because delayed treatment can result in infection, discomfort, psychological stress, and increased medical cost.

Dermatology services are not equally accessible in all regions. Many patients in rural and low-income communities cannot visit a specialist immediately because of distance, financial limitations, or limited clinical infrastructure. As a result, common conditions such as eczema, psoriasis, fungal infections, warts, shingles, impetigo, basal cell carcinoma, and melanoma may remain untreated or may be treated with inappropriate self-medication. A digital preliminary screening tool can reduce this accessibility gap by providing fast feedback and encouraging users to seek professional care when needed.

Deep learning models have become prominent in medical image analysis because they learn discriminative features automatically from image data. Convolutional Neural Networks (CNNs) extract hierarchical patterns from pixels, beginning with edges and color gradients and progressing toward complex lesion shapes and disease-specific textures. Transfer learning further improves performance by adapting models trained on large image repositories to specialized medical datasets. This strategy reduces training time and improves performance when labeled dermatology images are limited.

The thesis converted in this work proposes a skin disease classification system using transfer learning and an integrated web application. The system includes a user interface for image upload, Flask backend services for preprocessing and prediction, Kafka for real-time streaming, HBase for scalable storage, and a CNN-based classification model. Phase II emphasizes ResNet50 because residual learning enables deeper feature extraction while reducing degradation and vanishing-gradient effects.

B. Problem Statement

The central problem addressed by this study is the automated classification of skin disease images into disease categories using a reliable, scalable, and user-

accessible system. Manual diagnosis depends on the availability and experience of dermatologists, and traditional machine learning methods require handcrafted features that may not generalize across datasets, imaging devices, lighting conditions, or skin tones.

The system must therefore accept a skin image, preprocess it, extract meaningful features, classify the condition, and report a prediction with a confidence score in near real time. It should also be structured for web deployment so that nontechnical users can upload images without knowing the underlying machine learning process.

TABLE I
PROBLEM CONTEXT AND PROPOSED RESPONSE

Challenge	Impact	Proposed Response
Limited dermatologist access	Delayed diagnosis and treatment	Web-based preliminary screening
Subjective visual assessment	Variation between observers	Deep learning-based feature extraction
Small or imbalanced datasets	Overfitting and weak generalization	Transfer learning and augmentation
Real-time data handling	Slow response in large systems	Kafka streaming and Flask APIs
Large image storage needs	Scalability issues	HBase distributed storage
Clinical trust requirement	Low adoption without transparency	Performance reports and future XAI

C. Objectives and Contributions

The primary objective is to develop a skin disease classification. The technical objective is to classify skin disease images using transfer learning with strong accuracy and balanced precision-recall performance.

The specific contributions include a modular architecture for image acquisition, preprocessing, feature extraction, model fine-tuning, prediction, reporting, and deployment; a comparative study of Phase I models such as VGG16 and MobileNetV2 with Phase II ResNet50; and the inclusion of visual results such as accuracy curves, loss curves, confusion matrices, model

summaries, and system screenshots from the source thesis.

II. LITERATURE REVIEW

A. Traditional and Machine Learning Methods

Early automated dermatology systems relied on image processing and machine learning. Researchers extracted handcrafted color, texture, and shape descriptors such as Gabor features, Local Binary Patterns, color moments, and Gray-Level Co-occurrence Matrix features. These descriptors were classified using Support Vector Machines, Random Forests, K-Nearest Neighbors, or decision trees. Such methods demonstrated the feasibility of automated diagnosis, but they remained highly dependent on preprocessing quality and feature engineering.

Traditional systems usually face difficulty when lesion boundaries are irregular, images contain background clutter, or diseases share similar visual characteristics. Since handcrafted features are designed manually, they may not capture subtle patterns across diverse skin tones, acquisition devices, and disease stages. These limitations motivated the transition from handcrafted pipelines to CNN-based representation learning.

B. Deep Learning in Dermatology

Deep learning has significantly improved skin disease image recognition. CNNs learn features directly from images, eliminating the need for manual feature design. Landmark studies using Inception, VGG, ResNet, DenseNet, EfficientNet, and MobileNet architectures showed that neural networks can reach dermatologist-level performance in some skin lesion classification tasks. Public datasets such as HAM10000 and ISIC have accelerated this progress by providing benchmark dermoscopic image collections.

The literature reviewed in the thesis covers work from 2016 to 2025 and highlights a trend toward deeper networks, ensemble models, data augmentation, fairness-aware training, explainable artificial intelligence, and self-supervised learning. It also shows that transfer learning is practical for medical image classification because it reuses visual patterns learned from large general-purpose datasets and adapts them to dermatological images.

TABLE II SUMMARY OF RELATED WORK IN SKIN DISEASE CLASSIFICATION

Study Direction	Typical Technique	Key Insight
Handcrafted feature learning	Gabor, LBP, color descriptors + SVM/RF	Useful but sensitive to lighting and manual design
Large-scale CNN diagnosis	Inception and deep CNN classifiers	Deep features can match expert-level diagnosis in selected tasks
Benchmark datasets	HAM10000, ISIC	Dataset diversity and augmentation improve generalization
Transfer learning	VGG16, ResNet50, InceptionV3	Pre-trained networks adapt well to medical images
Ensemble methods	VGG + DenseNet + EfficientNet fusion	Model fusion improves robustness and accuracy
Explainable AI	Activation maps and saliency methods	Transparency supports clinical acceptance
Self-supervised learning	Contrastive pretraining	Reduces dependence on labeled data

C. Research Gap

Despite strong progress, several gaps remain. Many systems are trained on limited datasets that may not represent all skin tones, ages, or imaging conditions. Some models achieve high accuracy but are difficult to deploy in real-time environments because of computational cost or lack of integration with web applications. Others provide classification results but lack scalable storage and streaming components required for production-level workflows.

The thesis addresses these issues by combining transfer learning with an end-to-end application architecture. ResNet50 improves classification strength, while Flask, Kafka, and HBase provide a path for scalable deployment and real-time operation. The article emphasizes both model performance and practical

system design, which is useful for M.E. project implementation.

III. SYSTEM ANALYSIS

A. Existing System

Existing systems for skin disease diagnosis largely depend on dermatologists or conventional computer-aided methods. Manual diagnosis is accurate when performed by trained specialists, but it is time-consuming and not always accessible. Traditional machine learning systems use manually extracted texture, color, and shape features. These approaches often require careful segmentation and may fail when lesions are visually complex or images are collected under uncontrolled conditions.

In resource-limited environments, patients may postpone medical consultation because of cost or distance. This increases the importance of a preliminary screening tool that can guide users toward timely care. Such a tool should not replace medical professionals, but it can act as a first-level decision-support mechanism.

B. Proposed System

The proposed system is an automated transfer learning-based skin disease classification framework. It uses pre-trained CNN models such as VGG16, MobileNetV2, and ResNet50, with ResNet50 selected as the improved Phase II model. The system follows a complete workflow from dataset preparation to real-time prediction.

The input image is resized to 224×224 pixels, normalized, and optionally augmented. ResNet50 extracts high-level features through residual blocks. A classifier with fully connected layers and softmax activation generates the predicted class. The prediction is returned through a Flask backend and can be displayed in the web interface. Kafka can publish and consume image-prediction messages, and HBase stores disease-related records or streaming outputs.

TABLE III COMPARISON OF EXISTING AND PROPOSED SYSTEMS

Aspect	Existing System	Proposed System
Diagnosis method	Manual examination or handcrafted features	Transfer learning with ResNet50
Feature extraction	Texture, color, shape descriptors	Automatic deep feature learning
Scalability	Limited for	Kafka and HBase

	large datasets	supported workflow
Accuracy	Varies with expert or feature quality	95.8% accuracy in Phase II results
Deployment	Often offline or lab-based	Web-based upload and prediction
User access	Requires clinical visit	Preliminary remote screening

IV. PROPOSED METHODOLOGY

A. Architecture Overview

The Phase II architecture uses a modular workflow that separates user interaction, preprocessing, model serving, streaming, and storage. The user uploads a skin image through a client interface. The Flask backend receives the image and sends it to preprocessing. The preprocessed tensor is passed to the trained CNN model, which returns the prediction. Kafka publishes prediction data for streaming, and HBase stores the output for future reference and analysis.

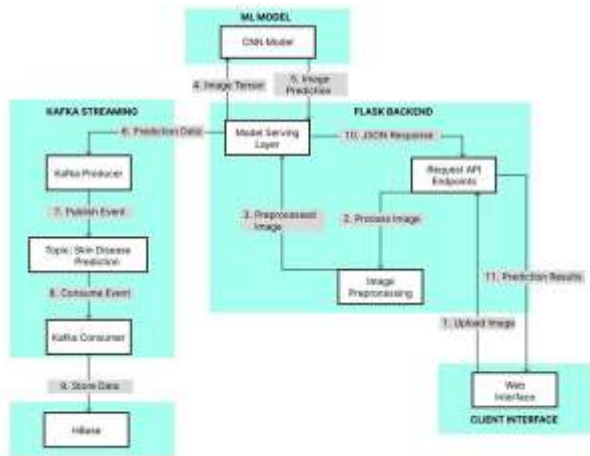


Fig. 1. System architecture showing client interface, Flask backend, CNN model serving layer, Kafka streaming, and HBase storage.

This architecture is appropriate for an intelligent healthcare prototype because it separates concerns. The model can be updated without changing the user interface, and the storage layer can be scaled independently. Kafka allows asynchronous processing when multiple prediction events occur, while Flask provides a lightweight API layer for web interaction.

B. Dataset Acquisition and Management

The dataset used in the thesis is described as dermoscopic and clinical skin disease image data collected from standard repositories such as HAM10000, ISIC, DermNet, and Kaggle-based skin disease datasets.

Disease categories include melanoma, eczema, psoriasis, benign lesions, basal cell carcinoma, atopic dermatitis, fungal infections, viral infections, shingles, impetigo, cellulitis, and related conditions.

Dataset management involves organizing images into disease-specific directories, checking class labels, splitting the dataset into training, validation, and testing subsets, and maintaining metadata for reproducible experiments. A typical split used in the thesis is 70% for training, 15% for validation, and 15% for testing. Such separation ensures that performance is evaluated on unseen data.

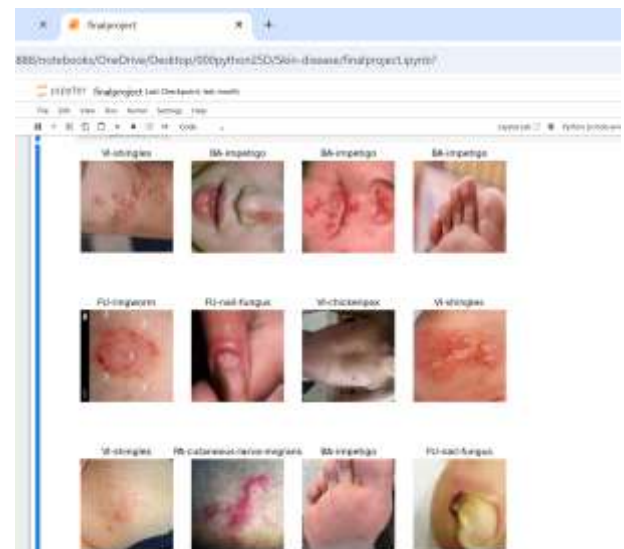


Fig. 2. Sample skin disease dataset view showing multiple disease categories used for model training and testing.

TABLE IV DATASET PREPARATION STRATEGY

Stage	Description	Purpose
Collection	Gather labeled skin disease images from public repositories	Build training and testing corpus
Organization	Arrange images by disease category	Maintain clean labels
Splitting	70% training, 15% validation, 15% testing	Evaluate generalization
Resizing	Convert images to 224×224 pixels	Match ResNet50 input dimensions
Normalization	Scale pixel values to stable ranges	Improve learning stability
Augmentation	Apply rotation, flipping, zooming,	Reduce overfitting

	brightness changes	
--	--------------------	--

C. Preprocessing and Augmentation

Preprocessing converts raw images into a consistent form suitable for CNN input. Images may differ in resolution, lighting, background, focus, and skin area coverage. Resizing standardizes dimensions, normalization stabilizes gradient updates, and noise reduction improves the clarity of lesion features. Augmentation creates additional training variations and helps the model handle real-world diversity.

Common augmentation operations include horizontal flipping, rotation, zooming, translation, brightness adjustment, and minor scaling. These transformations are medically acceptable when they preserve lesion identity. Augmentation is especially valuable when certain disease classes contain fewer examples than others.

D. Transfer Learning with ResNet50

ResNet50 is a deep residual neural network composed of convolutional layers and residual blocks. Its skip connections allow gradients to flow through deep layers more effectively and reduce degradation as network depth increases. Instead of learning a direct mapping, a residual block learns the difference between the input and the desired output, enabling deeper feature learning.

In transfer learning, ResNet50 is initialized with ImageNet weights. Early layers that capture generic edges and textures are retained, while later layers are adapted to skin disease classes. The final classification head is customized for the number of disease categories. Fine-tuning allows the model to learn domain-specific features such as lesion boundaries, pigmentation, scaling, and inflammatory patterns.

$$y = \text{softmax}(W \cdot \text{ResNet50}(x) + b), \quad x \in R^{224 \times 224 \times 3}. \quad (1)$$

$$H(x) = F(x) + x, \quad \text{where } F(x) \text{ is the learned residual mapping.} \quad (2)$$

TABLE V ROLE OF TRANSFER LEARNING LAYERS

Layer Group	Function	Treatment During Fine-Tuning
Early convolution layers	Detect edges, color gradients, simple textures	Often frozen to preserve generic features
Middle residual	Capture skin texture and lesion	Partially fine-tuned

blocks	patterns	
Deep residual blocks	Learn disease-specific structures	Fine-tuned with lower learning rate
Dense classifier	Map deep features to disease classes	Trained for target dataset

E. Classification Pipeline

The model computes a probability score for each skin disease class. The class with the highest probability is selected as the prediction. The result is presented with a confidence value so that users and healthcare personnel can understand the strength of the prediction. Because the output is preliminary, the interface should recommend professional consultation rather than replacing medical diagnosis.

Layer (type)	Output Shape	Param #
sequential (Sequential)	[None, 224, 224, 3]	0
conv2d (Conv2D)	[None, 222, 222, 32]	896
max_pooling2d (MaxPooling2D)	[None, 111, 111, 32]	0
conv2d_1 (Conv2D)	[None, 109, 109, 64]	38,496
max_pooling2d_1 (MaxPooling2D)	[None, 54, 54, 64]	0
conv2d_2 (Conv2D)	[None, 52, 52, 64]	36,928
max_pooling2d_2 (MaxPooling2D)	[None, 26, 26, 64]	0
conv2d_3 (Conv2D)	[None, 24, 24, 64]	36,928
max_pooling2d_3 (MaxPooling2D)	[None, 12, 12, 64]	0
conv2d_4 (Conv2D)	[None, 10, 10, 64]	36,928
max_pooling2d_4 (MaxPooling2D)	[None, 5, 5, 64]	0
conv2d_5 (Conv2D)	[None, 3, 3, 64]	36,928
max_pooling2d_5 (MaxPooling2D)	[None, 1, 1, 64]	0
Flatten (Flatten)	[None, 64]	0
dense (Dense)	[None, 64]	4,160
dense_1 (Dense)	[None, 7]	455

Fig. 3. Model summary showing CNN layers, max-pooling operations, dense layers, and parameter counts in a skin disease classification network.

TABLE VI ALGORITHMIC FLOW OF THE PROPOSED SYSTEM

Step	Operation
1	Acquire or upload skin image through the user interface
2	Resize, normalize, and preprocess the image
3	Apply data augmentation during training to improve robustness
4	Extract deep features using fine-tuned ResNet50
5	Classify the image using the softmax output layer
6	Return predicted disease class and confidence score
7	Publish prediction event through Kafka

	and store records in HBase
8	Display visual result and performance report to the user

V. SYSTEM DESIGN AND IMPLEMENTATION

A. Implementation Environment

The implementation environment combines machine learning, web development, streaming, and storage technologies. Python is used for backend logic and model development. TensorFlow/Keras supports CNN model construction and training. OpenCV supports image preprocessing. Flask provides a web server for handling upload and prediction requests. Kafka enables real-time message streaming, while HBase provides distributed storage for prediction data and disease-related records.

The front-end interface can be implemented with HTML, CSS, and JavaScript. The interface allows users to upload an image and view the predicted disease category. In a research prototype, the same workflow may also be tested using Jupyter Notebook for model training and visualization.

TABLE VII SOFTWARE AND HARDWARE CONFIGURATION

Component	Specification
Programming language	Python
Deep learning framework	TensorFlow / Keras
Image processing library	OpenCV
Web framework	Flask
Streaming platform	Apache Kafka
Storage layer	HBase
Frontend	HTML, CSS, JavaScript
Input image size	224 × 224 × 3
Recommended hardware	GPU or high-performance CPU with adequate RAM

B. Module Description

Data Acquisition and Dataset Management Module:

This module collects and organizes labeled images from standard datasets. It arranges images into disease-specific folders and performs training-validation-testing splitting. Proper dataset management reduces label confusion and supports reproducible experiments.

Image Preprocessing and Augmentation Module:

This module prepares images for the CNN. It resizes images, normalizes pixel intensity, removes noise when needed, and applies augmentation such as rotation, flipping, zooming, and brightness adjustment. These steps improve robustness and reduce overfitting.

Feature Extraction Module: This is the core learning module. ResNet50 extracts hierarchical image features through deep residual blocks. Low-level layers capture edges and color transitions, while deeper layers capture lesion patterns and disease-specific structures.

Transfer Learning and Fine-Tuning Module: This module adapts the pre-trained model to dermatological images. Earlier layers can be frozen, while later layers and classifier layers are fine-tuned to learn disease-specific characteristics.

Training and Optimization Module: The model is trained using forward propagation, loss calculation, backpropagation, and weight updates. The Adam optimizer, dropout, early stopping, and learning-rate tuning can improve performance and reduce overfitting.

Evaluation and Validation Module: This module computes accuracy, precision, recall, F1-score, confusion matrix values, and loss curves. These results provide quantitative evidence for the reliability of the system.

Prediction and Decision Support Module: The trained model receives a new image and returns a predicted disease class with confidence score. The result supports preliminary screening and can guide users to seek medical consultation.

Visualization and Reporting Module: This module displays accuracy graphs, loss curves, confusion matrices, sample predictions, and generated reports. These outputs are important for academic documentation and clinical interpretation.

Deployment and User Interface Module: This module exposes the trained model through a web interface. Users upload images and view predictions. The interface should be simple, responsive, and clear enough for nontechnical users.

C. Kafka and HBase Integration

Kafka is used to stream prediction events. When a user uploads an image and the model produces a prediction, the event can be published to a Kafka topic. A Kafka consumer can read the event, extract image or prediction metadata, and forward it to storage or visualization modules. Base64 encoding may be used to transmit image data through a text-based message pipeline.

HBase is used for scalable storage. It can store disease labels, prediction records, timestamps, image identifiers, confidence scores, and user/session information. The source thesis includes HBase shell output showing disease-related table names and confirms the storage layer concept.

```
Epoch 1/100: 37s 3s/step - accuracy: 0.3088 - loss: 1.0106 - val_accuracy: 0.1875 - val_loss: 1.9511
Epoch 2/100: 26s 3s/step - accuracy: 0.3097 - loss: 1.0014 - val_accuracy: 0.1875 - val_loss: 1.9201
Epoch 3/100: 26s 3s/step - accuracy: 0.3081 - loss: 1.0104 - val_accuracy: 0.1875 - val_loss: 1.8908
Epoch 4/100: 27s 3s/step - accuracy: 0.2708 - loss: 1.0073 - val_accuracy: 0.1812 - val_loss: 1.7986
Epoch 5/100: 26s 3s/step - accuracy: 0.2868 - loss: 1.0029 - val_accuracy: 0.2068 - val_loss: 1.6959
Epoch 6/100: 26s 3s/step - accuracy: 0.3438 - loss: 1.0121 - val_accuracy: 0.2108 - val_loss: 1.7407
Epoch 7/100: 27s 3s/step - accuracy: 0.3519 - loss: 1.0057 - val_accuracy: 0.4962 - val_loss: 1.5377
Epoch 8/100: 27s 3s/step - accuracy: 0.1865 - loss: 1.0131 - val_accuracy: 0.4375 - val_loss: 1.4398
Epoch 9/100: 26s 3s/step - accuracy: 0.4416 - loss: 1.4010 - val_accuracy: 0.4962 - val_loss: 1.4711
Epoch 10/100: 27s 3s/step - accuracy: 0.3181 - loss: 1.4736 - val_accuracy: 0.4219 - val_loss: 1.4393
Epoch 11/100: 27s 3s/step - accuracy: 0.4845 - loss: 1.0708 - val_accuracy: 0.4913 - val_loss: 1.4395
Epoch 12/100: 26s 3s/step - accuracy: 0.4951 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 13/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 14/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 15/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 16/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 17/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 18/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 19/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
Epoch 20/100: 26s 3s/step - accuracy: 0.4957 - loss: 1.0541 - val_accuracy: 0.4608 - val_loss: 1.2106
```

Fig. 4. Training log output from the model execution environment, showing iterative learning progress across epochs.

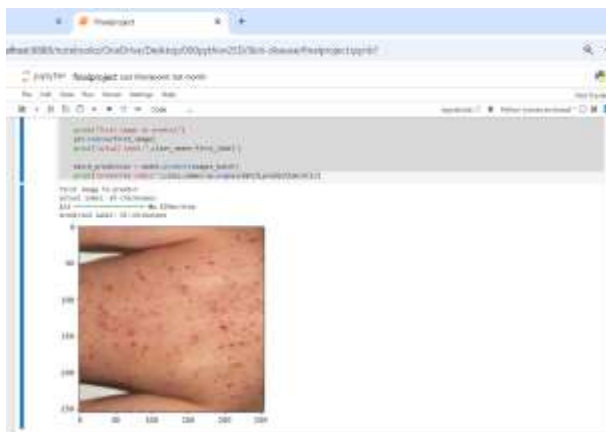


Fig. 5. Example prediction screen in Jupyter Notebook showing actual and predicted labels for a skin disease image.

VI. RESULTS AND DISCUSSION

A. Experimental Setup

The Phase II experiments evaluate ResNet50 on skin disease image data. The dataset is divided into training, validation, and testing partitions. All images are resized to 224×224 pixels and normalized. Augmentation is used to increase training diversity. The ResNet50 model is initialized with pre-trained ImageNet weights and then fine-tuned for disease classification.

Model performance is assessed using standard medical image classification metrics. Accuracy measures overall correctness, precision measures the reliability of positive predictions, recall measures the ability to identify actual disease cases, and F1-score balances

precision and recall. Confusion matrix analysis helps identify class-specific strengths and misclassification patterns.

TABLE VIII PERFORMANCE METRICS OF PHASE II RESNET50 MODEL

Metric	Value
Accuracy	95.8%
Precision	95.2%
Recall	94.9%
F1-Score	95.0%



Fig. 6. Bar-chart representation of ResNet50 performance metrics reported in the Phase II results.

B. Training and Validation Analysis

The training and validation accuracy curves show that the ResNet50 model learns progressively across epochs. Training accuracy increases steadily, while validation accuracy remains close to the training curve. This indicates that the model is not simply memorizing training images but is learning features that generalize to unseen samples.

The loss curve decreases during training, showing that the model reduces prediction error over time. A close relationship between training and validation loss suggests stable optimization and reduced overfitting. Residual connections and augmentation contribute to this behavior by improving gradient flow and data diversity.

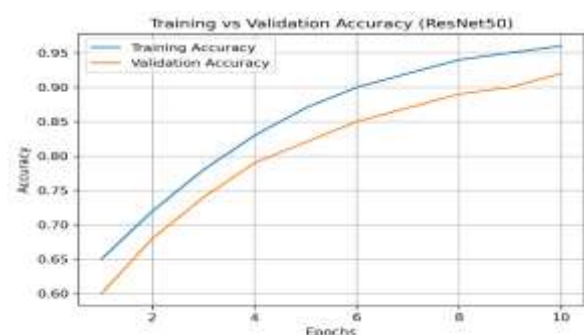


Fig. 7. Training versus validation accuracy curve for ResNet50, showing steady learning and improved generalization.

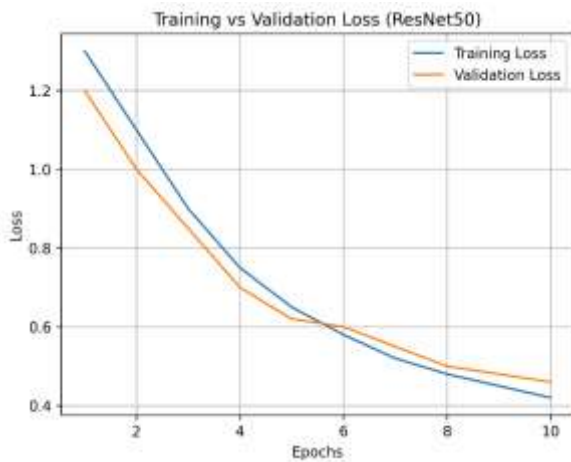


Fig. 8. Training versus validation loss curve for ResNet50, showing decreasing error across epochs.

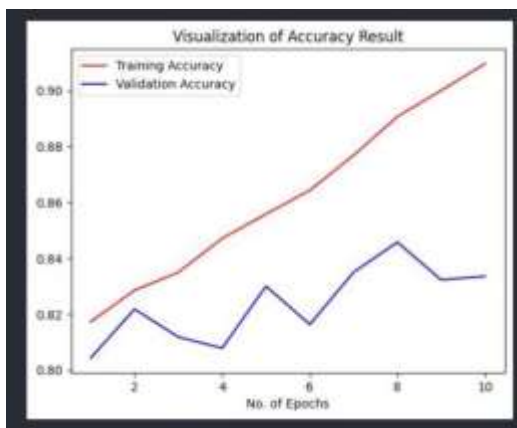


Fig. 9. Additional accuracy result visualization from the source implementation, showing training and validation behavior.

C. Confusion Matrix Analysis

The confusion matrix provides class-wise insight into model predictions. High diagonal values indicate correctly classified samples, while off-diagonal values show misclassifications. In the source result, the ResNet50 confusion matrix shows strong classification across four classes, with most values concentrated along the diagonal.

A second confusion matrix in the thesis represents a six-class CNN prediction scenario. It shows that some classes are classified very strongly, while visually similar disease categories can be confused. Such misclassifications are expected in medical imaging because multiple skin conditions may share visual features such as redness, scaling, pigmentation, or lesion boundaries.

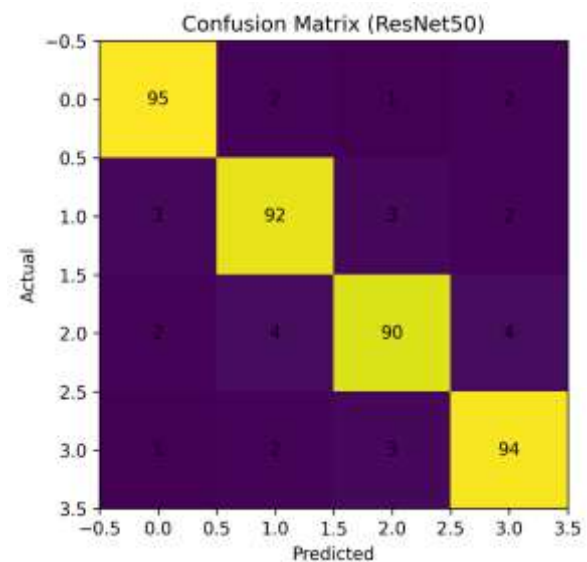


Fig. 10. Confusion matrix of the Phase II ResNet50 model with high diagonal values across disease classes.

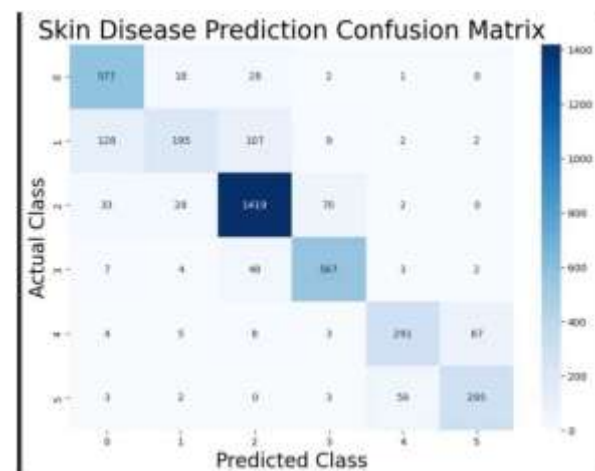


Fig. 11. Six-class skin disease prediction confusion matrix showing true positives and challenging class-pair confusions.

D. Comparative Analysis

TABLE IX MODEL COMPARISON BETWEEN PHASE I AND PHASE II

Model	Accuracy (%)	Strength	Limitation
VGG16	91.2	Strong feature extraction	High computation
MobileNetV2	92.4	Lightweight and fast	Lower detail capture
ResNet50	95.8	High accuracy and deep residual learning	Moderate complexity

The comparative results demonstrate that ResNet50 outperforms Phase I models. VGG16 provides strong

feature extraction but requires more computation. MobileNetV2 is lightweight and efficient but may capture fewer fine-grained lesion details. ResNet50 offers a balanced solution with deeper representation, residual learning, and improved accuracy.

The improvement from 91.2% and 92.4% to 95.8% is significant for prototype because it shows how model selection affects medical image classification. It also demonstrates the value of using deeper transfer learning when sufficient preprocessing and augmentation are applied.

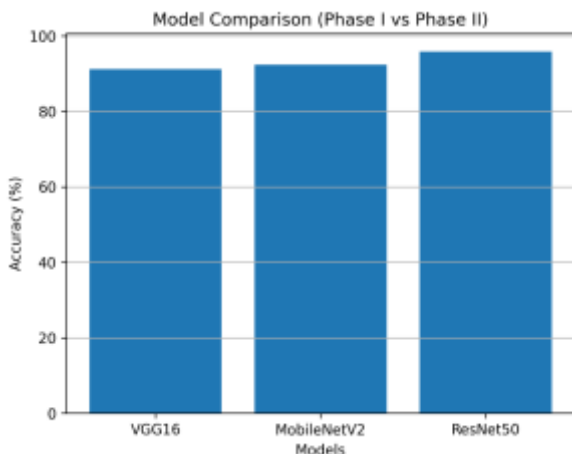


Fig. 12. Model comparison graph showing that ResNet50 outperforms VGG16 and MobileNetV2.

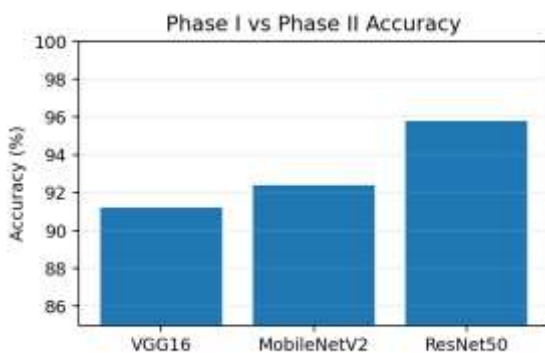


Fig. 13. Supplemental comparison chart generated from the reported Phase I and Phase II accuracy values.

E. Output and System Demonstration

The output screens in the thesis show how the trained system can be used for sample predictions. The user can view actual and predicted labels along with confidence values. This supports result verification and provides a simple demonstration of the prediction module.

The HBase output screen shows how tables can be listed in the storage environment. Disease-related table names indicate that the backend can be organized to store disease classes, prediction records, and application metadata. In a production system, additional privacy and security controls should be applied to protect patient data.

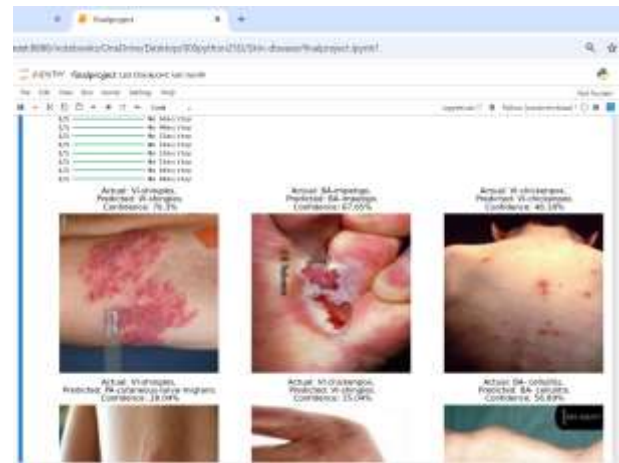


Fig. 14. Prediction output screen showing actual and predicted skin disease classes with confidence scores.

VII. DISCUSSION

A. Technical Interpretation

The results confirm that transfer learning is effective for skin disease classification. ResNet50 learns discriminative features that distinguish disease categories even when lesion appearance is visually complex. Residual connections improve training stability and allow deeper layers to represent high-level structures. The close relationship between training and validation curves suggests that augmentation and model design reduce overfitting.

The system architecture is also important. A model alone is not enough for a practical decision-support tool. The thesis integrates the classifier into a pipeline that includes web interaction, streaming, storage, and reporting. This makes the system more suitable for real-world application scenarios, such as telemedicine portals, dermatology education, and preliminary rural screening.

B. Medical and Ethical Considerations

Automated skin disease classifiers should be presented as preliminary tools rather than replacements for doctors. The output may help users understand whether a condition appears similar to known disease classes, but final diagnosis requires clinical examination and medical history. The system should therefore include a disclaimer and recommend consultation for severe, changing, painful, or persistent lesions.

Ethical deployment requires dataset diversity and fairness. If training data is dominated by certain skin tones, camera settings, or geographic populations, the model may perform less reliably on underrepresented groups. Future implementations should evaluate model performance across skin tone, age, gender, and acquisition conditions. Privacy is also essential because

skin images are sensitive medical data; storage and transmission must use secure mechanisms.

TABLE X CLINICAL DEPLOYMENT CONSIDERATIONS

Consideration	Requirement
Decision-support role	Clearly state that predictions are preliminary and not final diagnosis
Patient privacy	Use secure storage, access control, and encrypted communication
Dataset fairness	Evaluate performance across skin tones and populations
Model transparency	Add heatmaps or explainable AI methods in future versions
Clinical validation	Test with expert-labeled clinical data before deployment
User safety	Recommend dermatologist consultation for serious symptoms

C. Limitations

The study depends on the quality and diversity of the dataset. If images are blurred, poorly illuminated, or mislabeled, model performance may decline. The system also focuses on selected disease categories and may not recognize conditions outside the training set. Another limitation is that visual images alone cannot capture full clinical context such as patient age, symptoms, lesion history, allergies, or previous treatment.

Computational requirements may also affect deployment. ResNet50 is more powerful than MobileNetV2 but less lightweight. For mobile or embedded deployment, pruning, quantization, knowledge distillation, or lightweight architectures may be required. In addition, real-time storage and streaming components must be secured and optimized before clinical use.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

This article on skin disease classification using transfer learning. The proposed system combines image preprocessing, CNN-based feature extraction, ResNet50 transfer learning, Flask-based web deployment, Kafka streaming, and HBase storage. The design supports both academic experimentation and practical decision-support applications.

The Phase II ResNet50 model achieved 95.8% accuracy, 95.2% precision, 94.9% recall, and 95.0% F1-score, outperforming VGG16 and MobileNetV2 baseline models. Accuracy graphs, loss curves, confusion matrices, system architecture, model summaries, and output screens demonstrate that the system learns meaningful features and can classify multiple skin disease images with high reliability.

The system should be viewed as a preliminary screening aid rather than a replacement for professional medical diagnosis. With careful validation, privacy protection, and integration of explainable AI, it can become a useful component of telemedicine and intelligent healthcare systems.

B. Future Work

Future work can extend the system in several directions. EfficientNet, DenseNet, Vision Transformer, and hybrid attention models can be evaluated to improve classification accuracy and interpretability. Explainable AI methods such as Grad-CAM can be added to highlight lesion regions that influenced the prediction. This would improve clinician trust and make the system more transparent.

The dataset can be expanded to include more diseases, diverse skin tones, different age groups, and images captured from multiple devices. Patient metadata such as age, symptoms, duration, and medical history can be fused with image features for stronger diagnosis support. The model can also be optimized for mobile and edge deployment using quantization and pruning.

Future deployment should include secure authentication, encrypted image transmission, privacy-preserving storage, and clinical validation with dermatologist-labeled data. These improvements can help transform the prototype into a more robust, ethical, and clinically useful AI-assisted skin health platform.

REFERENCES

- [1] "Skin diseases image dataset," Kaggle. Available: <https://www.kaggle.com/datasets/ismailpromus/skin-diseases-image-dataset/data>
- [2] "Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images," IEEE Xplore, 2019.
- [3] E. Goceri, "Analysis of Deep Networks with Residual Blocks and Different Activation Functions: Classification of Skin Diseases," 2019 Ninth International Conference on Image Processing Theory, Tools and Applications, 2019.

- [4] G. Singh, K. Guleria, and S. Sharma, "A Transfer Learning-based Pre-trained VGG16 Model for Skin Disease Classification," MysuruCon, 2023.
- [5] N. Akmalia, P. Sihombing, and Suherman, "Skin Diseases Classification Using Local Binary Pattern and Convolutional Neural Network," IEEE Xplore, 2019.
- [6] A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," Nature, 2017.
- [7] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset: A large collection of multi-source dermoscopic images," Scientific Data, 2018.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE CVPR, 2016.
- [9] J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database," Proc. IEEE CVPR, 2009.
- [10] F. Chollet, "Keras," 2015. Available: <https://keras.io>
- [11] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.
- [12] Apache Kafka Documentation. Available: <https://kafka.apache.org/documentation/>
- [13] Apache HBase Reference Guide. Available: <https://hbase.apache.org/book.html>
- [14] G. Brinker et al., "Deep learning outperformed dermatologists in melanoma image classification," European Journal of Cancer, 2019.
- [15] H. Mahbod et al., "Skin lesion classification using hybrid deep neural networks," Computerized Medical Imaging and Graphics, 2019.