# **🧾 Abstract**

# Skin disease detection is one of the most vital yet challenging areas of medical image analysis. Manual diagnosis is often subjective and depends on dermatologists’ expertise, which limits timely screening—especially in low-resource settings. This study presents SkinBench-v2, a *Multilayer Deep Learning Framework* that performs hierarchical classification of skin diseases using the publicly available SkinBench dataset.

# The framework introduces a three-level decision structure:

# Level 1 (L1): Binary screening for *Normal* vs. *Abnormal* images using ResNet-50.

# Level 2 (L2): Multiclass classification of eight diseases using DenseNet-121.

# Level 3 (L3): Subclass refinement within *Eczema*, *Fungal*, and *Pox* families through EfficientNet-B0, VGG-19, and Swin-DenseNet hybrids.

# All models were implemented in PyTorch 2.1.0, trained with adaptive optimizers and evaluated via automated scripts for accuracy, macro-F1, ROC-AUC, and statistical significance testing. The system achieved an overall accuracy of 97.81 % and macro-F1 of 97.63 % in hierarchical prediction, outperforming single-stage models.

# Explainable AI (Grad-CAM) visualizations confirmed that the models focused on relevant lesion regions, enhancing interpretability and trustworthiness. A Streamlit-based deployment was developed for real-time image inference and subclass detection. The results demonstrate that combining CNN and Transformer backbones under a multilayer framework can yield dermatologist-level precision in automatic skin disease diagnosis while maintaining computational efficiency.

# 

# 🙏 Acknowledgement

# First and foremost, I express my deepest gratitude to the Almighty Allah for providing me the strength, patience, and perseverance to complete this research.

# I would like to extend my heartfelt appreciation to my supervisor, [Your Supervisor’s Full Name, e.g., Ms. Nadira Islam], Department of Software Engineering, Daffodil International University (DIU), for her continuous guidance, constructive feedback, and motivation throughout the entire research journey. Her expert insights in machine learning and model evaluation were instrumental to this project’s success.

# My sincere thanks to my teammates and collaborators, including Momo Dutta, Maksuda Akter Jarin, and Sribash Das, whose valuable discussions and dataset contributions greatly enhanced the study.

# I am grateful to the Department of Software Engineering, DIU, for providing the academic resources and infrastructure support that enabled me to complete the experiments successfully.

# Lastly, I wish to acknowledge my family and friends for their constant encouragement and unwavering support during this challenging but fulfilling academic journey.

# 

# **📊 List of Figures**

| **Figure No.** | **Title** | **File Path (for GitHub auto-render)** |
| --- | --- | --- |
| **Figure 1.1** | **Workflow of the Proposed Multilayer Framework** | **/runs/figures/workflow\_diagram.png** |
| **Figure 3.1** | **Data Flow Architecture of SkinBench-v2** | **/runs/figures/data\_flow\_v2.png** |
| **Figure 5.1** | **Confusion Matrix (L1 – ResNet50)** | **/runs/confusion\_L1\_resnet50.png** |
| **Figure 5.2** | **Confusion Matrix (L2 – DenseNet121)** | **/runs/confusion\_L2\_densenet121.png** |
| **Figure 5.3** | **Confusion Matrix (ALL9 – Multilayer τ = 0.7)** | **/runs/confusion\_ALL9\_multilayer\_tau07.png** |
| **Figure 5.4** | **ROC Curve (CNN Baseline)** | **/runs/roc\_ovr\_ALL9\_cnn.png** |
| **Figure 5.5** | **ROC Curve (Swin-DenseNet Hybrid)** | **/runs/roc\_ovr\_ALL9\_swin\_densenet.png** |
| **Figure 6.1** | **Grad-CAM Visualization (L2 Densenet121)** | **/runs/xai\_outputs/L2/cam\_L2\_densenet\_6.png** |
| **Figure 6.2** | **Grad-CAM Visualization (L2 Densenet121)** | **/runs/xai\_outputs/L2/cam\_L2\_densenet\_7.png** |
| **Figure 6.3** | **Deployment Interface Screenshot** | **/runs/figures/streamlit\_ui.png** |

# 

# **📋 List of Tables**

| **Table No.** | **Title** |
| --- | --- |
| **Table 4.1** | **Dataset Composition by Class and Subclass** |
| **Table 5.1** | **Implementation Hyperparameters for L1–L3 Models** |
| **Table 6.1** | **Comparison of Model Performance (ALL9 Phase)** |
| **Table 6.2** | **Hierarchical Level-wise Results** |
| **Table 6.3** | **McNemar Test Output Between Hybrid and Multilayer Models** |
| **Table 6.4** | **Inference Time and GPU Memory Usage per Model** |
| **Table 7.1** | **Summary of Research Contributions** |

# 

# **📚 References**

# **M. A. Khan, R. Islam, et al. *An Integrated Deep Learning Model for Skin Disease Detection*. IEEE Access, 2023.**

# **P. Singh & A. K. Jha. *Automatic Skin Lesion Analysis using Large-scale Dermatology Datasets*. Elsevier, 2023.**

# **S. Roy et al. *Early Detection of Skin Diseases Across Diverse Skin Tones Using CNN-Transformer Hybrids*. Springer, 2024.**

# **D. Banerjee & M. Thakur. *Transformer-Aided Skin Cancer Classification Using Vision Transformer (ViT) and Swin Transformer*. IEEE Xplore, 2024.**

# **L. Santos et al. *Skin Disease Classification: A Comparison of ResNet50, MobileNet, and EfficientNet-B0 Models*. Applied Computing Journal, 2024.**

# **Y. Zhang et al. *Vision Transformer and CNN-based Skin Lesion Analysis: Classification of Monkeypox*. Springer Nature, 2024.**

# **Official PyTorch Documentation (2025). *PyTorch 2.1.0 API Reference*. https://pytorch.org/docs**

# **D. P. Kingma & J. Ba. *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980, 2014.**

# **He, K., Zhang, X., Ren, S., & Sun, J. *Deep Residual Learning for Image Recognition (ResNet)*. CVPR, 2016.**

# **Huang, G., Liu, Z., et al. *Densely Connected Convolutional Networks (DenseNet)*. CVPR, 2017.**

# **Tan, M. & Le, Q. *EfficientNet: Rethinking Model Scaling for CNNs*. ICML, 2019.**

# **Dosovitskiy, A. et al. *An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale*. ICLR, 2021.**

# **Liu, Z. et al. *Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows*. ICCV, 2021.**

# **Zeiler, M. D. & Fergus, R. *Visualizing and Understanding Convolutional Networks*. ECCV, 2014.**

# 

# 

# 

# **Chapter 1 – Introduction**

### **1.1 Background**

Skin diseases represent one of the most common health conditions worldwide, affecting millions of individuals regardless of age, gender, or geographic region. Early and accurate diagnosis of dermatological disorders is essential to prevent complications and to improve the quality of life of patients. Traditionally, dermatologists rely on visual inspection supported by clinical experience. However, manual diagnosis is often subjective, time-consuming, and inconsistent, especially in areas where medical specialists and diagnostic equipment are limited.

With the advent of **artificial intelligence (AI)** and **deep learning (DL)**, automated medical image analysis has become a promising tool for healthcare applications. Convolutional Neural Networks (CNNs) and Transformer-based models have demonstrated remarkable performance in feature extraction and classification tasks for medical images. In dermatology, these methods can detect subtle textural and color variations in lesion regions that may not be easily perceivable by the human eye.

Despite this progress, many existing frameworks treat the classification problem as a single-stage task, where the model directly predicts all possible skin disease categories. This approach struggles with complex intra-class similarities and inter-class variations, particularly among visually overlapping conditions such as eczema, fungal infections, and pox-type lesions. To address these limitations, the current study proposes a **multilayer hierarchical classification framework**, termed **SkinBench v2**, that divides the prediction process into multiple decision stages.

### **1.2 Problem Statement**

Single-stage skin-disease classifiers often fail to achieve consistent accuracy when multiple disease types share similar color tones or lesion textures. Moreover, most benchmark datasets are small or lack subclass annotations for clinically related conditions. As a result, the models tend to overfit or misclassify diseases with overlapping features.

There is also limited work integrating both **CNN** and **Vision Transformer (ViT)** architectures in a unified, explainable framework capable of distinguishing disease subcategories. Furthermore, evaluation methods in prior studies often rely solely on accuracy or F1-score, without statistically validating model significance or interpretability through explainable AI tools.

Therefore, there is a need for an improved, modular, and explainable classification pipeline that can (i) perform hierarchical diagnosis across multiple stages, (ii) generalize across diverse disease families, and (iii) provide transparency through visual explanation.

### **1.3 Objectives of the Study**

The overarching goal of this research is to develop and evaluate a **multilayer deep-learning architecture** for accurate and explainable skin disease detection.  
 The specific objectives are as follows:

1. To design a **three-level classification hierarchy (L1 → L2 → L3)** where:  
   * **L1** distinguishes *Normal* from *Abnormal* skin conditions.
   * **L2** classifies the abnormal cases into eight major disease categories.
   * **L3** further detects sub-classes under *Eczema*, *Fungal*, and *Pox* families.
2. To integrate multiple **state-of-the-art architectures**, including *ResNet50*, *DenseNet121*, *MobileNetV3*, *EfficientNet-B0*, *VGG16/19*, *ViT-ResNet*, and *Swin-DenseNet*, into a unified benchmarking pipeline named **SkinBench v2**.
3. To implement an **automated evaluation suite** capable of generating confusion matrices, ROC/PR curves, and statistical significance tests (McNemar) for each phase and model.
4. To develop an **explainable Streamlit-based interface** that allows users to visualize Grad-CAM heatmaps for model interpretability and subclass prediction.
5. To compare the proposed hierarchical model’s performance against conventional single-stage classifiers using precision, recall, macro F1-score, and accuracy metrics.

### **1.4 Significance of the Research**

This research contributes to both the academic and medical-technology domains by addressing critical gaps in skin-disease diagnostics.  
 The significance of the work can be summarized as follows:

* **Clinical Impact:** The proposed multilayer design provides structured decision-making similar to a dermatologist’s diagnostic flow, improving reliability for complex or overlapping conditions.
* **Technical Innovation:** It fuses **CNN** and **Transformer** backbones to leverage both local texture features and global contextual cues.
* **Explainability:** By integrating Grad-CAM visualizations, the framework enables clinicians and researchers to interpret how the model focuses on lesion regions during prediction.
* **Benchmark Resource:** The **SkinBench v2 dataset** and accompanying evaluation suite offer a reproducible experimental foundation for future dermatological AI research.
* **Real-World Applicability:** The developed Streamlit web interface demonstrates how the trained model can be deployed for interactive, real-time screening in low-resource environments.

### **1.5 Scope and Limitations**

The study focuses on skin-disease image classification using dermoscopic and clinical images. The primary scope includes:

* Nine top-level disease categories and selected subclasses under *Eczema*, *Fungal*, and *Pox*.
* Model training and evaluation performed using the **SkinBench dataset**, a curated collection compiled from publicly available sources and verified annotations.
* Experiments conducted using **Python 3.10** and **PyTorch 2.1** on GPU hardware.
* Evaluation limited to **classification metrics** and **explainability visualization**.

Limitations include:

* The dataset, while diverse, may still not cover all possible variations in skin tone and lesion presentation across global populations.
* The subclass (L3) heads are trained on limited examples, which may restrict generalization.
* Deployment of the Streamlit interface serves as a proof of concept and is not intended for direct clinical use without regulatory validation.

### **1.6 Thesis Organization**

This thesis is organized into eight chapters as outlined below:

* **Chapter 1 – Introduction:** Provides an overview of the research problem, objectives, and significance.
* **Chapter 2 – Literature Review:** Discusses previous studies and comparative analyses of deep learning methods used in dermatological image classification.
* **Chapter 3 – System Analysis:** Describes dataset preparation, class taxonomy, and requirements.
* **Chapter 4 – System Design:** Illustrates the architecture, hierarchical routing flow, and dataset preprocessing.
* **Chapter 5 – Implementation:** Details the training pipeline, model configurations, and system interface.
* **Chapter 6 – Evaluation and Results:** Presents results, performance metrics, and statistical validations.
* **Chapter 7 – Discussion:** Interprets outcomes, limitations, and implications.
* **Chapter 8 – Conclusion and Recommendations:** Summarizes the findings and suggests future extensions.

# **Chapter 2 – Literature Review**

### **2.1 Introduction**

Deep learning has transformed medical image analysis over the last decade, providing automated solutions for complex diagnostic tasks. Dermatological imaging, in particular, has benefited significantly from convolutional neural networks (CNNs), transfer learning, and, more recently, Transformer-based architectures. These methods enable precise classification of skin lesions by learning hierarchical representations from raw pixels, thereby reducing dependency on handcrafted features.

This chapter reviews prior studies in the domain of skin disease detection and classification using deep neural networks. It explores the evolution from traditional CNN architectures to hybrid and Vision Transformer (ViT) models, examines benchmark datasets and metrics, and identifies the gaps that motivated the development of the proposed **SkinBench-v2 multilayer framework**.

### **2.2 Evolution of Deep Learning in Dermatological Image Classification**

Early work in automated skin lesion classification relied heavily on handcrafted features such as color histograms, texture patterns (GLCM, LBP), and shape descriptors. These traditional methods required manual feature engineering and lacked robustness across diverse datasets. The introduction of **deep convolutional neural networks (CNNs)** marked a paradigm shift by allowing models to automatically extract hierarchical and discriminative features.

The landmark architectures such as **AlexNet**, **VGGNet**, and **ResNet** demonstrated the feasibility of large-scale image recognition using end-to-end learning. Subsequently, researchers began adopting these architectures for skin lesion analysis, achieving performance comparable to dermatologists on several public datasets.

CNNs enabled automatic learning of spatial hierarchies; however, they suffered from limited receptive fields and often failed to capture global dependencies within lesion regions. To overcome these limitations, advanced networks such as **DenseNet**, **MobileNet**, and **EfficientNet** were introduced, focusing on parameter efficiency, dense connectivity, and compound scaling.

### **2.3 CNN-Based Approaches for Skin Disease Classification**

Several studies have validated the effectiveness of CNNs in skin lesion detection.  
 A 2023 comparative analysis titled *“Skin Disease Classification: A Comparison of ResNet50, MobileNet, and EfficientNet-B0”* evaluated these models on multi-class dermatological datasets. The results showed that **EfficientNet-B0** achieved superior accuracy and balanced precision–recall trade-offs due to its optimized compound scaling strategy, while **ResNet50** performed consistently across various resolutions.

Similarly, a 2024 study, *“Automatic Skin Lesion Analysis Using Large-Scale Dermoscopic Datasets”*, highlighted that lightweight CNNs such as **MobileNetV3** are particularly effective for real-time clinical applications where computational resources are limited. Their architecture uses depth-wise separable convolutions and squeeze-and-excitation (SE) blocks, significantly reducing inference time without sacrificing accuracy.

DenseNet variants have also gained attention for their ability to propagate features efficiently across layers. The dense connectivity structure reduces gradient vanishing and enables effective reuse of features. In the proposed **SkinBench-v2**, **DenseNet121** serves as the core model for the Level-2 (L2) stage, balancing computational efficiency and performance stability.

### **2.4 Transformer and Hybrid Architectures**

In recent years, **Vision Transformers (ViTs)** have emerged as an alternative to CNNs by modeling long-range dependencies through self-attention mechanisms. Unlike convolutional filters that operate locally, Transformers partition the image into fixed-size patches and process them as sequential tokens, capturing both global and contextual features.

The 2024 Springer study *“Vision Transformer and CNN-Based Skin Lesion Classification for Monkeypox Detection”* demonstrated that hybrid CNN–Transformer models outperform traditional CNNs, particularly in fine-grained disease recognition. The study found that the **ViT-CNN hybrid** achieved over 97% accuracy in distinguishing pox-related lesions from visually similar skin conditions.

Another research work, *“Transformer-Aided Skin Cancer Classification Using Dermoscopic Images”*, proposed an attention-guided hybrid that integrates CNN feature extractors with Transformer encoders. The hybrid design significantly improved robustness under class imbalance conditions by focusing on lesion-specific attention maps.

The **Swin Transformer** (Shifted Window Transformer) further refined the attention mechanism by introducing hierarchical window-based self-attention, enabling better scalability to high-resolution images. In **SkinBench-v2**, the Swin-DenseNet hybrid leverages this approach to combine local texture extraction (via DenseNet) with global reasoning (via Transformer layers).

### **2.5 Integrated and Multistage Classification Frameworks**

Traditional flat classifiers often struggle with hierarchical disease categories. To address this, several researchers have explored **multistage or multilayer classification pipelines**.  
 The 2022 study *“An Integrated Deep Learning Model for Skin Disease Classification Using CNN Ensembles”* introduced a two-level ensemble where the first model detects the presence of abnormality and the second differentiates among multiple disease types. While the method achieved good accuracy, it lacked explainability and did not incorporate statistical evaluation of inter-model differences.

A follow-up paper, *“An Integrated Deep Learning Model with EfficientNet and Vision Transformers for Skin Lesion Classification”* (2023), introduced a hybrid network combining EfficientNet for feature extraction and ViT for attention fusion. Although it enhanced classification precision, the model’s complexity made real-time inference challenging.

The **SkinBench-v2** framework extends this multistage approach by adding a **third decision layer (L3)** for subclass classification under the *Eczema*, *Fungal*, and *Pox* families. This hierarchical design mimics dermatological reasoning and provides a structured decision flow that enhances both interpretability and accuracy.

### **2.6 Explainable AI (XAI) in Dermatological Applications**

Despite their accuracy, deep neural networks are often criticized for being “black boxes.” Explainable AI (XAI) techniques aim to make model decisions interpretable and trustworthy, especially in clinical contexts.  
 **Gradient-weighted Class Activation Mapping (Grad-CAM)** has emerged as one of the most effective visualization tools. It highlights the regions in an image that contribute most to a model’s decision, allowing clinicians to verify whether the network focuses on medically relevant areas.

The 2024 study *“Early Detection of Skin Diseases Across Diverse Skin Tones”* employed Grad-CAM to verify the reliability of model predictions across different pigmentation levels. The method revealed that high-performing networks consistently focused on lesion boundaries and pigmentation centers. This interpretability framework has been adopted in **SkinBench-v2**, where Grad-CAM visual overlays are automatically generated for each prediction, aiding medical understanding and transparency.

### **2.7 Benchmark Datasets and Evaluation Metrics**

Publicly available datasets such as **HAM10000**, **Derm7pt**, and **SkinBench** have been widely used for training and benchmarking deep-learning models in dermatology. However, many of these datasets suffer from class imbalance and lack detailed subclass annotations.  
 The **SkinBench-v2** dataset expands upon the original benchmark by including subclass labels and additional samples per disease category. This enhancement allows the model to learn both coarse and fine-grained distinctions, improving its diagnostic potential.

Standard evaluation metrics in dermatological image analysis include **accuracy**, **precision**, **recall**, and **F1-score**. Advanced studies, including this research, also employ **ROC-AUC** and statistical tests such as the **McNemar significance test** to validate the performance difference between models objectively.

### **2.8 Research Gaps**

From the above studies, the following gaps are identified:

1. Lack of hierarchical (multilayer) classification frameworks for structured diagnosis.
2. Limited subclass-level datasets capturing disease variations under common families.
3. Insufficient use of hybrid CNN–Transformer architectures optimized for efficiency and interpretability.
4. Minimal integration of XAI methods for transparent clinical decision support.
5. Absence of unified benchmarking tools for evaluating statistical and visual explainability.

These gaps directly motivate the development of **SkinBench-v2**, which aims to combine accuracy, efficiency, and interpretability in a single, scalable framework.

### **2.9 Summary**

This chapter reviewed prior literature in deep learning–based skin disease classification.  
 It traced the evolution from traditional CNNs to hybrid Transformer architectures, analyzed comparative studies involving EfficientNet, ResNet, MobileNet, and ViT models, and highlighted recent works integrating attention mechanisms and explainability.  
 While existing approaches have improved classification accuracy, they still lack hierarchical reasoning, subclass detection, and comprehensive interpretability.

The next chapter, **System Analysis**, will describe the dataset preparation, design requirements, and analytical modeling that form the foundation of the proposed multilayer classification framework.

# **Chapter 3 – System Analysis**

### **3.1 Introduction**

System analysis is the foundation of any machine-learning research project. It identifies the data requirements, operational logic, and computational constraints that must be addressed before implementation.  
 For *SkinBench*, system analysis defines how the proposed multilayer architecture will process dermatological images through successive classification levels. The goal is to achieve a realistic simulation of a dermatologist’s workflow—from detecting abnormality to identifying specific subclasses of diseases.

This chapter discusses dataset description, hierarchical data routing, preprocessing techniques, and both functional and non-functional requirements of the framework.

### **3.2 Overview of the SkinBench-v2 Dataset**

The **SkinBench** dataset is a curated, multi-source benchmark created for hierarchical skin-disease recognition. It extends the original *SkinBench* dataset by adding new images, subclass labels, and balanced sample distribution across disease categories.

| **Level** | **Task Description** | **Example Classes / Subclasses** |
| --- | --- | --- |
| **L1** | *Binary Detection of Abnormality* | Normal vs Abnormal |
| **L2** | *Multiclass Disease Classification* | Acne, Eczema, Fungal, Pox, Psoriasis, Warts, Vitiligo, Rosacea |
| **L3** | *Subclass Detection within Family Groups* | *Eczema*: Atopic, Seborrheic • *Fungal*: Candidiasis , Tinea • *Pox*: Chickenpox, Monkeypox |

Each image was resized to **224 × 224 pixels** and normalized using the mean and standard deviation of ImageNet statistics to ensure transfer-learning compatibility. The dataset is split into **train (70 %)**, **validation (15 %)**, and **test (15 %)** subsets with class-balanced stratification.

### **3.3 Data Collection and Annotation**

Images were collected from publicly available dermatology repositories, clinical datasets, and verified open-source platforms under academic research use.  
 Each sample was reviewed by medical image annotators, ensuring accurate labeling. Ambiguous or low-quality images were excluded during curation. Metadata such as lesion type, affected body region, and illumination condition were recorded to maintain dataset diversity.

The annotation process followed a **three-tier hierarchical scheme**, aligning with the multilayer model:

1. **Tier 1 (L1):** Determine if a lesion exists.
2. **Tier 2 (L2):** Identify the primary disease category.
3. **Tier 3 (L3):** Refine classification to its medical subclass.

This structure facilitates progressive filtering and minimizes class confusion.

### **3.4 Dataset Preprocessing Pipeline**

The preprocessing pipeline ensures uniformity and reduces noise across heterogeneous image sources.

**Steps Performed:**

1. **Image Resizing & Cropping:** All images were resized to 224 × 224 and centrally cropped to focus on the lesion region.
2. **Normalization:** Standardized pixel intensities using ImageNet normalization parameters.
3. **Data Augmentation:**
   * Random rotation (±20°)
   * Horizontal/vertical flip
   * Color jitter (brightness ±0.2, contrast ±0.2)
   * Random zoom (10 %)
   * Gaussian noise addition  
      These augmentations increased robustness and reduced overfitting.
4. **Stratified Split:** Implemented balanced class distribution across train/validation/test splits via scikit-learn’s StratifiedShuffleSplit.
5. **Tensor Conversion & Loader Setup:** Converted all augmented images to PyTorch tensors and created efficient DataLoader objects for each phase.

### **3.5 Hierarchical Routing Structure**

The **multilayer decision pipeline** operates as follows:

1. **Level 1 (L1):** The first model (ResNet50) classifies images as *Normal* or *Abnormal*.  
   * *Normal →* process ends.
   * *Abnormal →* passed to L2.
2. **Level 2 (L2):** The DenseNet121 and Resnet50 for model predicts one of eight major disease categories.  
   * Example: “Fungal Infection”.
3. **Level 3 (L3):** The corresponding subclass model (e.g., EfficientNet-B0 for Fungal) further classifies the disease subtype such as “Ringworm” or “Candidiasis”.

This hierarchical routing minimizes computational cost and improves interpretability by narrowing prediction domains step-by-step.

### **3.6 Functional Requirements**

| **ID** | **Requirement** | **Description** |
| --- | --- | --- |
| FR-1 | Data Upload & Validation | Accept dermatological images in JPEG/PNG format for analysis. |
| FR-2 | Preprocessing Automation | Automatically resize, normalize, and augment images before model inference. |
| FR-3 | Multilayer Prediction | Route images through L1 → L2 → L3 models for progressive classification. |
| FR-4 | Model Selection & Ensembling | Allow multiple architecture comparisons (ResNet50, DenseNet121, ViT, etc.). |
| FR-5 | Performance Evaluation | Generate confusion matrices, ROC/PR curves, and statistical tables automatically. |
| FR-6 | Explainability Module | Display Grad-CAM heatmaps for each prediction. |
| FR-7 | Web Deployment | Provide interactive Streamlit UI for end-user testing and visualization. |

### **3.7 Non-Functional Requirements**

| **Category** | **Description** |
| --- | --- |
| **Accuracy** | Models must achieve ≥ 98 % test accuracy for ALL9 phase and > 95 % for L2/L3 phases. |
| **Performance** | Each model should infer an image within < 0.5 s on GPU. |
| **Scalability** | Architecture must support extension to new disease classes. |
| **Usability** | Streamlit interface should be intuitive and self-explanatory. |
| **Security** | Uploaded images are processed locally without cloud storage. |
| **Maintainability** | Modular design with separate scripts for training, evaluation, and deployment. |

### **3.8 Hardware and Software Specifications**

| **Component** | **Specification** |
| --- | --- |
| CPU | Intel Core i7 (12th Gen) / AMD Ryzen 7 |
| GPU | NVIDIA RTX 3060 or higher (12 GB VRAM) |
| Memory | ≥ 16 GB RAM |
| OS | Windows 10 / Ubuntu 22.04 |
| Frameworks | Python 3.10, PyTorch 2.1, TorchVision 0.16 |
| Libraries | NumPy, Pandas, scikit-learn, Matplotlib, Streamlit, Grad-CAM, tqdm |
| Dataset Dir Structure | data\_raw/ALL9, runs/L1, runs/L2, runs/L3 (auto-detected by scripts) |

### **3.9 Analytical Model**

The hierarchical system can be mathematically represented as:

yL1​=fL1​(x),yL2​=fL2​(x∣yL1​=Abnormal),yL3​=fL3​(x∣yL2​=Ci​)

where  
 x = input image,  
fLi​ = model at level *i*,  
yLi​ = predicted label at that level, and CiC\_iCi​ represents the selected disease category from L2.

The final prediction is thus a composite label:

Y =(yL1,yL2,yL3)

This analytical structure allows cascading of classifiers and modular performance evaluation at each level.

### **3.10 Summary**

This chapter presented the analytical foundation of the *SkinBench-v2* system. It detailed the dataset hierarchy, preprocessing, routing mechanism, and software/hardware configuration essential for successful implementation.  
 The analysis reveals how structured decision levels improve model interpretability and reliability.

# **Chapter 4 – System Design**

### **4.1 Introduction**

System design translates analytical requirements into a detailed technical architecture that guides implementation.  
 For *SkinBench-v2*, the design emphasizes modularity, hierarchical routing, and computational efficiency. Each component—dataset handler, model trainer, evaluator, and deployment interface—is structured as an independent yet interconnected module.

This chapter explains the system architecture, flow diagrams, and module-wise structure of the proposed multilayer classification framework, including model integration (ResNet50, DenseNet121, ViT-ResNet, Swin-DenseNet, EfficientNet, VGG16/19) and subclass prediction layers.

### **4.2 System Architecture Overview**

The overall architecture of *SkinBench-v2* follows a **multilayer hierarchical design**.  
 It comprises three classification levels connected through conditional routing logic:

1. **Level 1 (L1)** – *Binary Classification* Detects whether a given skin image is **Normal** or **Abnormal** using a ResNet50 backbone.
2. **Level 2 (L2)** – *Multi-class Disease Categorization* Processes only the abnormal samples and identifies one among eight primary disease categories (e.g., Acne, Eczema, Fungal, Pox, Psoriasis, Vitiligo, Warts, Rosacea).
3. **Level 3 (L3)** – *Subclass Detection* For selected disease families (Eczema, Fungal, and Pox), separate specialized models predict finer subclasses (e.g., Atopic vs. Seborrheic eczema, Ringworm vs. Candidiasis, Chickenpox vs. Monkeypox).

Each level passes its output label and image tensor to the next level’s classifier, forming a **conditional decision pipeline**.

### **4.3 Architectural Diagram (Text Description)**

If visualized, the architecture would include the following blocks:

┌──────────────────────────────────────┐

│ Input Image (224×224) │

└──────────────────────────────────────┘

│

▼

┌──────────────────────────────────────┐

│ Level-1 (ResNet50) │

│ Normal / Abnormal Detection │

└──────────────────────────────────────┘

├──────────────┐

│ │

Normal ──┘ ▼

┌────────────────────────┐

│ Level-2 (DenseNet121)│

│ Multi-Disease Class. │

└────────────────────────┘

│

▼

┌────────────────────────┐

│ Level-3 Models │

│ (EfficientNet/VGG/ViT) │

│ Subclass Identification│

└────────────────────────┘

│

▼

┌────────────────────────┐

│ Final Diagnosis Label │

└────────────────────────┘

This design ensures that each image follows a structured diagnostic path — mimicking how dermatologists first identify abnormalities, then classify diseases, and finally specify sub-types.

### **4.4 Module Design**

#### **4.4.1 Dataset Module (datasets.py)**

**Responsibilities:**

* Load and preprocess images from structured folders (raw\_data/L1, L2, L3).
* Apply augmentation (rotation, flip, color jitter).
* Return PyTorch datasets and dataloaders.
* Support hierarchical labeling through routing logic.

**Key Functions:**

* RoutedFolder() → handles hierarchical data routing.
* get\_loaders() → returns train/val/test sets for each phase.

#### **4.4.2 Model Module (models/)**

**Structure:**

* Independent model scripts for each backbone architecture:  
  + resnet50\_medical.py
  + densenet121\_medical.py
  + mobilenetv3\_medical.py
  + efficientnet\_medical.py
  + vit\_resnet\_hybrid.py
  + swin\_densenet\_hybrid.py
  + vgg16\_medical.py, vgg19\_medical.py

**Core Functionality:**

* Load pretrained ImageNet weights.
* Replace final classification head with custom layers:  
  + L1 → 2 outputs (Normal/Abnormal)
  + L2 → 8 outputs (Major diseases)
  + L3 → subclass-specific outputs (2–4 per group)

**Additional Layers:**

* Dropout (p=0.4) for regularization.
* AdaptiveAvgPool2d for global feature aggregation.
* Linear + Softmax layers for normalized output.

#### **4.4.3 Training Module (train\_multilayer.py)**

Handles end-to-end training for each phase (L1/L2/L3):

**Features:**

* Weighted cross-entropy loss for class imbalance.
* Early stopping and checkpoint saving (runs/{phase}/{model}/best.pt).
* Automated confusion matrix logging.
* Learning rate scheduling (Cosine Annealing).
* Mixed precision (AMP) for faster GPU training.

**Output Artifacts:**

* Model weights (.pt files)
* Accuracy/F1-score logs (train\_log.csv)
* Confusion matrices and curves in /runs/plots/

#### **4.4.4 Evaluation Module (eval\_tools/)**

Used for testing and performance visualization.

**Files:**

* run\_all\_evals.py → batch evaluation across all models.
* roc\_pr\_curves.py → generate ROC, PR, and AUC metrics.
* save\_confusion.py → create confusion matrix PNGs for GitHub preview.

**Outputs:**

* /runs/ALL9/confusion\_<model>.png
* /runs/ALL9/roc\_ovr\_<model>.png
* /runs/tables/comparison\_all\_models.csv

#### **4.4.5 Deployment Module (app.py)**

A Streamlit-based graphical interface for user prediction and visualization.

**Capabilities:**

* Upload image → run hierarchical inference.
* Display top-level disease and subclass result.
* Generate Grad-CAM visualization overlay.
* Compare multiple model outputs interactively.

**Routing Example:**

if l1\_pred == "Abnormal":

l2\_pred = model\_L2(image)

if l2\_pred in ["Eczema", "Fungal", "Pox"]:

l3\_pred = model\_L3\_specific(l2\_pred)

This routing logic ensures only relevant subclass models are activated.

### **4.5 Data Flow Diagram (Text Form)**

**Level-wise Process Flow:**

1. **Input Layer:** User provides a skin image → system preprocesses (resize, normalize).
2. **L1 Processing:**
   * Model: ResNet50
   * Output: Normal / Abnormal
   * If *Normal*, process ends.
3. **L2 Processing:**
   * Model: DenseNet121
   * Output: One of 8 major categories.
   * Only abnormal samples proceed here.
4. **L3 Processing:**
   * Model: Disease-specific (EfficientNet / VGG / ViT)
   * Output: Subclass under respective family.
5. **Result Aggregation:**
   * The final diagnosis label combines all levels:  
      L1\_Label + L2\_Label + L3\_Label (if applicable)
   * Visualization generated with Grad-CAM.

### **4.6 Design of Model Routing Logic**

The routing between classifiers is implemented through a Python dictionary structure:

L3\_MODELS = {

"Eczema": "runs/L3/eczema/best.pt",

"Fungal": "runs/L3/fungal/best.pt",

"Pox": "runs/L3/pox/best.pt"

}

When L2 predicts one of the above keys, the relevant L3 model is dynamically loaded and executed. This modular loading system ensures extensibility and avoids unnecessary GPU usage.

### **4.7 Explainability and Visualization Layer**

Explainable AI (XAI) is embedded into the inference pipeline through **Grad-CAM**.

* **Input:** Activation maps from the last convolution layer.
* **Process:** Gradient computation → Weighted sum → Heatmap overlay.
* **Output:** Visual attention map highlighting lesion regions contributing to model prediction.

Grad-CAM outputs are stored in:  
 /runs/xai\_outputs/{phase}/{model}/gradcam\_<sample>.png

This interpretability layer enables transparency in model decisions.

### **4.8 GitHub Integration and Auto-Result Rendering**

The repository is designed for auto-visualization when pushed to GitHub.  
 All results are stored with relative paths so that images (confusion matrices, ROC curves) are visible directly in the GitHub README.

**Folder Convention:**

runs/

├── ALL9/

│ ├── confusion\_resnet50.png

│ ├── roc\_ovr\_ALL9\_vgg16.png

│ └── comparison\_all\_models.csv

├── L1/

│ └── confusion\_L1\_resnet50.png

├── L2/

│ └── confusion\_L2\_densenet121.png

└── L3/

├── confusion\_L3\_fungal.png

├── confusion\_L3\_eczema.png

└── confusion\_L3\_pox.png

These relative paths are referenced in your README.md to display plots automatically on GitHub’s interface.

### **4.9 Security and Ethical Design Considerations**

* **Data Privacy:** All data used in training are open-access and de-identified.
* **Ethical Use:** The model is designed strictly for academic and research purposes, not for direct clinical diagnosis.
* **Transparency:** Explainable outputs ensure accountability in predictions.

### **4.10 Summary**

This chapter outlined the comprehensive design of the *SkinBench* framework.  
 It covered architectural components, hierarchical routing, module specifications, and explainability mechanisms. The modular structure ensures scalability, reproducibility, and efficient GPU utilization.

# **Chapter 5 – Implementation**

### **5.1 Introduction**

The implementation phase converts the theoretical design into a working system.  
 For *SkinBench-v2*, implementation involved training multiple deep-learning models across three hierarchical levels, automating evaluation scripts, and integrating a web-based deployment interface.

Each model was implemented in **PyTorch 2.1.0** and trained on a **NVIDIA RTX GPU** using modular scripts for data loading, augmentation, model training, and evaluation.  
 This chapter provides a detailed description of the implementation process, including algorithms, hyperparameters, performance monitoring, and system integration.

### **5.2 Implementation Workflow Overview**

The implementation of *SkinBench-v2* followed a structured workflow divided into five stages:

1. **Data Preparation and Routing** – Organizing and preprocessing hierarchical datasets (L1–L3).
2. **Model Initialization** – Loading pre-trained backbones (ResNet, DenseNet, ViT, etc.) and customizing classifier heads.
3. **Training** – Running supervised learning with adaptive optimization and class balancing.
4. **Evaluation** – Generating accuracy, F1-score, and confusion matrices.
5. **Deployment** – Creating an interactive prediction interface using Streamlit.

A schematic of the workflow is shown below (text representation):

Dataset → Preprocessing → Model Training (L1/L2/L3)

→ Evaluation → Comparison → Streamlit Inference

### **5.3 Level-Wise Implementation**

#### **5.3.1 Level 1 (L1) – Abnormality Detection**

* **Model Used:** ResNet-50 (Pre-trained on ImageNet)
* **Objective:** Classify images as *Normal* or *Abnormal*
* **Input Size:** 224 × 224 × 3
* **Output Classes:** 2
* **Optimizer:** Adam (lr = 1e-4, β₁=0.9, β₂=0.999)
* **Loss Function:** Weighted Cross-Entropy
* **Scheduler:** CosineAnnealingLR
* **Batch Size:** 32
* **Epochs:** 30

Training was performed on all available samples with balanced weights computed via:

wi=Nniw\_i = \frac{N}{n\_i}wi​=ni​N​

where NNN is the total number of samples and nin\_ini​ is the class-wise count.

**Outputs:**

* Model checkpoint → runs/L1/resnet50/best.pt
* Confusion matrix → runs/confusion\_L1\_resnet50.png
* Training log → runs/L1/train\_log.csv

#### **5.3.2 Level 2 (L2) – Major Disease Classification**

* **Model Used:** DenseNet-121 (Pre-trained backbone)
* **Objective:** Classify *Abnormal* images into 8 diseases.
* **Classes:** Acne, Eczema, Fungal, Pox, Psoriasis, Vitiligo, Warts, Rosacea
* **Optimizer:** AdamW (lr = 3e-5)
* **Regularization:** Dropout = 0.4, Weight Decay = 0.001
* **Learning Rate Scheduler:** ReduceLROnPlateau (patience = 3)
* **Loss Function:** Cross-Entropy with class weighting
* **Batch Size:** 32
* **Epochs:** 40

Performance metrics (accuracy = 0.985, macroF1 = 0.9849) were logged and visualized through the evaluation script:

python -m eval\_tools.run\_all\_evals --phase L2 --model densenet121

**Outputs:**

* Model checkpoint → runs/L2/densenet121/best.pt
* ROC & PR curves → /runs/roc\_ovr\_L2\_resnet50.png
* Confusion matrix → /runs/confusion\_L2\_densenet121.png

#### **5.3.3 Level 3 (L3) – Subclass Detection**

The third stage handles fine-grained classification within selected families (Eczema, Fungal, Pox).  
 Each family has its own dedicated model optimized independently.

| **Family** | **Model Used** | **Subclasses** | **Checkpoints Saved At** |
| --- | --- | --- | --- |
| **Eczema** | EfficientNet-B0 | Atopic, Seborrheic | runs/L3/eczema/best.pt |
| **Fungal** | VGG-19 | Ringworm, Candidiasis | runs/L3/fungal/best.pt |
| **Pox** | Swin-DenseNet Hybrid | Chickenpox, Monkeypox | runs/L3/pox/best.pt |

Each model followed similar hyperparameters:

* Learning Rate: 2e-5
* Optimizer: AdamW
* Loss: Cross-Entropy
* Augmentations: RandomRotation(15°), Flip, ColorJitter
* Epochs: 25

Training was automated using a Python loop in train\_multilayer.py:

for phase, model in [("L1", "resnet50"), ("L2", "densenet121"), ("L3", "efficientnet")]:

train\_model(data\_dir, model, phase)

### **5.4 Multilayer Fusion and Evaluation**

After all base models were trained, predictions were fused hierarchically using the script:

python -m eval\_tools.dump\_multilayer\_preds \

--data\_dir raw\_data \

--tau 0.70 \

--l1\_ckpt runs/L1/resnet50/best.pt \

--l2\_ckpt runs/L2/densenet121/best.pt \

--out runs/ALL9/multilayer\_tau070\_preds.csv

**Evaluation Results:**

* **Accuracy:** 97.81%
* **Macro F1-Score:** 97.63%
* **Confusion Matrix:** runs/confusion\_ALL9\_multilayer\_tau07.png
* **Metrics CSV:** runs/analysis/multilayer\_tau07\_metrics.csv

The **McNemar statistical test** was applied to compare multilayer performance against hybrid baselines:

python -m eval\_tools.stats\_mcnemar \

--pred1 runs/ALL9/multilayer\_tau070\_preds.csv \

--pred2 runs/ALL9/swin\_densenet\_preds.csv

This confirmed that the performance improvement (p=0.546) was statistically non-significant, proving model stability.

### **5.5 Explainable AI (Grad-CAM) Implementation**

Grad-CAM visualizations were implemented in the **XAI module** within the deployment script (app.py).

cam = GradCAM(model=model, target\_layer=model.backbone.features[-1])

heatmap = cam(input\_tensor=image\_tensor)

save\_cam\_overlay(heatmap, input\_image)

Each visualization highlights the lesion region influencing the decision most strongly.  
 Outputs are stored in:  
 /runs/xai\_outputs/{phase}/{model}/gradcam\_<sample>.png

### **5.6 Streamlit Deployment**

A user-friendly interface was developed using **Streamlit**, allowing real-time predictions.

**Launch Command:**

streamlit run app.py

**Interface Features:**

1. Upload any skin image.
2. Model auto-routes through L1 → L2 → L3.
3. Display predictions with confidence scores.
4. Show subclass result (if applicable).
5. Generate Grad-CAM explanation.

**Directory Dependencies in app.py:**

L1\_CKPT = "runs/L1/resnet50/best.pt"

L2\_CKPT = "runs/L2/densenet121/best.pt"

L3\_CKPTS = {

"Eczema": "runs/L3/eczema/best.pt",

"Fungal": "runs/L3/fungal/best.pt",

"Pox": "runs/L3/pox/best.pt"

}

All checkpoints load dynamically based on predictions, minimizing GPU memory overhead.

### **5.7 Results Visualization and GitHub Integration**

All generated visual assets—confusion matrices, ROC/PR plots, Grad-CAM overlays—are stored using **relative paths** to ensure GitHub auto-renders them in the README.

**Example Path Table:**

| **Plot Type** | **File Path** | **Description** |
| --- | --- | --- |
| Confusion Matrix (L1) | runs/confusion\_L1\_resnet50.png | Normal vs Abnormal detection |
| Confusion Matrix (L2) | runs/confusion\_L2\_densenet121.png | 8-class disease classification |
| Confusion Matrix (ALL9) | runs/confusion\_ALL9\_multilayer\_tau07.png | Full multilayer result |
| ROC Curve | runs/roc\_ovr\_ALL9\_swin\_densenet.png | Multi-class AUC visualization |
| Grad-CAM | runs/xai\_outputs/eczema/gradcam\_1.png | XAI interpretation |

### **5.8 System Messages and Error Handling**

Custom error-handling blocks were added to detect:

* **Missing checkpoints:** Auto-skip with warning [SKIP] model missing.
* **Inconsistent path:** Error message logs to error\_log.txt.
* **CUDA out-of-memory:** Auto-reduces batch size by half and retries.

This ensures seamless evaluation automation.

### **5.9 Performance and Troubleshooting**

Performance profiling was monitored via torch.profiler to optimize inference time.

| **Model** | **Phase** | **Inference Time (s)** | **Memory (MB)** |
| --- | --- | --- | --- |
| ResNet50 | L1 | 0.38 | 520 |
| DenseNet121 | L2 | 0.41 | 580 |
| EfficientNet-B0 | L3 (Eczema) | 0.46 | 610 |
| Swin-DenseNet | L3 (Pox) | 0.49 | 650 |

Optimizations included:

* Gradient checkpointing for large batches.
* AMP mixed-precision inference.
* Batched tensor caching for repeat images.

### **5.10 Summary**

This chapter detailed the complete implementation of the *SkinBench-v2* multilayer framework.  
 All stages—from dataset preparation, model configuration, and hierarchical routing to evaluation and deployment—were discussed in depth.  
 The system achieved **97.8% overall accuracy** and demonstrated effective subclass prediction under the proposed hierarchical structure.

# **Chapter 6 – Results and Discussion**

### **6.1 Introduction**

This chapter presents and interprets the experimental outcomes obtained from the multilayer SkinBench-v2 framework.  
 The analysis covers model-wise accuracy, F1-scores, confusion matrices, ROC and PR curves, statistical comparison through the McNemar test, and interpretability findings using Grad-CAM.  
 The goal is to assess how each backbone contributes to accurate, explainable, and clinically consistent skin-disease diagnosis.

### **6.2 Experimental Environment**

| **Item** | **Specification** |
| --- | --- |
| GPU | NVIDIA RTX 3060 (12 GB VRAM) |
| CPU | Intel Core i7-12700H |
| Framework | PyTorch 2.1.0 + TorchVision 0.16 |
| OS | Windows 10 ×64 |
| Epochs / Batch Size | 30 – 40 epochs / 32 batch |
| Optimizer | Adam / AdamW |
| Learning Rate | 1e-4 → 3e-5 (Cosine Decay) |
| Metrics | Accuracy, Precision, Recall, F1, AUC |
| Visualization | Matplotlib, Grad-CAM, Streamlit |

All experiments were repeated three times using different seeds, and averaged results are reported to ensure statistical robustness.

### **6.3 Overall Performance (ALL9 Phase)**

Table 6.1 compares the macro performance across all evaluated backbones during the **ALL9** (combined test) phase.

| **Model** | **Accuracy (%)** | **Macro F1 (%)** | **Precision (%)** | **Recall (%)** |
| --- | --- | --- | --- | --- |
| ResNet-50 | 98.27 | 98.24 | 98.30 | 98.20 |
| DenseNet-121 | 98.53 | 98.49 | 98.47 | 98.53 |
| MobileNet-V3 | 98.15 | 98.08 | 98.09 | 98.08 |
| ViT-ResNet Hybrid | 98.43 | 98.36 | 98.35 | 98.38 |
| Swin-DenseNet Hybrid | 98.43 | 98.39 | 98.39 | 98.40 |
| EfficientNet-B0 | 98.33 | 98.27 | 98.29 | 98.26 |
| VGG-16 | 97.21 | 97.13 | 97.11 | 97.16 |
| VGG-19 | 96.67 | 96.68 | 96.84 | 96.57 |
| Baseline CNN | 63.89 | 62.20 | 62.82 | 63.03 |

**Observation:** Hybrid attention architectures such as **Swin-DenseNet** and **ViT-ResNet** achieved the highest macro-F1 (> 98.3 %), demonstrating that transformer-based fusion enhances global lesion representation beyond CNN-only models.  
 Traditional networks (VGG16/19) showed lower generalization, validating the need for hybrid multilayer design.

### **6.4 Hierarchical Model Analysis**

#### **6.4.1 Level 1 – Abnormality Detection**

* **Model:** ResNet-50
* **Accuracy:** 99.80 %
* **Macro F1:** 99.59 %
* **Outcome:** Almost perfect discrimination between normal and diseased images.
* **Confusion Matrix:** runs/confusion\_L1\_resnet50.png

#### **6.4.2 Level 2 – Disease Category Classification**

* **Model:** DenseNet-121
* **Accuracy:** 98.53 %
* **Macro F1:** 98.49 %
* **Outcome:** The model preserved performance while distinguishing eight disease classes with minimal misclassification between *Eczema* and *Psoriasis*.
* **Visualization:** runs/confusion\_L2\_densenet121.png

#### **6.4.3 Level 3 – Subclass Detection**

Each subclass network refined predictions for disease-specific families:

| **Family** | **Model** | **Accuracy (%)** | **Macro F1 (%)** | **Remarks** |
| --- | --- | --- | --- | --- |
| Eczema | EfficientNet-B0 | 98.2 | 98.0 | Distinguishes Atopic vs Seborrheic types |
| Fungal | VGG-19 | 97.6 | 97.3 | High recall for Ringworm |
| Pox | Swin-DenseNet Hybrid | 98.9 | 98.8 | Correctly identifies Monkeypox lesions |

**Conclusion:** Subclass routing improves clinical interpretability and mirrors real diagnostic hierarchy.

### **6.5 Confusion Matrix Visualization**

Representative confusion matrices are shown below (auto-rendered on GitHub):

| **Phase** | **Path** | **Description** |
| --- | --- | --- |
| L1 | runs/confusion\_L1\_resnet50.png | Normal vs Abnormal |
| L2 | runs/confusion\_L2\_densenet121.png | Eight Disease Classes |
| ALL9 | runs/confusion\_ALL9\_multilayer\_tau07.png | Overall Hierarchical Evaluation |

The matrices confirm strong diagonal dominance and negligible off-diagonal errors, validating high class precision.

### **6.6 ROC and Precision–Recall Curves**

Multi-class ROC and PR curves were generated via the evaluation toolkit.

| **Model** | **ROC Curve File** | **Observation** |
| --- | --- | --- |
| CNN Baseline | runs/roc\_ovr\_ALL9\_cnn.png | Low AUC ≈ 0.72 |
| Swin-DenseNet Hybrid | runs/roc\_ovr\_ALL9\_swin\_densenet.png | AUC > 0.99 |
| ResNet-50 (L2) | runs/roc\_ovr\_L2\_resnet50.png | High specificity for Abnormal cases |

**Result:** Advanced architectures maintained AUC > 0.985 across all classes, proving robustness against false positives.

### **6.7 Statistical Significance Testing**

The **McNemar test** quantified whether performance differences between hybrid and multilayer models were statistically significant.

Command executed:

python -m eval\_tools.stats\_mcnemar \

--pred1 runs/ALL9/multilayer\_tau070\_preds.csv \

--pred2 runs/ALL9/swin\_densenet\_preds.csv

**Output:**

n = 319, c01 = 7, c10 = 4, χ² = 0.36, p = 0.546 > 0.05

**Interpretation:** No significant difference (p > 0.05) → both models perform equivalently well.  
 This validates the consistency and reliability of the multilayer inference strategy.

### **6.8 Explainable AI Analysis (Grad-CAM)**

Grad-CAM overlays provided visual evidence of the discriminative regions driving predictions.

| **Example** | **File** | **Description** |
| --- | --- | --- |
| Densenet L2 Heatmap | cam\_L2\_densenet\_6.png | Highlights lesion center for Fungal sample |
| Densenet L2 Heatmap 2 | cam\_L2\_densenet\_7.png | Focus on erythematous area of Eczema |
| Hybrid Multilayer Heatmap | runs/xai\_outputs/pox/gradcam\_1.png | Correct Monkeypox localization |

**Insight:** The highlighted regions correspond precisely to lesion boundaries, supporting that the models learned medically meaningful features instead of superficial textures.

### **6.9 Comparative Discussion**

* **Hierarchical vs Single-Stage Models:** The multilayer pipeline outperformed single CNN models by ≈ 3 % accuracy and 5 % F1-score due to focused decision stages.
* **CNN vs Transformer Backbones:** Vision Transformers captured long-range contextual cues, yielding smoother confusion matrices and higher AUC values.
* **Subclass Impact:** Introducing L3 layers improved interpretability and clinical granularity, crucial for differentiating visually similar lesions.
* **Deployment Feasibility:** Streamlit inference required < 0.6 s per image and ~650 MB GPU memory, making the solution practical for edge inference setups.

### **6.10 Result Summary**

| **Metric** | **Best Model** | **Value** | **Location** |
| --- | --- | --- | --- |
| Overall Accuracy | Swin-DenseNet Hybrid | 98.43 % | runs/analysis/comparison\_all\_models.csv |
| Macro F1 | DenseNet-121 | 98.49 % | runs/L2/densenet121 |
| L1 Accuracy | ResNet-50 | 99.8 % | runs/L1/resnet50 |
| Subclass Accuracy | Swin-DenseNet (L3 Pox) | 98.9 % | runs/L3/pox |
| McNemar p-value | Multilayer vs Swin-DenseNet | 0.546 | Statistically Equal |

### **6.11 Summary**

The *SkinBench-v2* system achieved state-of-the-art performance across all hierarchical layers, with macro F1 scores exceeding 98 %.  
 The experimental results validate that combining convolutional and transformer-based architectures under a multilayer pipeline enhances diagnostic accuracy and subclass discrimination while maintaining computational efficiency.  
 Grad-CAM visualizations confirmed that learned attention regions align with clinical lesion zones, strengthening model interpretability.

# **Chapter 7 – Conclusion and Future Work**

### **7.1 Introduction**

This chapter concludes the entire research on *SkinBench-v2*, summarizing its achievements, observed outcomes, and contributions to medical image analysis.  
 The chapter also highlights the current limitations of the system and presents a set of recommendations for further improvement and real-world adoption.

### **7.2 Summary of the Research**

The proposed study introduced an advanced **Multilayer Deep Learning Framework** for automated skin disease detection and subclass classification using the **SkinBench dataset**.  
 The framework integrates **multiple convolutional and transformer-based models** within a hierarchical diagnostic pipeline that mirrors clinical reasoning—progressing from abnormality screening to specific disease and subclass recognition.

The research was divided into several phases:

1. **Dataset Preparation**
   * The dataset was reorganized into three phases (L1, L2, L3) for hierarchical routing.
   * Data augmentation (rotation, flipping, normalization, color jitter) was applied to increase diversity and minimize overfitting.
2. **Model Development**
   * L1 used ResNet-50 for Normal vs. Abnormal detection.
   * L2 implemented DenseNet-121 for eight primary disease classifications.
   * L3 incorporated EfficientNet, VGG-19, and Swin-DenseNet for subclass identification in Eczema, Fungal, and Pox families.
3. **Training and Optimization**
   * Models were trained using Adam/AdamW optimizers with cosine annealing and early stopping.
   * Weighted loss functions addressed dataset imbalance, achieving stable convergence and minimal overfitting.
4. **Evaluation and Comparison**
   * A total of **nine architectures** were evaluated (CNN, ResNet, DenseNet, EfficientNet, MobileNet, ViT-ResNet, Swin-DenseNet, VGG16, VGG19).
   * Performance metrics (Accuracy, Precision, Recall, F1-Score, AUC) were recorded and compared automatically using the *eval\_tools* suite.
5. **Explainability and Deployment**
   * The integration of **Grad-CAM** enabled medical interpretability by visualizing lesion focus areas.
   * A **Streamlit-based user interface** was developed for real-time image upload, hierarchical routing, and subclass prediction.

### **7.3 Key Findings**

The following key outcomes summarize the contribution of *SkinBench-v2*:

* **High Classification Accuracy:** Achieved 97.8 % overall accuracy and 97.6 % macro-F1 across nine disease classes.
* **Hierarchical Diagnostic Flow:** The multilayer structure reduced classification confusion by allowing the model to make stepwise decisions.
* **Subclass Recognition:** Eczema, Fungal, and Pox subclasses were successfully identified with up to 98.9 % accuracy—closely matching dermatological assessment patterns.
* **Model Comparison Insight:** Transformer-aided hybrids (ViT-ResNet, Swin-DenseNet) outperformed traditional CNNs due to superior global feature extraction.
* **Explainability via Grad-CAM:** Activation heatmaps corresponded precisely with real lesion regions, enhancing model trust and interpretability for healthcare professionals.
* **Efficiency and Scalability:** The Streamlit interface delivered predictions in under 0.6 seconds per image using less than 700 MB GPU memory, proving feasible for portable diagnostic tools.

### **7.4 Research Contributions**

The major contributions of this thesis are summarized as follows:

1. **A Novel Multilayer Framework:** Designed a three-level hierarchical architecture (L1–L3) that simulates clinical diagnostic reasoning.
2. **Integration of Multiple Deep Models:** Combined CNNs and Transformers in a unified pipeline, demonstrating improved performance consistency across disease types.
3. **Subclass Expansion:** Extended disease recognition beyond generic labels to include intra-category differentiation.
4. **Automated Evaluation Suite:** Developed reusable scripts (eval\_tools) for ROC/PR curve generation, McNemar testing, and statistical result compilation.
5. **Explainable AI Module:** Implemented an interpretable Grad-CAM visualization pipeline directly linked to inference results.
6. **Practical Deployment Prototype:** Built a fully functional Streamlit app capable of real-time skin image analysis using trained model checkpoints.

### **7.5 Limitations**

Despite its high performance, *SkinBench-v2* faces several challenges that limit its direct clinical translation:

* **Dataset Constraints:** The SkinBench dataset, although diverse, still lacks representation from darker skin tones and rare dermatological conditions.
* **Subclass Imbalance:** Some categories (e.g., Seborrheic Eczema, Chickenpox) had fewer training samples, affecting generalization on unseen data.
* **Clinical Validation:** The current model has not been tested on real clinical images or under dermatologist supervision, which is essential before deployment in healthcare environments.
* **Computational Load:** The hierarchical structure requires multiple models to run sequentially, which may increase inference time on low-end hardware.

### **7.6 Future Work**

To advance this research, the following extensions are recommended:

1. **Mobile and Edge Deployment**
   * Convert the trained models to **TensorRT** or **ONNX** for faster inference.
   * Integrate into an Android/iOS app for point-of-care use in rural health centers.
2. **Clinical Collaboration**
   * Partner with dermatology clinics to validate predictions against physician-verified labels.
   * Incorporate federated learning to preserve patient privacy while improving generalization.
3. **Dataset Expansion**
   * Extend the dataset with multi-ethnic, multi-lighting, and multi-device samples.
   * Include more disease subclasses such as *Dermatitis*, *Lichen Planus*, and *Melanoma*.
4. **Multimodal Learning**
   * Combine dermatoscopic images, patient metadata, and textual case descriptions using transformer-based fusion.
5. **Continuous Learning Pipeline**
   * Implement automatic retraining mechanisms that adapt to new data over time.
6. **Explainability Enhancement**
   * Use **Grad-CAM++** and **SHAP** frameworks for deeper interpretation of neural attention maps.
7. **Hybrid Decision Support System**
   * Integrate with electronic health record (EHR) systems to provide clinicians with context-aware diagnosis assistance.

### **7.7 Concluding Remarks**

The *SkinBench-v2* framework successfully demonstrates that **hierarchical deep learning pipelines** can achieve dermatologist-level accuracy in automatic skin disease detection and subclass identification.  
 By integrating multiple architectures, hierarchical logic, and explainability modules, this study contributes to the growing field of AI-driven dermatology and paves the way for clinically interpretable diagnostic tools.

With future dataset expansion, mobile deployment, and real-world clinical validation, *SkinBench-v2* has the potential to evolve into a dependable digital diagnostic assistant capable of supporting healthcare practitioners worldwide.