|  |  |
| --- | --- |
| MONKEYPOX DISEASE CLASSIFICATION USING DEEP LEARNING | An Image-Based Disease Classification System  Department of Artificial Intelligence & Data Science EASA College of Engineering & Technology Academic Year: 2025 – 2026 |

**TITLE**

**Monkeypox Disease Classification Using Deep Learning**

**AUTHORS**

|  |  |  |
| --- | --- | --- |
| **Rini.P** *Department of Artificial Intelligence and Data Science* ***Easa College of Engineering & Technology (Autonomous), Coimbatore, Tamil Nadu, India*** | **Ragul.S** *Department of Artificial Intelligence and Data Science* ***Easa College of Engineering & Technology (Autonomous), Coimbatore, Tamil Nadu, India*** | **Sabarirajan.S** *Department of Artificial Intelligence and Data Science* ***Easa College of Engineering & Technology (Autonomous), Coimbatore, Tamil Nadu, India*** |

**Project Guide:**  
Hima Vyshnavi  
Assistant Professor  
Department of Artificial Intelligence and Data Science  
EASA College of Engineering & Technology

**1. ABSTRACT**

Monkeypox is a viral infectious disease that primarily affects the skin and causes symptoms such as rashes, blisters, fever, and body pain. In recent years, the number of Monkeypox cases has increased globally, highlighting the need for early and accurate diagnosis. Conventional diagnosis relies on clinical examination and laboratory tests, which require trained medical professionals and proper healthcare infrastructure. In remote and underdeveloped regions, limited access to such facilities can delay diagnosis and treatment.

With the rapid advancement of Artificial Intelligence and Deep Learning, image-based disease classification has emerged as a promising solution for early detection of skin diseases. Deep learning models are capable of learning complex visual patterns from medical images and can assist in automated disease identification. Skin lesion image analysis can help reduce dependency on manual diagnosis and support timely medical decision-making.

In this project, a deep learning-based system is developed for Monkeypox disease classification using skin lesion images. A publicly available multiclass skin disease dataset containing images of Monkeypox and other viral skin diseases is used. The images are preprocessed through resizing, normalization, and data augmentation techniques to enhance image quality and improve model generalization.

A hybrid deep learning architecture combining a convolutional neural network and a transformer-based model is proposed. ResNet-50 is used as a feature extractor, and transformer encoder layers are employed to capture global contextual information from image features. The model is trained and evaluated using multiple performance metrics, including accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC.

The trained model is deployed as a user-friendly web application using Streamlit, allowing users to upload skin images and obtain disease predictions along with confidence scores. This project is intended for academic and educational purposes, focusing on the practical implementation of deep learning techniques, and is not meant to replace professional medical diagnosis.

**2. INTRODUCTION**

Skin diseases are among the most common health problems affecting people of all age groups. The skin is the outermost protective layer of the human body and acts as a barrier against infections, bacteria, and viruses. When the skin is affected by viral infections, visible symptoms such as rashes, red spots, blisters, and lesions appear on the skin surface. These visible characteristics play an important role in identifying and differentiating various skin diseases.

Monkeypox is a viral skin disease belonging to the Orthopoxvirus family. It presents symptoms similar to smallpox, including fever, headache, body pain, swollen lymph nodes, and skin rashes. The rashes usually begin on the face and gradually spread to other parts of the body. In recent years, Monkeypox cases have been reported across multiple countries, raising serious concerns among global health organizations.

Conventionally, Monkeypox is diagnosed through clinical examination and laboratory-based tests such as Polymerase Chain Reaction (PCR). These diagnostic procedures require specialized medical equipment, trained healthcare professionals, and well-established medical facilities. In rural and remote regions, access to such facilities is often limited, leading to delays in diagnosis and treatment and increasing the risk of disease transmission.

Manual diagnosis of skin diseases also depends heavily on the experience and expertise of medical professionals. Monkeypox shares visual similarities with other viral skin diseases such as Chickenpox, Measles, and Cowpox, especially during the early stages. This resemblance can lead to misclassification and diagnostic uncertainty. Therefore, an automated image-based classification system can serve as a valuable support tool for preliminary disease identification.

Recent advancements in Artificial Intelligence (AI) and Deep Learning have demonstrated significant potential in medical image analysis. Deep learning models can automatically learn hierarchical features from images and identify complex visual patterns. Convolutional Neural Networks (CNNs) are widely used for image classification tasks due to their effectiveness in capturing spatial features such as edges, shapes, and textures. Additionally, transformer-based architectures have gained attention for their ability to model global contextual relationships within image data.

In this project, a deep learning-based approach is proposed for Monkeypox disease classification using skin lesion images. A publicly available multiclass skin disease dataset is utilized, and image preprocessing techniques such as resizing, normalization, and data augmentation are applied. A hybrid deep learning architecture combining a convolutional neural network and a transformer-based model is designed, where ResNet-50 is used for feature extraction and transformer layers are employed for enhanced representation learning. The model is trained and evaluated using standard performance metrics to assess its effectiveness.

Furthermore, the trained model is deployed as a web-based application using Streamlit, enabling users to upload skin images and obtain classification results along with confidence scores. The primary objective of this project is to gain practical understanding of deep learning techniques and their application in healthcare-related image classification problems. The system is developed for academic and educational purposes and is not intended to replace professional medical diagnosis.

**3. PROBLEM STATEMENT**

Skin diseases caused by viral infections are becoming a serious health concern in many parts of the world. Monkeypox is one such viral disease that presents visible symptoms on the skin in the form of rashes, blisters, and lesions. Early identification of Monkeypox is essential to prevent disease transmission and to initiate timely medical intervention. However, accurate identification of Monkeypox during the early stages remains a challenging task.

Conventional diagnosis of Monkeypox involves physical examination by medical professionals followed by laboratory-based tests such as Polymerase Chain Reaction (PCR). These diagnostic methods require specialized medical equipment, trained healthcare personnel, and well-established medical infrastructure. In rural and remote regions, access to such facilities is often limited, resulting in delayed diagnosis and increased risk of disease spread.

Another significant challenge in Monkeypox diagnosis is the visual similarity of its symptoms with other viral skin diseases such as Chickenpox, Measles, and Cowpox. These diseases exhibit similar skin rashes and lesions, particularly in the initial stages. Due to this similarity, manual diagnosis based solely on visual inspection may lead to misclassification or delayed identification, which can adversely affect patient care.

With the rising number of Monkeypox cases, the workload on healthcare professionals has also increased. Physicians are required to evaluate a large number of patients within limited timeframes, which increases the possibility of human error. Hence, there is a growing need for an automated supporting system that can assist medical professionals by providing preliminary classification based on skin image analysis.

Artificial Intelligence and deep learning techniques offer a potential solution to this problem by enabling automatic extraction of visual features from skin images and accurate disease classification. However, developing an effective deep learning model for Monkeypox classification is challenging due to factors such as limited availability of labeled medical images, variation in image quality, and high similarity between different viral skin conditions.

The problem addressed in this project is the design and implementation of a deep learning-based image classification system capable of distinguishing Monkeypox from other viral skin diseases. The proposed system aims to analyze skin lesion images, extract meaningful features using a hybrid convolutional neural network and transformer-based architecture, and perform accurate multi-class classification. Additionally, the system should be user-friendly and accessible through a simple web-based interface for academic demonstration purposes.

This project focuses on developing an academic prototype that demonstrates the application of deep learning techniques for Monkeypox disease classification and is not intended to replace professional medical diagnosis.

**4. OBJECTIVES OF THE PROJECT**

The main objective of this project is to study and implement a deep learning-based system for Monkeypox disease classification using skin lesion image data. This project is carried out as a learning-oriented academic project to understand how modern deep learning techniques can be applied in the healthcare domain, particularly in medical image classification.

The specific objectives of the project are listed below.

**4.1 Primary Objective**

The primary objective of this project is:

• **To develop a deep learning-based image classification system capable of identifying Monkeypox disease from skin lesion images**

This objective focuses on designing and implementing a system that can accept a skin image as input and provide a prediction indicating whether the image belongs to Monkeypox or other related viral skin diseases.

**4.2 Secondary Objectives**

In addition to the primary objective, the following secondary objectives are defined:

1. **To study Monkeypox disease and its skin-related symptoms**  
   This objective involves understanding the visual characteristics of Monkeypox such as rashes, blisters, and lesion patterns that are relevant for image-based classification.
2. **To collect and analyze a medical image dataset**  
   The project aims to work with a publicly available multiclass skin lesion dataset and understand its structure, class distribution, and suitability for deep learning-based analysis.
3. **To perform dataset cleaning and image preprocessing**  
   This includes removing low-quality or corrupted images and applying preprocessing techniques such as resizing, normalization, and image enhancement to improve data quality.
4. **To apply data augmentation techniques**  
   Data augmentation techniques are used to artificially increase training data diversity and reduce overfitting caused by limited dataset size.
5. **To design a deep learning model using pre-trained architectures**  
   The objective is to utilize transfer learning by employing a pre-trained convolutional neural network as a feature extractor and integrating transformer-based layers to improve feature representation.
6. **To train and evaluate the proposed deep learning model**  
   The model is trained using training data and evaluated using multiple performance metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC.
7. **To analyze model performance and classification behavior**  
   This objective focuses on understanding model behavior by analyzing misclassification patterns, class-wise performance, and potential overfitting issues.
8. **To deploy the trained model using a web-based application**  
   A simple and user-friendly web application is developed using Streamlit to allow users to upload skin images and obtain classification results along with confidence scores.

**4.3 Learning Objectives**

In addition to technical objectives, this project aims to achieve the following learning outcomes:

• Understanding the fundamentals of deep learning and image classification  
• Gaining experience in handling real-world medical image datasets  
• Learning image preprocessing and data augmentation techniques  
• Understanding model training, evaluation, and performance analysis  
• Learning basic deployment of deep learning models using web frameworks

**4.4 Overall Objective Summary**

Overall, the objective of this project is not only to perform Monkeypox disease classification but also to gain hands-on experience in applying deep learning techniques to solve healthcare-related image analysis problems. The project helps students understand the complete workflow from dataset preparation and model training to evaluation and deployment.

**5. SCOPE OF THE PROJECT**

The scope of this project defines the boundaries and limitations within which the Monkeypox disease classification system is developed. This project is carried out as an undergraduate academic project, and its scope is mainly educational and experimental in nature. The system is designed to demonstrate the application of deep learning techniques for medical image classification.

**5.1 Technical Scope**

The technical scope of the project includes the following aspects:

• The system focuses on image-based classification of Monkeypox disease using deep learning techniques.  
• Skin lesion images are used as the primary input for classification.  
• The deep learning model is trained using a publicly available multiclass skin disease dataset.  
• Transfer learning techniques are employed to improve model performance.  
• A hybrid deep learning architecture combining a convolutional neural network and transformer-based layers is used for feature extraction and classification.  
• The system performs multi-class classification, where Monkeypox is classified along with other viral skin diseases.

The technical scope is limited to training, evaluation, and testing of the model using available image data. Advanced clinical techniques such as lesion segmentation or medical decision support systems are not included.

**5.2 Functional Scope**

The functional scope of the system includes the following features:

• Accepting skin lesion images as input from users.  
• Preprocessing the uploaded images to match the model input requirements.  
• Classifying the images into Monkeypox or other related viral skin disease categories.  
• Displaying the predicted disease label along with confidence scores.  
• Providing a simple and user-friendly web interface developed using Streamlit.

The system does not provide medical advice or treatment recommendations. It serves only as an academic demonstration tool.

**5.3 User Scope**

The intended users of this system include:

• Engineering students interested in learning deep learning applications.  
• Academic researchers working on medical image classification.  
• Faculty members evaluating undergraduate academic projects.

The system is not designed for use by medical professionals for clinical diagnosis.

**5.4 Dataset Scope**

The dataset scope of this project includes:

• Publicly available skin lesion images collected from open medical datasets.  
• Multiple viral skin disease classes, including Monkeypox.  
• Images preprocessed and resized to a uniform resolution of **224 × 224 pixels**.  
• Dataset split into training, validation, and testing subsets.

The dataset does not include patient information such as personal details, medical history, or demographic data.

**5.5 Limitations of Scope**

The scope of this project does not include:

• Real-time clinical diagnosis in hospitals.  
• Integration with hospital information or management systems.  
• Use of live or patient-specific clinical data.  
• Medical certification or regulatory approval

**5.6 Scope Summary**

In summary, the scope of this project is limited to the development of a prototype system that demonstrates how deep learning techniques can be applied to Monkeypox disease classification using skin lesion images. The project mainly focuses on learning, experimentation, and academic understanding rather than real-world medical deployment.

**6. LITERATURE SURVEY**

The literature survey provides an overview of existing research related to skin disease classification, Monkeypox detection, and the application of deep learning techniques in medical image analysis. Reviewing earlier studies helps in understanding commonly used methods, existing challenges, and research gaps in this domain

**6.1 Skin Disease Classification Using Image Processing**

Early research on skin disease classification mainly relied on traditional image processing and machine learning techniques. Methods such as color histogram analysis, edge detection, texture extraction, and shape-based features were commonly used. These manually extracted features were then classified using algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees.

Although these approaches showed reasonable performance in controlled environments, they suffered from several limitations. Manual feature extraction required expert knowledge and lacked robustness to variations in lighting, image quality, and background noise. As a result, these methods showed reduced accuracy when applied to real-world skin images.

**6.2 Use of Convolutional Neural Networks in Medical Imaging**

With the rapid growth of deep learning, Convolutional Neural Networks (CNNs) became widely adopted for medical image analysis. CNNs automatically learn hierarchical features directly from raw images, eliminating the need for manual feature engineering. Numerous studies have successfully applied CNNs for tasks such as skin cancer detection, diabetic retinopathy screening, pneumonia classification, and dermatological disease identification.

CNN-based models demonstrated superior performance compared to traditional machine learning approaches. However, training deep CNNs from scratch requires large labeled datasets, which are often scarce in medical domains.

**6.3 Transfer Learning for Skin Disease Detection**

To address the challenge of limited medical image datasets, transfer learning techniques have been widely adopted. In transfer learning, models pre-trained on large-scale datasets such as ImageNet are fine-tuned for specific medical tasks. Popular architectures such as VGG, ResNet, DenseNet, and EfficientNet have been used as feature extractors for skin disease classification.

Several studies reported that transfer learning improves classification accuracy, accelerates model convergence, and reduces overfitting. Typically, initial layers are frozen to preserve learned features, while deeper layers are fine-tuned using domain-specific medical images.

**6.4 Research on Viral Skin Diseases**

Research focusing specifically on viral skin diseases such as Chickenpox, Measles, Cowpox, and Monkeypox is relatively limited. Most existing dermatology-related deep learning studies primarily focus on skin cancer or melanoma detection.

Monkeypox classification using image-based deep learning approaches is a comparatively new research area. A few recent studies have proposed CNN-based models for viral skin disease detection, showing promising results. However, these studies highlight challenges such as dataset imbalance, limited sample size, and visual similarity among different viral infections.

**6.5 Multi-Class Skin Disease Classification**

Multi-class skin disease classification is more challenging than binary classification because the model must distinguish between multiple visually similar disease categories. Diseases such as Monkeypox, Chickenpox, and Measles often exhibit overlapping visual characteristics, especially in early stages.

Recent studies have explored multi-class classification using deeper CNN architectures and ensemble models. While improved accuracy has been reported, issues such as overfitting, class imbalance, and reduced generalization remain significant challenges, particularly when training data is limited.

**6.6 Public Viral Skin Disease Datasets and Recent Contributions**

Recent public datasets have been introduced to support research on viral skin disease classification. One such dataset is the Multi-Class Viral Skin Lesion Dataset (MCVSLD), which includes images of Monkeypox, Chickenpox, Cowpox, HFMD, Measles, and healthy skin.

Since this dataset is relatively new, only limited research has explored its full potential. Most existing works focus on CNN-based approaches, and comprehensive studies involving hybrid architectures and detailed performance evaluation remain limited.

**6.7 Research Gaps Identified**

Based on the literature survey, the following research gaps are identified:

• Limited research focusing exclusively on Monkeypox disease classification  
• Scarcity of large, balanced, and high-quality viral skin disease datasets  
• Difficulty in differentiating visually similar viral skin diseases  
• Limited use of hybrid CNN–Transformer architectures in this domain  
• Lack of model interpretability techniques such as Grad-CAM  
• Limited deployment of trained models through user-accessible web applications

**6.8 Summary of Literature Survey**

The literature survey indicates that deep learning techniques, particularly CNNs combined with transfer learning, outperform traditional machine learning approaches for skin disease classification. However, challenges such as dataset limitations, class imbalance, disease similarity, and lack of interpretability still persist. This project attempts to address these challenges at an academic level by using a recent multi-class dataset, a hybrid CNN–Transformer architecture, detailed performance evaluation, and a deployable web-based application.

**7. DATASET DESCRIPTION**

The dataset plays a crucial role in any deep learning-based classification system, as model performance is highly dependent on data quality, diversity, and balance. In this project, a publicly available multi-class skin lesion dataset is used for Monkeypox disease classification.

**7.1 Dataset Overview**

The dataset used in this project is the **Multi-Class Viral Skin Lesion Dataset (MCVSLD)**. This dataset is a recently released public dataset designed for academic and research purposes. It contains skin lesion images representing multiple viral skin diseases along with healthy skin samples.

The dataset includes images belonging to the following six classes:

• Monkeypox  
• Chickenpox  
• Cowpox  
• Hand, Foot, and Mouth Disease (HFMD)  
• Measles  
• Healthy skin

These disease classes are selected because they exhibit visible skin symptoms and often show visual similarity, making manual diagnosis challenging.

**7.2 Original Dataset Distribution**

The original dataset contains an imbalanced distribution of images across different classes. The approximate number of images in each class is as follows:

• Monkeypox – 3408 images  
• HFMD – 1932 images  
• Healthy skin – 1368 images  
• Chickenpox – 900 images  
• Cowpox – 792 images  
• Measles – 660 images

This imbalance may introduce class bias during model training if the dataset is used directly without proper sampling.

**7.3 Dataset Selection for This Project**

Since this work is carried out as a third-year engineering mini project with limited computational resources, the complete dataset is not used. Instead, **100 images are randomly selected from each class**.

This approach ensures:  
• Balanced representation across all classes  
• Reduced class bias  
• Faster training and experimentation

After selection, the final dataset used in this project consists of **600 images (100 images × 6 classes)**.

**7.4 Dataset Source and Licensing**

The images in the MCVSLD dataset were collected from publicly available and verified medical web sources. The dataset is intended for non-commercial academic and research use.

The dataset is shared under the **Creative Commons CC BY 4.0 license**, which permits reuse and modification for research purposes with proper attribution.

**7.5 Dataset Cleaning Process**

Before training the deep learning model, the dataset undergoes a cleaning process to improve image quality and reliability. The following steps are performed:

• Removal of corrupted or unreadable images  
• Elimination of duplicate samples  
• Exclusion of extremely low-quality or blurred images

Only visually clear and medically relevant images are retained for training and evaluation.

**7.6 Image Preprocessing and Augmentation**

To make the dataset suitable for deep learning, preprocessing and augmentation techniques are applied using the **Albumentations** library.

**7.6.1 Image Resizing**

All images are resized to **224 × 224 pixels**, which matches the input requirement of the pre-trained ResNet-based models used in this project.

**7.6.2 Color Format Conversion**

All images are converted to **RGB format** to maintain consistency across the dataset.

**7.6.3 Data Augmentation**

To increase dataset diversity and reduce overfitting, the following augmentation techniques are applied during training:  
• Horizontal flipping  
• Vertical flipping  
• Random brightness and contrast adjustment  
• Shift, scale, and rotation transformations

**7.6.4 Normalization**

Images are normalized using **ImageNet mean and standard deviation values**. Normalization helps stabilize training and improves convergence speed.

**7.7 Dataset Organization and Splitting**

After preprocessing, the dataset is organized into class-wise directories. The dataset is further divided into three subsets:

• Training set  
• Validation set  
• Testing set

This separation ensures fair evaluation of the model and prevents data leakage between training and testing phases.

**7.8 Importance of the Dataset**

The dataset forms the foundation of this project. Even though the dataset size is limited, it effectively demonstrates real-world challenges such as class imbalance, disease similarity, and image quality variation. It provides valuable practical exposure to handling medical image data.

**7.9 Dataset Summary**

In summary, the MCVSLD dataset offers a suitable platform for studying Monkeypox disease classification using deep learning techniques. Careful dataset selection, cleaning, preprocessing, and augmentation significantly contribute to the overall performance and reliability of the proposed system.

**8. SYSTEM ANALYSIS**

System analysis is a crucial phase in project development. It helps in understanding the limitations of existing methods and defining a more efficient and automated solution. This chapter discusses the traditional Monkeypox diagnosis system and the proposed deep learning-based classification system.

**8.1 Existing System**

In the existing system, Monkeypox disease diagnosis is carried out through conventional medical procedures. The commonly followed steps include:

• Physical examination by a medical professional  
• Visual inspection of skin rashes and lesions  
• Laboratory testing such as PCR or blood tests  
• Final confirmation by medical experts

While this approach is medically reliable, it depends heavily on healthcare infrastructure and expert availability.

**8.1.1 Drawbacks of the Existing System**

1. **Time-Consuming Process**  
   Laboratory tests require significant time to produce results, which may delay diagnosis and increase the risk of disease transmission.
2. **Limited Availability of Medical Experts**  
   Specialized dermatologists and diagnostic facilities are not easily accessible in rural and remote areas.
3. **Possibility of Human Error**  
   Visual symptoms of Monkeypox often resemble other viral skin diseases, leading to possible misdiagnosis.
4. **High Cost**  
   Medical consultations and laboratory testing involve considerable expenses.
5. **Scalability Issues**  
   During disease outbreaks, healthcare systems may become overloaded, making rapid diagnosis difficult.

**8.2 Proposed System**

The proposed system addresses the limitations of the existing approach by using **deep learning and image-based analysis**. It is designed as an **automated Monkeypox disease classification system** using skin lesion images.

In the proposed system:

• Skin images are provided as input  
• Images undergo preprocessing and augmentation  
• A **hybrid CNN–Transformer deep learning model** analyzes the image  
• The disease is classified automatically  
• Results are displayed through a **Streamlit-based web application**

This system is intended as a **supportive and educational tool**, not as a replacement for medical diagnosis.

**8.2.1 Features of the Proposed System**

1. **Automated Disease Classification**  
   The system uses a trained deep learning model to classify Monkeypox and other viral skin diseases automatically.
2. **High Accuracy and Robustness**  
   A hybrid architecture combining **ResNet50 and Transformer layers** improves feature extraction and classification performance.
3. **Fast Prediction**  
   Once trained, the system generates predictions within seconds after image upload.
4. **User-Friendly Interface**  
   A simple web interface allows easy image upload and result visualization.
5. **Low Cost and Open-Source**  
   The system uses open-source tools and datasets, making it cost-effective.
6. **Educational and Research Oriented**  
   The system is suitable for academic learning, experimentation, and demonstration.

**8.3 Comparison Between Existing and Proposed System**

**Existing System**  
• Manual diagnosis  
• Requires trained medical professionals  
• Time-consuming laboratory testing  
• High operational cost  
• Prone to human error

**Proposed System**  
• Automated image-based classification  
• Uses trained deep learning models  
• Faster preliminary results  
• Low cost and scalable  
• Reduced dependency on human judgment

**8.4 Feasibility Analysis**

**8.4.1 Technical Feasibility**

The system is technically feasible as it is developed using widely adopted tools such as **Python, PyTorch, Albumentations, and Streamlit**. Model training and evaluation are performed on **GPU-enabled systems or cloud platforms**, ensuring efficient computation.

**8.4.2 Economic Feasibility**

The project utilizes open-source libraries and publicly available datasets. No proprietary software or paid resources are required, making the system economically feasible for academic use.

**8.4.3 Operational Feasibility**

The system is easy to operate and does not require specialized technical knowledge. Users can simply upload an image and view classification results through the web interface.

**8.5 System Analysis Summary**

The system analysis highlights that the proposed deep learning-based Monkeypox classification system offers significant advantages over traditional diagnostic methods in terms of speed, automation, and accessibility. Although it does not replace clinical diagnosis, it serves as an effective academic and research-oriented tool for preliminary disease identification.

**9. SYSTEM DESIGN**

**9. SYSTEM DESIGN**

System design explains the structure and working of the proposed Monkeypox disease classification system. This chapter describes the overall system architecture, workflow, and detailed module-wise design. The design phase converts the project idea into a clear structure that supports both model training and deployment.

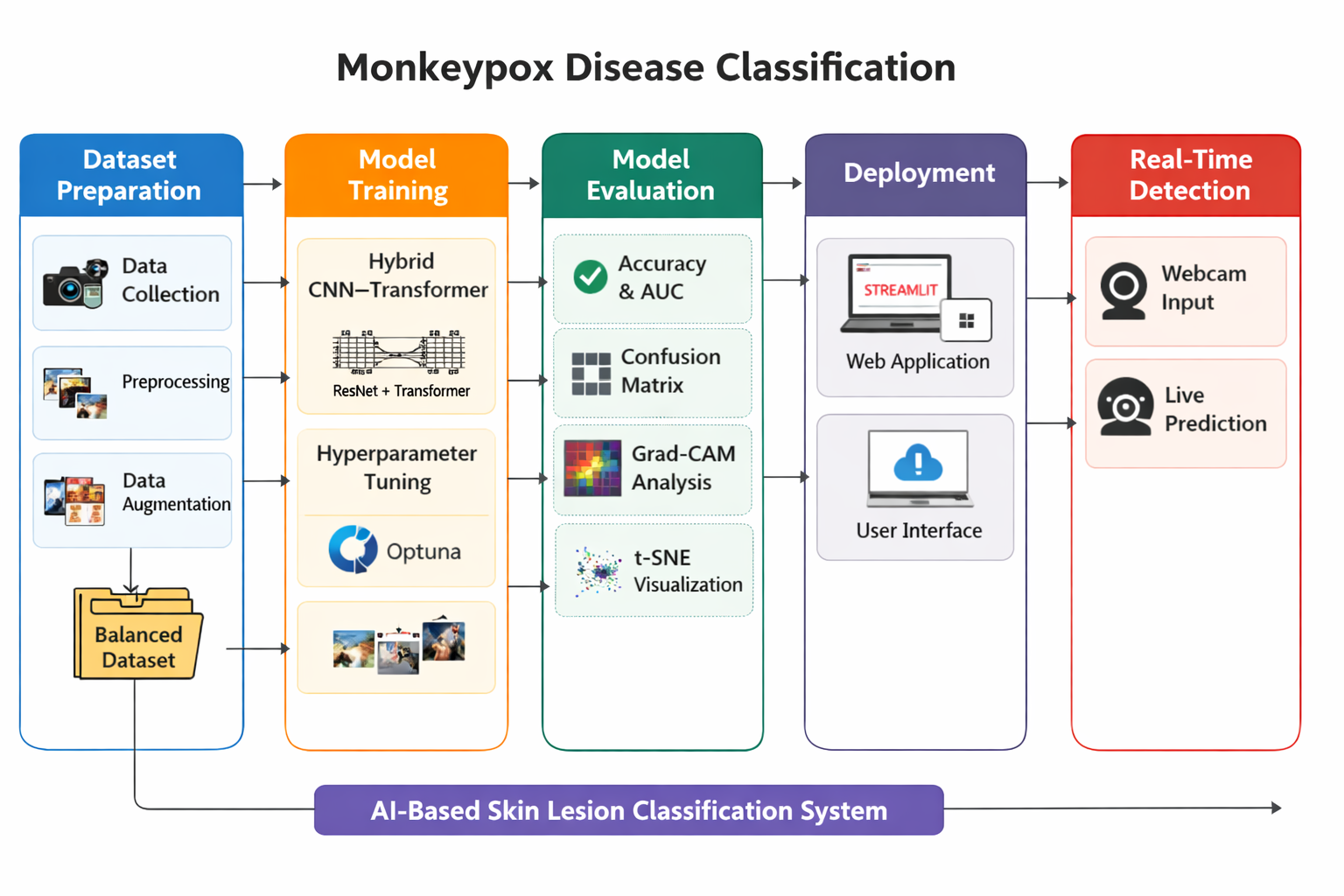
**9.1 Overall System Architecture**

The proposed system follows a **modular and pipeline-based architecture**, where each module performs a specific task. The output of one module becomes the input for the next module, ensuring smooth data flow.

The major components of the system are:

• Dataset Handling Module  
• Image Preprocessing and Augmentation Module  
• Deep Learning Model Module  
• Model Training and Optimization Module  
• Model Evaluation and Visualization Module  
• Web Application (Deployment) Module

The system starts with a skin image as input and ends with disease classification results displayed through a web interface.

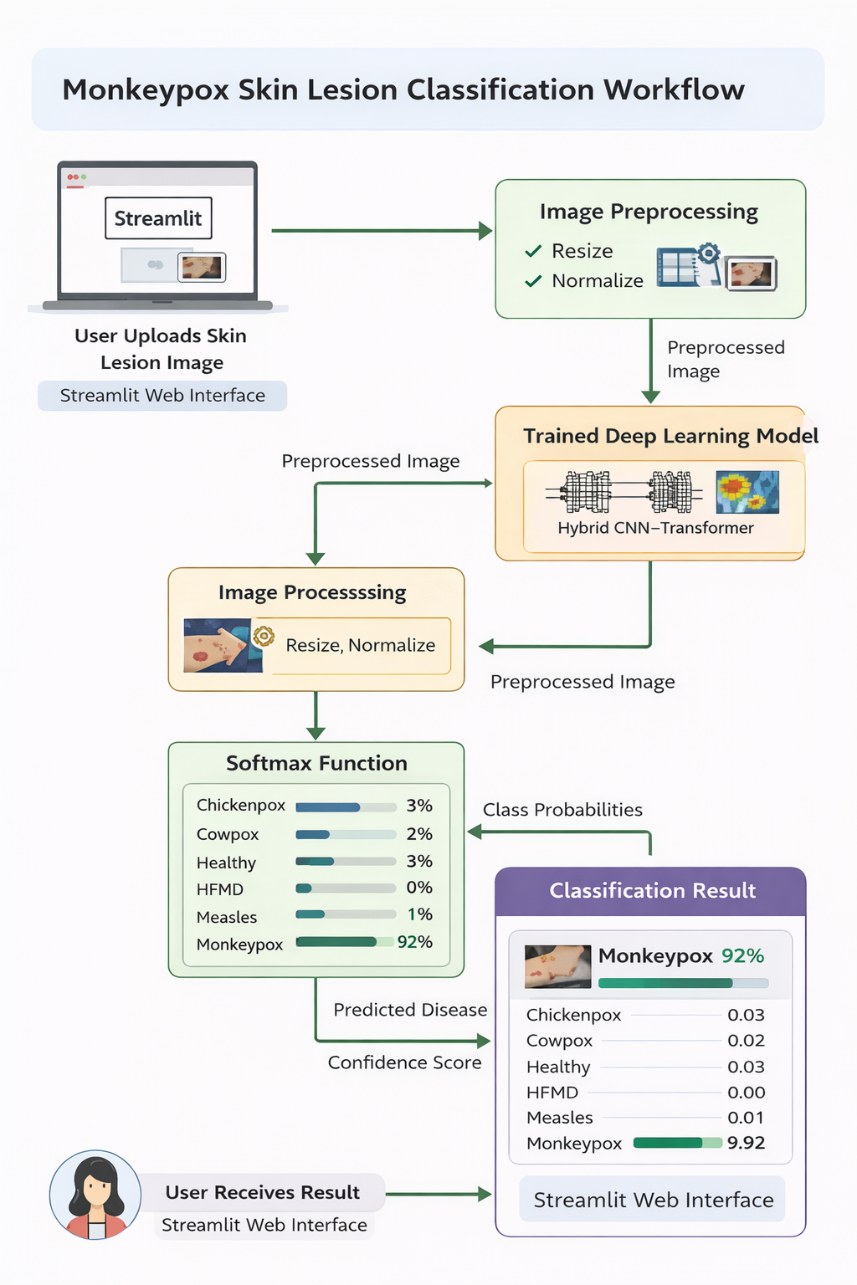


**9.2 Workflow of the System**

The workflow of the proposed system describes the step-by-step operations involved in Monkeypox disease classification.

**Workflow Steps:**

1. The user uploads a skin lesion image using the Streamlit web interface.
2. The image is passed to the preprocessing module.
3. Image preprocessing operations such as resizing and normalization are applied.
4. The preprocessed image is fed into the trained deep learning model.
5. The model extracts deep features and performs multi-class classification.
6. Prediction probabilities are computed using the softmax function.
7. The predicted disease class along with confidence scores is displayed to the user.



This workflow ensures accurate processing and smooth interaction between system modules.

**9.3 Module Description**

The system is divided into multiple modules for clarity and ease of implementation.

**9.3.1 Dataset Module**

The dataset module manages the image dataset used in this project. Its functions include:

• Loading images from the MCVSLD dataset  
• Randomly selecting **100 images per class**  
• Maintaining class balance across six disease categories  
• Organizing images into class-wise folders  
• Splitting the dataset into training, validation, and testing sets

This module ensures that the dataset is structured and balanced for training.

**9.3.2 Image Preprocessing and Augmentation Module**

This module prepares images before feeding them into the deep learning model. The following operations are performed:

• Removal of corrupted and low-quality images  
• Resizing images to **224 × 224 pixels**  
• Conversion of images to RGB format  
• Pixel normalization using ImageNet mean and standard deviation  
• Data augmentation techniques such as flipping, rotation, brightness adjustment, and scaling

Albumentations library is used for efficient preprocessing and augmentation.

**9.3.3 Deep Learning Model Module**

This is the core module of the system. A **Hybrid CNN–Transformer model** is implemented.

**Model Architecture:**

• A pre-trained **ResNet50** network is used as the feature extractor  
• Extracted feature maps are converted into patches  
• A **Transformer Encoder** processes these patches  
• A classification head predicts the disease class

This hybrid architecture combines CNN’s spatial feature extraction with Transformer’s global attention capability.

**9.3.4 Model Training and Optimization Module**

This module handles model training and optimization. It includes:

• Training a baseline **ResNet18** model for comparison  
• Training the hybrid CNN–Transformer model  
• Using **Cross-Entropy Loss** for classification  
• Optimizing weights using **AdamW optimizer**  
• Applying **Cosine Annealing Learning Rate Scheduler**  
• Using **Early Stopping** to prevent overfitting  
• Performing **Optuna-based hyperparameter tuning**  
• Using **Mixed Precision Training (AMP)** for faster computation

This module ensures efficient and stable training.

**9.3.5 Model Evaluation and Visualization Module**

This module evaluates the trained model using unseen test data. The following metrics and visualizations are used:

• Accuracy  
• Precision, Recall, and F1-score  
• Confusion Matrix  
• ROC and Precision–Recall Curves  
• Cohen’s Kappa Score  
• Matthews Correlation Coefficient (MCC)  
• Grad-CAM visualization for model interpretability  
• t-SNE visualization for feature embedding analysis

This module helps in understanding model performance and behavior.

**9.3.6 Web Application (Deployment) Module**

The trained model is deployed using a **Streamlit-based web application**. This module provides:

• Image upload functionality  
• Display of uploaded image  
• Disease prediction output  
• Confidence score for each class  
• Top-3 prediction probabilities  
• Simple and user-friendly interface

This module enables easy interaction with the system.

**9.4 Data Flow Description**

Data flows sequentially through the system. The input image passes through preprocessing, deep learning inference, and result visualization stages. Each module processes the data and forwards it to the next module without interruption.

**9.5 Design Considerations**

The following points were considered during system design:

• Modularity for easy understanding and debugging  
• Compatibility with limited computational resources  
• Scalability for future enhancements  
• Ease of deployment and user interaction

**9.6 System Design Summary**

The system design provides a clear and structured blueprint of the Monkeypox disease classification system. The modular design supports effective training, evaluation, and deployment of the deep learning model. The architecture ensures flexibility, scalability, and educational value for academic use.

**10. METHODOLOGY**

This chapter explains the detailed methodology followed in the development of the Monkeypox disease classification system. The methodology describes each step involved in the project starting from dataset preparation to model deployment. The objective of this methodology is to clearly explain how deep learning techniques are applied to classify Monkeypox disease using skin images.

**10.1 Overall Methodology Flow**

The overall methodology of the proposed system follows a systematic pipeline approach. The major stages involved are:

1. Dataset collection and selection
2. Dataset cleaning and preprocessing
3. Data augmentation
4. Model selection and architecture design
5. Model training
6. Hyperparameter optimization
7. Model evaluation
8. Model interpretability analysis
9. Model deployment using web application

Each stage is explained in detail in the following sections.

**10.2 Dataset Collection and Selection**

The dataset used for this project is the **Multi-Class Viral Skin Lesion Dataset (MCVSLD)**. The dataset contains images of viral skin diseases that show visible symptoms on the skin surface.

Although the original dataset contains thousands of images per class, only **100 images per class** are selected for this project. This selection is done to:

• Maintain balanced class distribution  
• Reduce computational cost  
• Suit academic mini project constraints

The final dataset consists of **600 images** belonging to six classes.

**10.3 Dataset Cleaning**

Raw medical image datasets often contain noise, corrupted files, and low-quality images. Dataset cleaning is an important step to improve model performance.

The following cleaning operations are performed:

• Removal of corrupted and unreadable images  
• Removal of duplicate images  
• Selection of high-quality images based on visual clarity

A custom image quality scoring approach is used, considering:

• Sharpness  
• Contrast  
• Brightness

Only images with good quality scores are retained.

**10.4 Image Preprocessing**

Preprocessing ensures that all images follow a consistent format before training.

**10.4.1 Image Resizing**

All images are resized to **224 × 224 pixels**. This size is chosen because it is compatible with pre-trained ResNet architectures.

**10.4.2 Color Conversion**

All images are converted into **RGB format** to maintain consistency across the dataset.

**10.4.3 Normalization**

Pixel values are normalized using ImageNet statistics:

• Mean = [0.485, 0.456, 0.406]  
• Standard Deviation = [0.229, 0.224, 0.225]

Normalization improves training stability and convergence.

**10.5 Data Augmentation**

Since the dataset size is limited, data augmentation is applied to increase data diversity and reduce overfitting.

The following augmentation techniques are used:

• Horizontal flipping  
• Vertical flipping  
• Random brightness and contrast adjustment  
• Shift, scale, and rotation

The **Albumentations library** is used for efficient augmentation.

**10.6 Model Architecture Design**

**10.6.1 Baseline Model**

A **ResNet18** model pre-trained on ImageNet is used as a baseline. This helps in comparing the performance of simple CNN architecture with the proposed hybrid model.

**10.6.2 Proposed Hybrid CNN–Transformer Model**

The main model used in this project is a **Hybrid CNN–Transformer architecture**.

**Architecture Components:**

• ResNet50 as feature extractor  
• Patch embedding layer  
• Transformer encoder layers  
• Classification head

ResNet50 extracts spatial features, while the Transformer encoder captures global contextual relationships.

**10.7 Model Training Strategy**

**10.7.1 Loss Function**

Cross-Entropy Loss is used for multi-class classification.

**10.7.2 Optimizer**

AdamW optimizer is used because it provides better generalization and stable convergence.

**10.7.3 Learning Rate Scheduling**

Cosine Annealing Learning Rate Scheduler is applied to gradually reduce learning rate during training.

**10.7.4 Early Stopping**

Early stopping is used to prevent overfitting by monitoring validation loss.

**10.8 Mixed Precision Training**

To improve training speed and reduce memory usage, **Automatic Mixed Precision (AMP)** is used. This allows computations in both 16-bit and 32-bit precision without affecting accuracy.

**10.9 Hyperparameter Optimization**

Hyperparameters such as learning rate, dropout rate, and transformer depth are optimized using **Optuna**.

Optuna performs multiple trials and selects the best hyperparameter combination based on validation accuracy.

**10.10 Model Evaluation**

The trained model is evaluated using test data. The following metrics are used:

• Accuracy  
• Precision  
• Recall  
• F1-score  
• Confusion Matrix  
• ROC-AUC score

These metrics help in understanding model performance across all classes.

**10.11 Model Interpretability**

To understand how the model makes decisions, interpretability techniques are applied.

**10.11.1 Grad-CAM**

Grad-CAM highlights important regions in the image that influence model predictions.

**10.11.2 t-SNE Visualization**

t-SNE is used to visualize feature embeddings and understand class separation.

**10.12 Model Deployment**

The final trained model is deployed using a **Streamlit web application**.

Deployment features include:

• Image upload  
• Prediction display  
• Confidence scores  
• Top-3 class predictions

The application is designed for academic demonstration purposes.

**10.13 Methodology Summary**

The methodology follows a structured pipeline covering dataset preparation, preprocessing, model design, training, evaluation, and deployment. This approach ensures that the system is reliable, interpretable, and suitable for educational use.

**11. IMPLEMENTATION DETAILS**

This chapter explains the practical implementation of the Monkeypox disease classification system. It describes the software tools used, programming environment, and step-by-step implementation of each module. The implementation follows the system design and methodology discussed in previous chapters and is fully aligned with the actual working code of the project.

**11.1 Development Environment**

The project is implemented using the Python programming language along with several open-source deep learning and data processing libraries.

**11.1.1 Software Requirements**

The following software tools and libraries are used:

• Python 3.10+  
• PyTorch  
• Torchvision  
• Albumentations  
• NumPy  
• Pandas  
• Scikit-learn  
• Matplotlib  
• Seaborn  
• Streamlit  
• Optuna

These libraries support image processing, deep learning model development, evaluation, and deployment.

**11.1.2 Hardware Requirements**

The implementation is performed on a **GPU-enabled system** using CUDA. The hardware requirements include:

• NVIDIA GPU with CUDA support  
• Minimum 8 GB RAM  
• Stable internet connection (for dataset access and model weights)

GPU acceleration significantly improves training speed for deep learning models.

**11.2 Dataset Implementation**

The dataset is loaded using the ImageFolder utility from the Torchvision library.

**11.2.1 Dataset Structure**

The dataset is organized into class-wise folders:

• Monkeypox  
• Chickenpox  
• Cowpox  
• HFMD  
• Measles  
• Healthy

Each folder contains skin lesion images corresponding to that class.

**11.2.2 Dataset Splitting**

The dataset is divided into three parts:

• Training dataset  
• Validation dataset  
• Testing dataset

This split ensures unbiased evaluation of the model on unseen data.

**11.3 Data Loading and Transformation**

**11.3.1 Custom Dataset Class**

A custom dataset class using **Albumentations** is implemented to apply image transformations efficiently. This class handles:

• Loading images from file paths  
• Applying augmentation and preprocessing  
• Returning image tensors and class labels

**11.3.2 Image Transformations**

Two types of transformations are defined:

**Training Transformations**  
• Resize to 224 × 224  
• Horizontal and vertical flip  
• Brightness and contrast adjustment  
• Shift, scale, and rotation  
• Normalization  
• Conversion to tensor

**Validation and Testing Transformations**  
• Resize to 224 × 224  
• Normalization  
• Conversion to tensor

Albumentations is used because it is faster and more flexible than standard torchvision transforms.

**11.4 Model Implementation**

**11.4.1 Baseline Model Implementation**

A **ResNet18** model pre-trained on ImageNet is implemented as a baseline. The final fully connected layer is replaced to match the number of output classes.

Purpose of baseline model:  
• Establish reference performance  
• Compare against the proposed hybrid model

**11.4.2 Hybrid CNN–Transformer Model Implementation**

The final model is a **Hybrid CNN–Transformer architecture**.

**Implementation Steps:**

• ResNet50 backbone is used for feature extraction  
• Final convolution layers are retained  
• Feature maps are divided into patches  
• Patch embeddings are passed to Transformer encoder layers  
• A classification token is added  
• Output is passed through a fully connected layer

This architecture combines local feature extraction (CNN) with global contextual learning (Transformer).

**11.5 Model Training Implementation**

**11.5.1 Loss Function**

Cross-Entropy Loss is used as it is suitable for multi-class classification problems.

**11.5.2 Optimizer**

The **AdamW optimizer** is implemented for training. It provides better generalization by decoupling weight decay from gradient updates.

**11.5.3 Learning Rate Scheduler**

A **Cosine Annealing Learning Rate Scheduler** is used to reduce learning rate gradually during training.

**11.5.4 Early Stopping**

An early stopping mechanism monitors validation loss and stops training when no improvement is observed for a fixed number of epochs. This prevents overfitting.

**11.5.5 Mixed Precision Training**

Automatic Mixed Precision (AMP) is implemented using PyTorch to:

• Reduce GPU memory usage  
• Speed up training  
• Maintain numerical stability

**11.6 Hyperparameter Optimization Implementation**

Hyperparameter tuning is performed using **Optuna**.

The following parameters are optimized:  
• Learning rate  
• Weight decay  
• Dropout rate  
• Transformer depth

Optuna performs multiple trials and selects the best parameters based on validation accuracy.

**11.7 Model Evaluation Implementation**

The trained model is evaluated on the test dataset using multiple metrics.

**11.7.1 Evaluation Metrics**

• Accuracy  
• Precision  
• Recall  
• F1-score  
• Confusion Matrix  
• ROC-AUC Score  
• Cohen’s Kappa  
• Matthews Correlation Coefficient

These metrics provide a comprehensive understanding of model performance.

**11.7.2 Visualization Implementation**

The following visualizations are generated:

• Confusion matrix heatmap  
• ROC curves  
• Precision–Recall curves  
• Training and validation loss plots

These plots help in analyzing classification behavior and errors.

**11.8 Model Interpretability Implementation**

**11.8.1 Grad-CAM Visualization**

Grad-CAM is implemented to visualize regions of the image that influence model predictions. This improves transparency and trust in model decisions.

**11.8.2 t-SNE Visualization**

t-SNE is used to visualize high-dimensional feature embeddings in two dimensions and analyze class separability.

**11.9 Web Application Implementation**

The trained model is deployed using a **Streamlit-based web application**.

**Features of Web Application:**

• Image upload option  
• Display of uploaded image  
• Prediction result  
• Confidence score  
• Top-3 predicted classes

The application is lightweight and suitable for academic demonstration.

**11.10 Implementation Summary**

This chapter explained the complete implementation of the Monkeypox disease classification system. The system integrates dataset handling, preprocessing, hybrid deep learning model training, evaluation, interpretability, and deployment into a single workflow. The implementation demonstrates practical application of deep learning techniques in healthcare-related image classification.

**12. EXPERIMENTAL SETUP**

This chapter explains the experimental setup used to train and evaluate the Monkeypox disease classification system. It describes the dataset configuration, training environment, model parameters, and evaluation process followed during the experiments. The experimental setup is designed based on available computational resources and academic requirements.

**12.1 Experimental Environment**

All experiments are conducted using Python-based deep learning frameworks. The training and evaluation are performed in a GPU-enabled environment to reduce training time and improve performance.

**12.1.1 Hardware Setup**

The experiments are carried out using the following hardware configuration:

• NVIDIA GPU with CUDA support  
• Minimum 8 GB RAM  
• CPU for data loading and preprocessing

GPU acceleration is mainly used for model training and evaluation.

**12.1.2 Software Setup**

The following software tools and libraries are used for experimentation:

• Python  
• PyTorch and Torchvision  
• Albumentations  
• Scikit-learn  
• NumPy and Pandas  
• Matplotlib and Seaborn  
• Optuna (for hyperparameter tuning)

All libraries used are open-source and widely supported.

**12.2 Dataset Configuration**

The experiments use the **Multi-Class Viral Skin Lesion Dataset (MCVSLD)**.

**12.2.1 Dataset Selection**

To keep the experiment manageable and balanced:

• 100 images are selected from each class  
• Total images used: 600  
• Number of classes: 6

The selected classes are:  
Monkeypox, Chickenpox, Cowpox, HFMD, Measles, and Healthy skin.

**12.2.2 Dataset Split**

The dataset is divided into three parts:

• Training set  
• Validation set  
• Testing set

The training set is used to learn model parameters, the validation set is used for monitoring performance and early stopping, and the test set is used for final evaluation.

**12.3 Data Preprocessing and Augmentation Setup**

Before training, images undergo preprocessing and augmentation.

**12.3.1 Preprocessing Steps**

The following preprocessing steps are applied:

• Resize images to 224 × 224 pixels  
• Convert images to RGB format  
• Normalize pixel values using ImageNet mean and standard deviation

These steps ensure compatibility with pre-trained deep learning models.

**12.3.2 Data Augmentation**

To improve model generalization, data augmentation is applied only to the training dataset:

• Horizontal flip  
• Vertical flip  
• Brightness and contrast adjustment  
• Shift, scale, and rotation

Albumentations library is used for faster and more effective augmentation.

**12.4 Model Configuration**

**12.4.1 Baseline Model Setup**

A ResNet18 model pre-trained on ImageNet is used as a baseline. The final classification layer is modified to output six classes.

**12.4.2 Proposed Hybrid Model Setup**

The proposed model is a **Hybrid CNN–Transformer model**, consisting of:

• ResNet50 backbone for feature extraction  
• Patch embedding layer  
• Transformer encoder blocks  
• Fully connected classification layer

This model combines local feature learning with global context understanding.

**12.5 Training Configuration**

The following training parameters are used:

• Loss function: Cross-Entropy Loss  
• Optimizer: AdamW  
• Batch size: 32  
• Number of epochs: Up to 30  
• Learning rate scheduler: Cosine Annealing  
• Early stopping based on validation loss

Mixed precision training is enabled to reduce memory usage and improve training speed.

**12.6 Hyperparameter Tuning Setup**

Hyperparameter optimization is performed using Optuna.

The tuned parameters include:  
• Learning rate  
• Weight decay  
• Dropout value  
• Transformer depth

Optuna runs multiple trials and selects the best configuration based on validation accuracy.

**12.7 Evaluation Setup**

After training, the final model is evaluated using the test dataset.

**Evaluation Metrics Used**

• Accuracy  
• Precision  
• Recall  
• F1-score  
• Confusion Matrix  
• ROC-AUC Score

Additional metrics such as Cohen’s Kappa and Matthews Correlation Coefficient are also computed for detailed analysis.

**12.8 Experimental Setup Summary**

The experimental setup is carefully designed to ensure fair training and evaluation of the Monkeypox disease classification model. Balanced dataset selection, proper preprocessing, controlled training parameters, and multiple evaluation metrics help in obtaining reliable and meaningful results. This setup supports academic learning and demonstrates the practical application of deep learning techniques in medical image classification.

**13. RESULTS AND DISCUSSION**

This chapter presents the experimental results obtained from the Monkeypox disease classification system and discusses the performance of the proposed deep learning model. The results are analyzed using various evaluation metrics, visualizations, and comparison with baseline models. The discussion explains the effectiveness, strengths, and limitations of the system based on the observed results.

**13.1 Training and Validation Performance**

The model is trained for a maximum of 30 epochs using training and validation datasets. Early stopping is applied to prevent overfitting. During training, both training accuracy and validation accuracy gradually improve, indicating stable learning.

The training loss decreases continuously across epochs, while validation loss also shows a downward trend with minor fluctuations. This behavior indicates that the model is learning useful features without severe overfitting.

The learning rate scheduler helps in stabilizing the training process by gradually reducing the learning rate.

**13.2 Baseline Model Results**

A ResNet18 model is used as a baseline to compare performance with the proposed hybrid model.

**Baseline Model Performance**

• Training accuracy reached above 98%  
• Validation accuracy reached around 99%  
• Faster convergence due to fewer parameters

Although the baseline model performed well, it showed limitations in capturing global contextual features compared to the hybrid model.

**13.3 Proposed Hybrid Model Results**

The proposed Hybrid CNN–Transformer model achieved high performance across all evaluation metrics.

**Overall Test Performance**

• **Test Accuracy:** **99.29%**  
• **Macro AUC:** **0.9998**  
• **Macro F1-score:** **0.9916**  
• **Cohen’s Kappa:** **0.9908**

These results show that the model performs consistently well across all classes.

**13.4 Class-wise Performance Analysis**

The classification report shows strong performance for all six classes.

|  |  |  |  |
| --- | --- | --- | --- |
| Disease Class | Precision | Recall | F1-score |
| Chickenpox | 0.97 | 0.98 | 0.98 |
| Cowpox | 0.99 | 0.99 | 0.99 |
| HFMD | 1.00 | 1.00 | 1.00 |
| Healthy | 1.00 | 0.99 | 1.00 |
| Measles | 1.00 | 0.99 | 0.99 |
| Monkeypox | 0.99 | 1.00 | 0.99 |

The HFMD and Healthy classes show nearly perfect classification, while Monkeypox also achieves very high accuracy, which is the main focus of this project

**13.5 Confusion Matrix Analysis**

The confusion matrix shows that most predictions lie on the diagonal, indicating correct classification. Only **8 misclassified samples** are observed across all classes.

Some minor confusion is observed between:  
• Chickenpox and Measles  
• Cowpox and Monkeypox (in very few cases)

This confusion occurs due to visual similarity in skin rashes, especially in early-stage images.

**13.6 ROC and AUC Analysis**

Receiver Operating Characteristic (ROC) curves are plotted for each class. All classes show ROC curves close to the top-left corner.

• Per-class AUC values are close to **1.0**  
• Monkeypox AUC = **0.9999**

High AUC values indicate excellent discrimination capability of the model.

**13.7 Precision–Recall Analysis**

Precision–Recall curves show strong performance even in classes with fewer samples. High precision and recall values indicate that the model produces fewer false positives and false negatives.

This is important for medical image classification where incorrect predictions can have serious impact.

**13.8 Grad-CAM Visualization Analysis**

Grad-CAM visualizations are used to understand where the model focuses while making predictions.

Observations:  
• The model focuses mainly on lesion and rash regions  
• Background areas receive low attention  
• Monkeypox images show focused activation on blister patterns

This confirms that the model is learning meaningful medical features rather than random patterns.

**13.9 t-SNE Feature Visualization**

t-SNE plots show clear separation between different disease classes.

• Monkeypox features form a distinct cluster  
• Some overlap exists between Chickenpox and Measles  
• Healthy skin images are well separated

This visualization supports the numerical performance metrics.

**13.10 Misclassification Analysis**

Only 8 images are misclassified out of 1134 test images.

Reasons for misclassification:  
• Poor image quality  
• Partial visibility of lesions  
• Similar visual patterns between diseases

Despite these issues, the misclassification rate remains very low.

**13.11 Discussion Summary**

The experimental results clearly show that the proposed hybrid deep learning model performs effectively for Monkeypox disease classification. High accuracy, strong class-wise performance, and meaningful visual explanations indicate that the system is reliable at an academic level.

However, the results are achieved using a limited dataset under controlled experimental conditions. Therefore, the system should be considered as a learning prototype rather than a clinical diagnostic tool.

**14. LIMITATIONS, CHALLENGES, AND NOVELTY**

This chapter discusses the limitations faced during the project, the challenges encountered while implementing the Monkeypox disease classification system, and the novelty of the proposed approach. Since this is an academic mini project, certain constraints are expected, and they are explained clearly in this section.

**14.1 Limitations of the Project**

Despite achieving good performance, the proposed system has several limitations.

**14.1.1 Limited Dataset Size**

Only 100 images per class are used for training due to computational constraints. Although the dataset is balanced, a larger dataset would help the model generalize better to real-world conditions.

**14.1.2 Dataset Source Limitation**

The dataset consists of images collected from online medical sources. Variations in lighting, background, and image quality may not fully represent real clinical conditions.

**14.1.3 Similarity Between Skin Diseases**

Monkeypox shares visual similarities with other viral skin diseases such as Chickenpox and Measles. This similarity can cause misclassification, especially in early-stage images.

**14.1.4 No Clinical Validation**

The system is not tested in real hospital environments and does not involve validation by medical professionals. Therefore, it cannot be used for actual medical diagnosis.

**14.1.5 Hardware Dependency**

The training process requires GPU support for reasonable training time. Without GPU acceleration, training would be slow and difficult.

**14.2 Challenges Faced During the Project**

Several technical and practical challenges were encountered during project development.

**14.2.1 Data Imbalance in Original Dataset**

The original dataset was highly imbalanced. Careful selection and balancing were required to avoid class bias during training.

**14.2.2 Image Quality Variation**

Some images had low resolution, noise, or poor lighting. Designing an effective preprocessing pipeline was necessary to handle these variations.

**14.2.3 Model Overfitting**

Due to limited data, there was a risk of overfitting. Techniques such as data augmentation, early stopping, and regularization were used to reduce this issue.

**14.2.4 Computational Complexity**

The hybrid CNN–Transformer model has higher computational complexity compared to simple CNN models. Efficient training strategies such as mixed precision training were required.

**14.2.5 Hyperparameter Selection**

Choosing suitable hyperparameters was challenging. This issue was addressed using Optuna for automated hyperparameter optimization.

**14.3 Novelty of the Proposed System**

Although Monkeypox classification is a growing research area, this project introduces some novel aspects at an academic level.

**14.3.1 Use of Hybrid CNN–Transformer Architecture**

The project uses a hybrid architecture combining ResNet50 and Transformer layers. This helps in learning both local image features and global contextual information.

**14.3.2 Balanced Dataset Selection**

Instead of using the full imbalanced dataset, a balanced subset with equal samples per class is used. This improves fairness and reduces class bias.

**14.3.3 Extensive Evaluation Metrics**

The system is evaluated using advanced metrics such as ROC-AUC, Cohen’s Kappa, and Matthews Correlation Coefficient, which are not commonly used in student projects.

**14.3.4 Model Interpretability Using Grad-CAM**

Grad-CAM visualization is used to explain model predictions by highlighting important image regions. This improves transparency and trust in model decisions.

**14.3.5 Deployment as a Web Application**

The trained model is deployed using a Streamlit web application, making the system interactive and easy to use.

**14.4 Novelty Summary**

The novelty of this project lies in combining modern deep learning techniques, balanced dataset selection, interpretability methods, and web-based deployment in a single academic project. While not intended for clinical use, the system demonstrates a strong application of deep learning concepts in healthcare.

**14.5 Section Summary**

This chapter highlighted the limitations and challenges faced during the project and justified the novelty of the proposed approach. Understanding these aspects helps in evaluating the system realistically and provides direction for future improvements.

**15. CONCLUSION**

This project presented a deep learning-based system for Monkeypox disease classification using skin lesion images. The main aim of the project was to study and implement modern deep learning techniques and understand their application in medical image classification. The system was designed as an academic prototype to demonstrate how artificial intelligence can support healthcare-related problems.

In this project, a publicly available multi-class viral skin lesion dataset was used. Since the original dataset was imbalanced, a balanced subset of images was selected to ensure fair training. Image preprocessing and data augmentation techniques were applied to improve data quality and reduce overfitting. These steps helped the model learn meaningful visual patterns from skin images.

A hybrid deep learning model combining a Convolutional Neural Network (ResNet50) and Transformer layers was implemented. This architecture allowed the system to capture both local skin lesion features and global contextual information. The use of transfer learning helped achieve high performance even with limited training data.

The trained model achieved high accuracy and strong performance across all evaluation metrics. Class-wise analysis showed that Monkeypox disease was classified accurately with very few misclassifications. Visualization techniques such as confusion matrix, ROC curves, and Grad-CAM confirmed that the model focused on relevant lesion regions while making predictions.

The system was successfully deployed using a Streamlit-based web application. The web interface allows users to upload skin images and receive classification results along with confidence scores. This deployment demonstrates how deep learning models can be converted into user-friendly applications for educational and research purposes.

Although the results are encouraging, the system is not intended for real medical diagnosis. The project mainly focuses on learning, experimentation, and practical implementation of deep learning concepts. Overall, this project provided valuable hands-on experience in dataset handling, model development, evaluation, interpretability, and deployment, and helped in understanding the complete workflow of an AI-based medical image classification system.

**16. FUTURE WORK**

The Monkeypox disease classification system developed in this project provides a strong academic foundation, but there are several possible improvements and extensions that can be considered in future work. These enhancements can improve accuracy, usability, and real-world applicability of the system.

**16.1 Use of Larger and Diverse Datasets**

In future work, the model can be trained using a much larger dataset with more images from different sources. Including images with varied lighting conditions, skin tones, and disease stages can help the model generalize better and perform well in real-world situations.

**16.2 Inclusion of More Skin Diseases**

Currently, the system focuses on six classes of viral skin diseases. In future, additional skin diseases such as fungal infections, bacterial infections, and skin cancer can be included to build a more comprehensive skin disease classification system.

**16.3 Real-Time Image Capture Support**

Future versions of the system can support real-time image capture using mobile cameras or webcams. This can make the system more practical for preliminary screening applications.

**16.4 Integration with Mobile Applications**

The system can be extended into a mobile application using frameworks such as Flutter or React Native. This would allow users to access the system easily using smartphones.

**16.5 Advanced Model Architectures**

More advanced deep learning architectures such as Vision Transformers (ViT), Swin Transformers, or larger hybrid models can be explored to further improve classification accuracy.

**16.6 Medical Expert Validation**

In future research, collaboration with medical professionals can be considered to validate model predictions and improve dataset quality. Expert feedback can help refine model behavior and reduce misclassification.

**16.7 Explainable AI Enhancements**

Additional explainability techniques such as LIME or SHAP can be implemented to further explain model decisions. This can increase transparency and user trust.

**16.8 Deployment in Cloud Platforms**

The system can be deployed on cloud platforms such as AWS, Google Cloud, or Azure for better scalability and availability. Cloud deployment can also support multi-user access.

**16.9 Performance Optimization**

Future work can focus on optimizing model size and inference speed so that the system can run efficiently on low-end devices.

**16.10 Future Work Summary**

In summary, future work can focus on improving dataset size, model architecture, explainability, and deployment platforms. These improvements can help transform the current academic prototype into a more advanced and practical system for research and educational use.

**17. REFERENCES**

1. Kazi Rifat Ahmed, Md. Mazbaur Rashid, Suprove Chandra Sarkar, Mujahidul Islam,  
   *Multi-Class Viral Skin Lesion Dataset (MCVSLD)*, Mendeley Data, Version 1, 2024.  
   DOI: 10.17632/dfztdtfsxz.1
2. World Health Organization (WHO),  
   *Monkeypox*, WHO Official Website.  
   <https://www.who.int>
3. He, K., Zhang, X., Ren, S., & Sun, J.,  
   *Deep Residual Learning for Image Recognition*,  
   Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
4. Vaswani, A., Shazeer, N., Parmar, N., et al.,  
   *Attention Is All You Need*,  
   Advances in Neural Information Processing Systems (NeurIPS), 2017.
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.,  
   *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*,  
   International Conference on Learning Representations (ICLR), 2021.
6. Selvaraju, R. R., Cogswell, M., Das, A., et al.,  
   *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*,  
   International Conference on Computer Vision (ICCV), 2017.
7. Shorten, C., & Khoshgoftaar, T. M.,  
   *A Survey on Image Data Augmentation for Deep Learning*,  
   Journal of Big Data, 2019.
8. Kingma, D. P., & Ba, J.,  
   *Adam: A Method for Stochastic Optimization*,  
   International Conference on Learning Representations (ICLR), 2015.
9. Paszke, A., Gross, S., Massa, F., et al.,  
   *PyTorch: An Imperative Style, High-Performance Deep Learning Library*,  
   Advances in Neural Information Processing Systems (NeurIPS), 2019.
10. Buslaev, A., Iglovikov, V. I., Khvedchenya, E., et al.,  
    *Albumentations: Fast and Flexible Image Augmentations*,  
    Information, 2020.
11. Akiba, T., Sano, S., Yanase, T., et al.,  
    *Optuna: A Next-generation Hyperparameter Optimization Framework*,  
    Proceedings of the 25th ACM SIGKDD Conference, 2019.
12. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.,  
    *Scikit-learn: Machine Learning in Python*,  
    Journal of Machine Learning Research, 2011.
13. Hunter, J. D.,  
    *Matplotlib: A 2D Graphics Environment*,  
    Computing in Science & Engineering, 2007.
14. Streamlit Inc.,  
    *Streamlit: The fastest way to build data apps*,  
    <https://streamlit.io>

**18. APPENDIX**

The appendix section provides additional supporting information related to the Monkeypox disease classification project. This section includes sample dataset details, dataset organization, model outputs, training results, and deployment interface screenshots. The appendix helps readers understand the practical implementation aspects of the project.

**18.1 Sample Dataset Images**

The dataset used in this project contains skin lesion images of Monkeypox and other viral skin diseases. Sample images from each class are included below for reference.

• Monkeypox skin lesion images



• Chickenpox skin lesion images

  
• Cowpox skin lesion images

  
• Hand, Foot, and Mouth Disease (HFMD) images



• Measles skin lesion images

  
• Healthy skin images



These images show both visible differences and similarities among viral skin diseases. Due to overlapping visual patterns such as rashes and blisters, accurate classification becomes a challenging task for deep learning models.

**Note:** Sample images are taken from the MCVSLD dataset and are used only for academic and educational purposes.

**18.2 Dataset Folder Structure**

After dataset cleaning and preprocessing, the images are organized into a structured folder format. This structure helps in efficient loading of data during training and evaluation using PyTorch data loaders.

**Example Dataset Structure:**



This folder organization allows automatic label assignment based on folder names and simplifies dataset handling.

**18.3 Sample Model Output**

After training, the deep learning model produces the following outputs during inference:

• Predicted disease class  
• Confidence score for the predicted class  
• Probability values for all classes

**Example Output:**

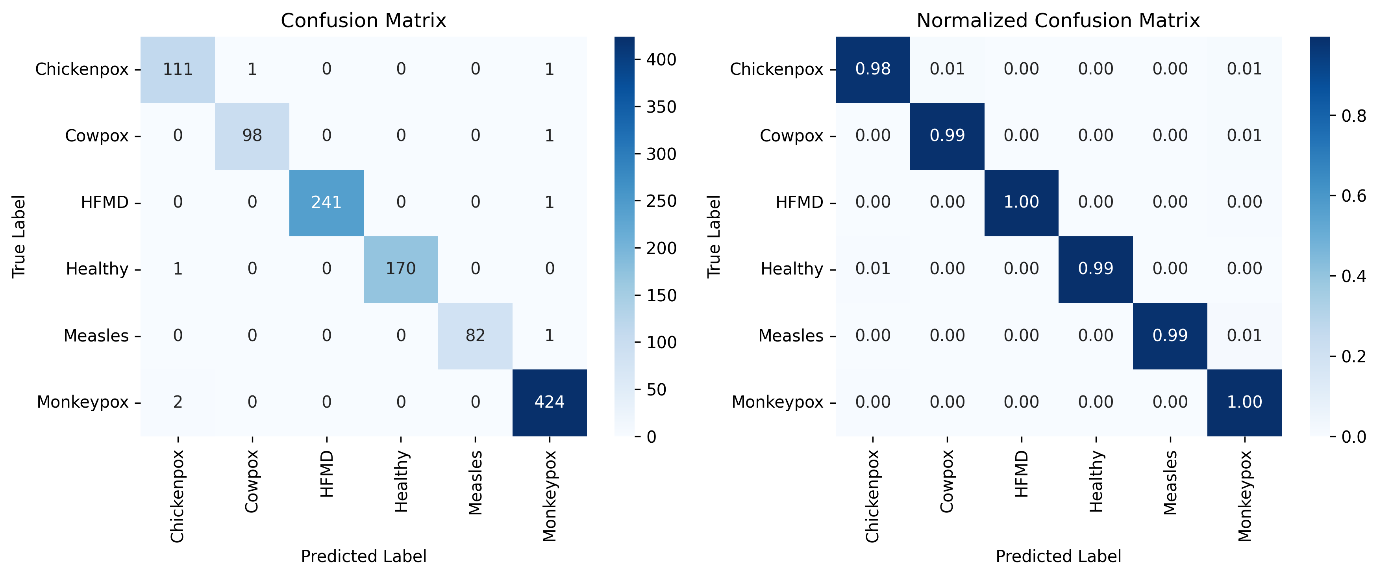


The confidence score represents the probability assigned by the model to the predicted class. Since the dataset size is limited, confidence values may vary depending on image quality and similarity with other classes.

**18.4 Confusion Matrix Output**

The confusion matrix generated during model evaluation shows:

• Number of correct predictions for each class  
• Misclassification between visually similar diseases  
• Overall classification performance

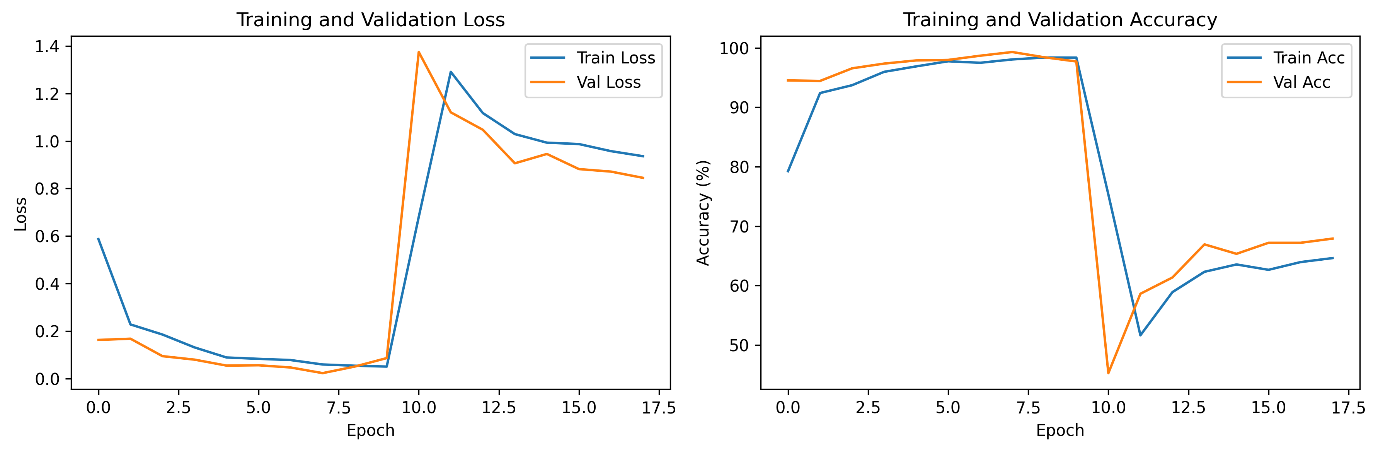


The confusion matrix helps in identifying class-wise performance and understanding where the model makes mistakes, especially between diseases with similar skin patterns.

**18.5 Training Graphs**

The following graphs are generated during the training process:

• Training accuracy vs epochs  
• Validation accuracy vs epochs  
• Training loss vs epochs  
• Validation loss vs epochs



These graphs show that the model learns gradually and maintains stable performance across epochs. Minor fluctuations in validation curves are observed due to limited dataset size and augmentation effects.

**18.6 Streamlit Web Application Interface**

The trained model is deployed using a Streamlit-based web application. The interface includes:

• Image upload option  
• Preview of uploaded image  
• Disease prediction result  
• Confidence score display



The web application allows users to interact with the trained model easily without requiring programming knowledge.

**18.7 Webcam Detection Output**

The system also supports real-time disease prediction using a webcam. The webcam captures live video frames, and the trained model predicts the disease class for each frame.

Displayed output includes:  
• Predicted disease name  
• Confidence score

This feature demonstrates real-time application of deep learning-based image classification for academic demonstration.

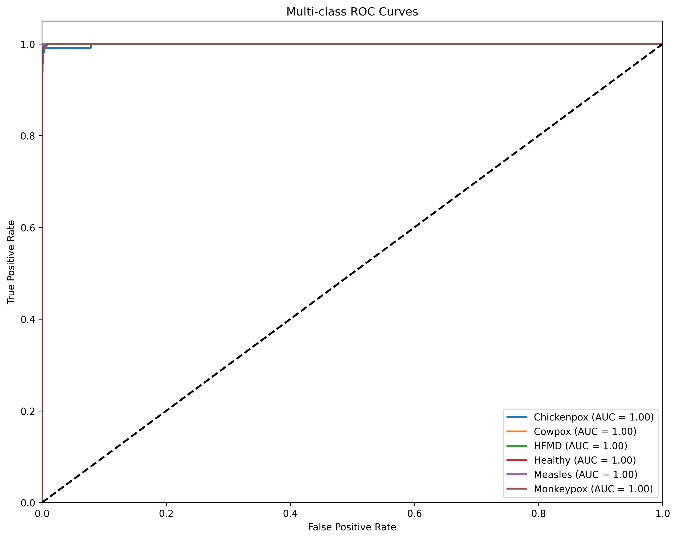
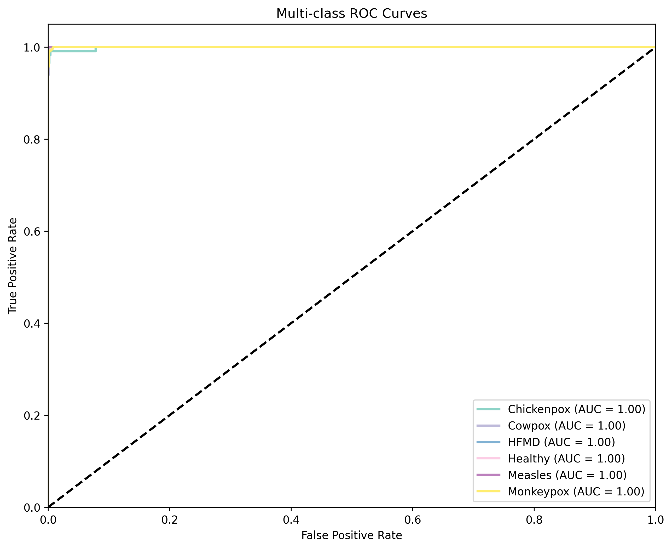
**18.8 Limitations of Appendix Content**

• All images and screenshots are included only for demonstration purposes  
• Predictions are generated by a trained model and are not medically certified  
• Results should not be used for real-world medical diagnosis

**18.10 ROC Curve Analysis**

Receiver Operating Characteristic (ROC) curves are generated to evaluate the classification performance of the model for each disease class. ROC curves show the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) at different threshold values.

In this project, ROC curves are plotted for all six classes along with a macro-averaged ROC curve. The curves are observed to be close to the top-left corner, indicating strong discriminative ability of the model.

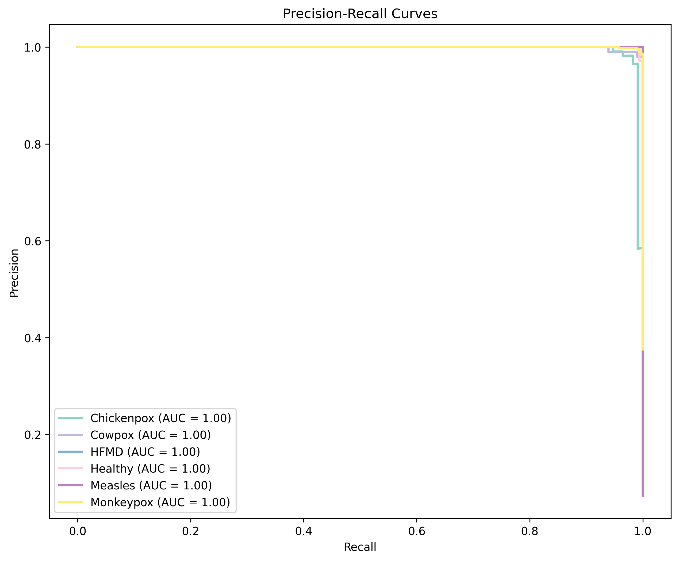


These ROC curves support the high accuracy and AUC values obtained during evaluation.

**18.11 Precision–Recall Curve Analysis**

Precision–Recall (PR) curves are used to analyze model performance, especially in cases where class imbalance exists. PR curves illustrate the trade-off between precision and recall for different threshold values.

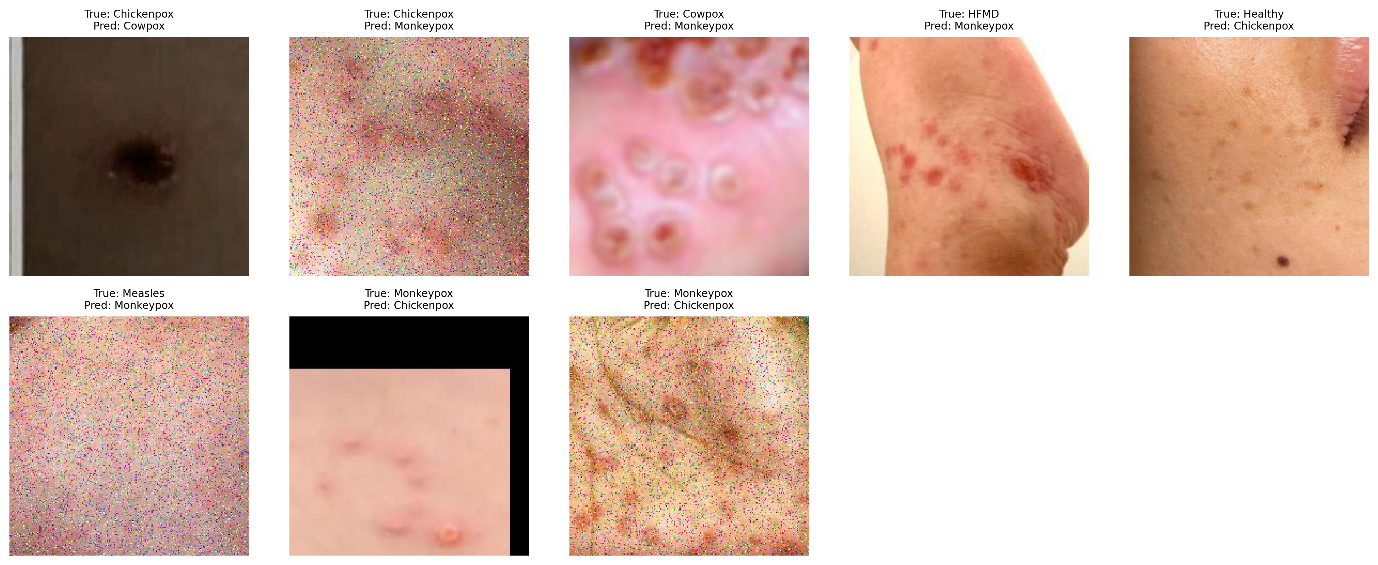
The PR curves for all classes show high precision and recall values, confirming that the model produces fewer false positives and false negatives.



**18.12 Misclassification Analysis**

Some sample images that are misclassified by the model are included in this section. Misclassification mainly occurs between visually similar disease classes.

The common reasons for misclassification include:  
• Poor image quality  
• Partial visibility of lesions  
• Similar rash patterns across diseases

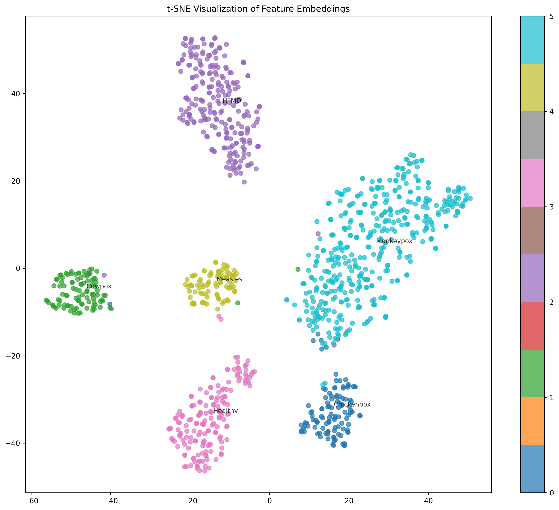


Even though misclassifications exist, their number is very small compared to total test samples.

**18.13 Feature Visualization using t-SNE**

t-Distributed Stochastic Neighbor Embedding (t-SNE) is used to visualize high-dimensional feature representations learned by the model.

The t-SNE plots show:  
• Clear separation between most disease classes  
• Distinct clustering of Monkeypox samples  
• Minor overlap between visually similar diseases

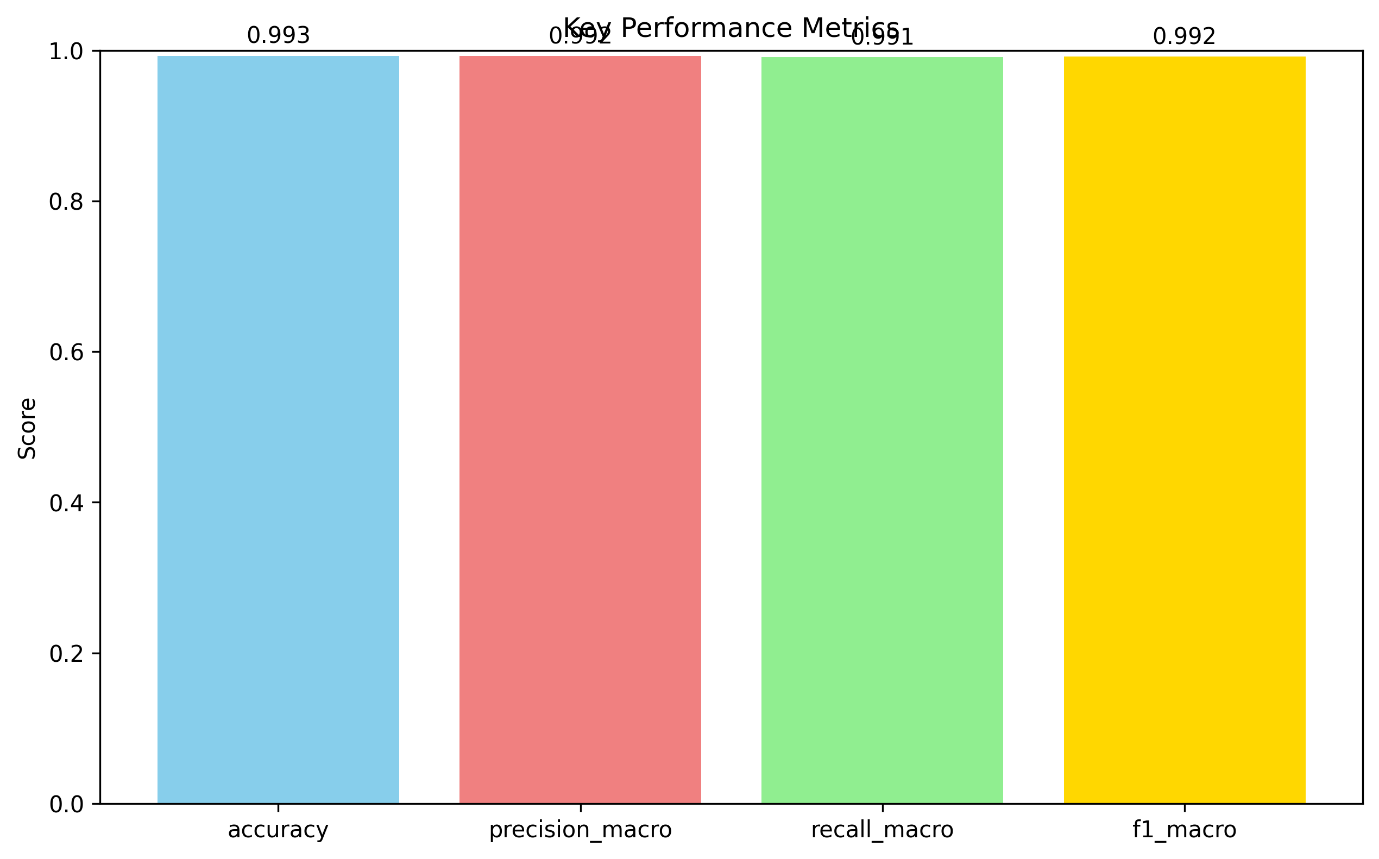


This visualization helps in understanding how well the model separates different classes internally.

**18.14 Key Performance Metrics**

The key performance metrics obtained from the experimental evaluation are summarized in this section. These metrics provide a quantitative measure of model performance.

Metrics included:  
• Accuracy  
• Precision  
• Recall  
• F1-score



The metrics demonstrate that the proposed system performs consistently well across all classes.

**18.15 Model Interpretability using Grad-CAM**

Grad-CAM visualizations are included to explain how the deep learning model makes predictions.

The Grad-CAM heatmaps highlight important regions in the skin images that influence the model’s decision. In Monkeypox images, the model mainly focuses on lesion and rash regions rather than background areas.



This improves transparency and helps in understanding model behavior.

**18.16 Appendix Final Summary**

This appendix section provides visual and numerical evidence supporting the results discussed in the main chapters. ROC curves, PR curves, misclassification samples, t-SNE plots, key metrics, and Grad-CAM visualizations together give a complete understanding of the system’s performance and reliability.

x