

A Deep Learning Framework with Metaheuristic Hyperparameter Optimization for Multi-Class Pox Skin Disease Classification

Noha El Menbawy^{1*}, Eman M. El-Gendy², and Mohamed M. Abdelsalam^{3,4}

¹Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt.

* Corresponding Author Email: nohahossam@mans.edu.eg - ORCID: 0000-0003-2351-3987

² Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt.

Email: eman_elgendy@mans.edu.eg - ORCID: 0000-0001-7468-4087

³Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt.

⁴ Faculty of Engineering, Mansoura National University, Gamasa, 35712, Egypt

Email: mohmoawad@mans.edu.eg - ORCID: 0000-0001-6792-9306

Article Info:

DOI: 10.22399/ijcesn.5435

Received : 10 May 2026

Revised : 10 July 2026

Accepted : 12 July 2026

Keywords

Monkeypox
Deep learning
Classification
Transfer learning
Vision Transformer

Abstract:

The precise and automated categorization of Monkeypox remains a challenging task due to its visual similarity to other infectious skin diseases, such as Chickenpox and Measles, as well as the limited availability of annotated medical datasets. This study suggests Conv2iT-BCAF, a novel hybrid framework that integrates parallel ConvNeXt and Vision Transformer (ViT) branches with Bidirectional cross-attention and Adaptive Fusion to acquire robust local and global representations. Furthermore, an image preprocessing technique, namely Self-Dynamic Hybrid Adaptive CLAHE and Adaptive Gamma correction (SD-HACAG), is developed to enhance the image quality. Moreover, the hyperparameters of Conv2iT-BCAF are optimized using the Black-necked Crane Optimizer (BNCO) metaheuristic algorithm, for finding the optimal model configuration. The experimental results on the MSID dataset show that the proposed Conv2iT-BCAF framework achieves an accuracy of 98.89%, precision of 98.94%, recall of 98.89%, specificity of 99.63%, AUC of 0.9972, and F1-score of 98.89% which is better than the existing approaches, and demonstrates its effectiveness for automated Monkeypox classification.

1 Introduction

Mpox (historically known as monkeypox) is a kind of infectious zoonotic sickness that happens by the Monkeypox virus (MPXV). It is a double-stranded DNA virus belonging to the family Poxviridae and genus Orthopoxvirus. In 1970, a 9-year-old child from the Democratic Republic of the Congo (DRC) was the first person in the world to be diagnosed with the disease; however, it was first discovered from captive cynomolgus monkeys in 1958 [1]. There are two main ways that MPXV may infect a human. Transmission is mostly zoonotic (animal-to-human), by direct contact with blood, bodily fluids, or skin lesions of infected animals, including

African rodents and wild primates [2]. Recent worldwide outbreaks have been attributed to person-to-person transmission through intimate contact and prolonged contact with respiratory droplets, infected skin lesions, body fluids, and contaminated objects such as clothes and bedding [3].

Although Mpox was previously endemic to regions of Central and West Africa, its rapid international spread has transformed it into a significant global health challenge. The widespread outbreak reported across multiple countries in 2022 led the World Health Organization (WHO) to issue its highest level of global health alert [4]. Skin lesions, rashes, and vesicular eruptions are characteristic symptoms of the condition, which can proceed in a specific pattern and cause serious illness and even death in

children and those with impaired immune systems [5]. Due to the quick spread of monkeypox in unaffected nations, there is an immediate need for scalable diagnostic technologies that can recognize and control the disease early.

Monkeypox is difficult to diagnose because it looks so much like other contagious skin disorders, including Chickenpox and Measles [6]. Clinical distinction is particularly challenging without laboratory confirmation because of the identical dermatological symptoms of all three disorders, which include extensive skin rashes and pustular lesions [7], [8]. In areas with limited resources, such as where Polymerase Chain Reaction (PCR) testing and laboratory equipment are restricted, misdiagnosis is common and can result in delayed treatment, incorrect isolation techniques, and increased community transmission [9]. Consequently, the initial diagnosis frequently relies on visual clinical assessment, which is inherently subjective, prone to human error, and further complicated by variations in skin tone and lesion progression [10], [11].

Deep Learning (DL) has been a popular tool for automating the examination of clinical images, which has allowed scientists to circumvent the limitations of manual diagnosis [12]. A large number of applications have taken use of Convolutional Neural Networks (CNNs) because of how well they extract localized, fine-grained spatial data, such the texture and margins of skin pustules [13], [14]. On the other hand, Vision Transformers (ViTs) have recently emerged as a promising approach due to their capacity to use self-attention processes to collect global context and long-range dependencies throughout the whole image [15], [16]. However, there are major limitations in applying these designs separately. Convolutional neural network approaches aren't always good at capturing global interactions across lesion locations, and transformer-based models usually need massive processing power and massive datasets [17]. In addition, the accuracy of the model is significantly diminished by the fact that raw clinical images taken with cellphones frequently have issues with uneven lighting, low contrast, and noise [18]. Lastly, deep learning models are extremely sensitive to their hyperparameter settings, which are frequently manually adjusted in an inefficient and computationally costly way [19]. The creation of a more efficient and robust framework that is capable of autonomously identifying optimal hyperparameter configurations and utilizing both local and global feature representations is motivated by these limitations. Moreover, the fluctuations in the illumination, contrast, image quality and lesion appearance

complicate the classification challenge, and hence demand excellent preprocessing techniques.

This study suggests a unique automated approach for classifying monkeypox skin diseases in order to overcome these difficulties. Initially, a Self-Dynamic Hybrid Adaptive CLAHE and Adaptive Gamma correction (SD-HACAG) preprocessing mechanism is implemented to enhance the quality of the image and increase the visibility of lesions. This is followed by the development of a hybrid deep learning architecture that captures both local and global discriminative features concurrently through the parallel integration of ConvNeXt and Vision Transformer branches. A Bidirectional Cross-Attention (BCA) module is employed to facilitate effective information exchange between the two feature streams, while an Adaptive Fusion (AF) mechanism dynamically combines complementary representations. Finally, the Black-necked crane Optimization (BNCO) algorithm is utilized to automatically optimize critical hyperparameters, enhancing classification performance and reducing manual tuning efforts. The following is a summary of the primary contributions of this paper:

- A novel SD-HACAG preprocessing mechanism is presented to increase image resolution to improve diagnostically significant lesion characteristics.
- A novel Conv²iT-BCAF framework is proposed, combining ConvNeXt and Vision Transformer branches in parallel and employing a Bidirectional Cross-Attention module to enhance local-global feature interaction, followed by an Adaptive Fusion mechanism to effectively integrate complementary representations.
- The automatic hyperparameter tuning is achieved by utilizing the BNCO metaheuristic optimization algorithm, which reduces the computational effort associated with manual optimization and enhances the robustness of the model.
- Extensive experiments conducted on the MSID dataset demonstrate the effectiveness of the proposed framework, achieving superior diagnostic accuracy and consistently outperforming competing approaches across multiple evaluation metrics.

The rest of this paper is organized as follows. Section 2 presents a review of related work. Section 3 details the proposed methodology, including the mathematical formulation of the SD-HACAG preprocessing mechanism, the architectural design of the hybrid Conv²iT-BCAF framework, and the BNCO-based hyperparameter optimization

process. A comprehensive analysis of the experimental outcomes, including ablation studies and comparisons with state-of-the-art models, is presented in Section 4. Lastly, Section 5 concludes the paper and delineates potential areas for future research studies.

2 Related Work

Research into diagnosing Mpox has recently centered on using Deep Learning (DL) methods, which reflects the rising significance of AI-based strategies in healthcare. The early study was mainly focused on traditional CNNs and transfer learning (TL) models. More recent studies have investigated transformer-based architecture, hybrid models, attention processes and optimization techniques to enhance diagnostic performance. This section outlines the most relevant studies on automated Mpox diagnosis and discusses their strengths, limitations, and key findings.

Azar et al. [20] proposed seven deep neural network models to identify Monkeypox from a dataset comprising skin images of Monkeypox, Chickenpox, Measles, and Normal cases, implementing both binary and four-class classification scenarios. Among the evaluated models, DenseNet201 achieved the best performance across all classification metrics in both scenarios. Specifically, it attained an accuracy of 97.63%, an F1-score of 90.51%, and an AUC of 94.27% in the binary classification task, while achieving an accuracy of 95.18%, an F1-score of 89.61%, and an AUC of 92.06% in the four-class classification task. Furthermore, LIME and Grad-CAM were employed to highlight lesion regions and improve the interpretability of the diagnostic decisions. Although the findings are impressive, repeatability and generalization are limited by the manual model selection procedure in the absence of automated hyperparameter adjustment. In order to develop a hybrid function learning model that can identify Monkeypox skin lesions, Altun et al. [21] optimized the hyperparameters of many transfer learning architectures, such as MobileNetV3-s, EfficientNetV2, ResNet50, VGG19, DenseNet121, and Xception. The optimized hybrid MobileNetV3-s model achieved the best performance, with an average F1-score of 0.98, AUC of 0.99, accuracy of 0.96, and recall of 0.97. Nevertheless, the investigation concentrated on binary classification, which limited its applicability to multiclass diagnostic environments.

Oztel [22] developed a hybrid Vision Transformer (ViT) and Convolutional Neural Network (CNN)-based classification framework to assist dermatologists with an objective tool for classifying seven skin lesion categories, including Monkeypox.

Experimental results showed that the model achieved an accuracy of 81.91%. Although this study was among the first to integrate ViT and CNN for Monkeypox-related skin lesion classification, it did not incorporate bidirectional cross-attention between feature streams or an adaptive fusion mechanism for dynamic feature weighting. Das et al. [23] suggested a hybrid deep learning framework combining Vision Transformers (ViTs) and Graph Convolutional Networks (GCNs) for effective Monkeypox diagnosis, reflecting the growing trend of integrating multiple complementary deep learning paradigms for improved skin lesion classification. Experimental results demonstrated that the proposed ViT+GCN model achieved an accuracy of 100% in binary classification and 97% in multi-class classification across two evaluated datasets. Despite its strong performance, the model introduces increased architectural complexity and does not incorporate an adaptive fusion mechanism to dynamically balance feature contributions from the two branches.

By merging lightweight convolutional neural network (CNN) architectures with a support vector machine (SVM) classifier optimized using Grid Search for parameter selection, Çetintaş [24] tackled the problem of unbalanced Monkeypox datasets. The proposed method achieved an accuracy of 98.67% on the Monkeypox Skin Images Dataset and 98.13% on the Monkeypox Skin Lesion Dataset. However, although Grid Search provides systematic optimization, it is computationally expensive and less efficient compared to metaheuristic-based approaches, particularly in large hyperparameter spaces. A hybrid CNN–Transformer architecture with lesion-focused fusion was introduced by Vadivel et al. [25] through MLF-Net for multiclass Mpox classification. The MSID dataset was used to achieve an accuracy of 93.68%. The model employs a relatively simple fusion strategy and does not integrate automated hyperparameter optimization, which may limit the effective interaction between local and global feature representations, despite its promising performance. Nazaimi et al. [26] suggested a deep learning-based detection method of Monkeypox by employing GoogLeNet architecture with transfer learning techniques on the MSID dataset. The model classified four classes namely Monkeypox, Chickenpox, Measles and Normal, with the total accuracy of 91.56%. However, the effectiveness of the technique is limited by a single CNN design to capture the global contextual information. Moreover, the model does not possess advanced feature fusion processes or automated hyperparameter tuning procedures, which may limit the model's classification performance for visually comparable skin conditions.

Chauhan et al. [27] proposed a hybrid deep learning and evolutionary feature-selection framework for

Monkeypox detection that combines DenseNet121 and MobileNetV2 for feature extraction, a Genetic Algorithm (GA) for feature selection, and a Random Forest classifier for final prediction. The proposed framework achieved classification accuracies of 92.71% on the MSID dataset and 97.77% on the MSLD dataset. However, the model relies solely on CNN-based feature extraction, which may limit its ability to capture global contextual relationships among lesion regions. Furthermore, feature fusion is performed through simple feature concatenation, potentially restricting effective interaction between complementary features and limiting overall classification performance. Al-Gaashani et al. [28] introduced a novel Multi-Scale Context Aggregation and Feature Pooling Network for Mpox classification based on MobileNetV2, incorporating a Multi-Scale Context Aggregator (MSCA) and a Feature Pooling Block to enhance feature representation. The proposed model achieved an accuracy of 96.15% on the MSID dataset. Despite its strong performance, the model relies on a fixed feature aggregation strategy

that may not effectively capture complex interdependencies among extracted features. Furthermore, the absence of automated hyperparameter optimization increases dependence on manual tuning and may prevent the model from reaching its optimal classification performance. Despite the promising results achieved in the existing studies, there are still many challenges. The limitations are mainly the use of manually tuned hyperparameters, sub-optimal feature fusion strategies, limited exploitation of complimentary local and global feature representations and performance degradation due to variations in the lesion appearance and image quality. These challenges highlight the importance of a more robust and adaptive framework that adopts efficient preprocessing, smart hyperparameter optimization, and sophisticated feature interaction mechanisms to improve the performance of the multiclass skin disease classification. Table 1 presents a comparative summary of the reviewed studies.

Table 1 Summary of recent studies on pox-related skin disease classification

Ref.	Year	Dataset	Method/ Architecture	Accuracy	Limitations
Azar et al. [20]	2023	MSID	7 DNNs (DenseNet201, VGG16, ResNet50, etc.) + Explainability (LIME + Grad-CAM)	• 97.63% (binary classification) • 95.18% (four-class classification)	Manual model selection, no automated hyperparameter optimization, no hybrid/transformer architecture
Altun et al. [21]	2023	Custom + MSLD	Hybrid CNN + Transfer Learning (MobileNetV3-s, EfficientNetV2, ResNet50, VGG19, DenseNet121, Xception) + Hyperparameter Optimization	• 96% (binary classification)	Restricted to binary classification only, reliance on a small MSLD dataset limits generalization and increases sensitivity to data imbalance.
Oztel [22]	2024	PAD-UFES-20 + MSLD combined dataset	ViT + DenseNet201	• 81.91% (seven-class classification)	Low quality mobile phone images considerably reduced classification accuracy, especially for certain lesion types
Das et al. [23]	2025	MSLD + MSID	Hybrid ViT + ResNet50 + AlexNet + Graph Convolutional Network (GCN)	• 100% binary classification • 97% multiclass classification	High architectural complexity due to GCN and potential overfitting on small datasets, limiting robustness and generalization.
Çetintaş [24]	2025	MSID + MSLD	Hybrid CNN (MobileNetV3Large + EfficientNetB0) + Optimized SVM with Grid Search	• 98.67% on MSID • 98.13% on MSLD	Grid search is computationally expensive compared to metaheuristic optimizers and SVM limits end-to-end learning
Vadivel et al. [25]	2026	MSID	Hybrid CNN–Transformer with Lesion-Focused Fusion, Spatial-Channel Attention, and Gated CNN–ViT Fusion	• 93.68% (four-class classification)	Manual hyperparameter tuning and limited adaptive interaction between local and global features
Nazaimi et al. [26]	2025	MSID	GoogLeNet with transfer learning	• 91.56% (four-class classification)	Single CNN architecture with limited global feature representation

Chauhan et al. [27]	2026	MSID+MSLD	DenseNet121 + MobileNetV2 + Genetic Algorithm (GA) feature selection + Random Forest classifier	• 92.71% on MSID • 97.77% on MSLD	Relies on CNN-based feature extraction and Genetic Algorithm feature selection, with no automated hyperparameter optimization or adaptive feature fusion.
Al-Gaashani et al. [28]	2026	MSID	MobileNetV2 with Multi-Scale Context Aggregation (MSCA) and Feature Pooling Network (FPN)	• 96.15% (four-class classification)	Manually tuned hyperparameters and fixed feature aggregation

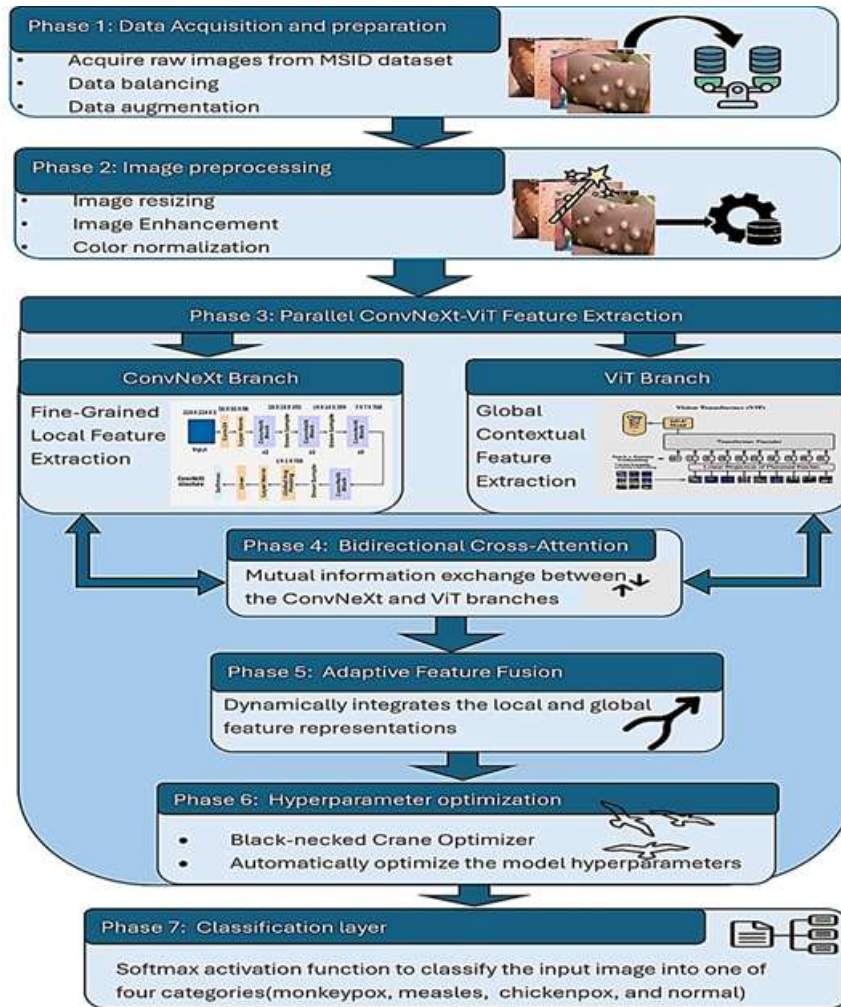


Figure 1 the proposed Conv²iT-BCAF Framework

3 Proposed Methodology

The suggested Conv²iT-BCAF framework for automated multiclass categorization of monkeypox disease is presented in this section. The framework is constructed to take advantage of both the global contextual information that Vision Transformers have acquired and the local spatial representations that convolutional neural networks have retrieved. Moreover, it introduces a novel image preprocessing strategy, an efficient bidirectional feature interaction mechanism, adaptive feature fusion, and an

intelligent hyperparameter optimization process to improve classification performance. Fig. 1 shows the comprehensive design and process flow of the Conv²iT-BCAF framework that has been suggested, which consists of seven main stages. In order to improve image contrast and increase lesion visibility while maintaining critical structural information, input images from the MSID dataset are initially subjected to Self-Dynamic Hybrid Adaptive CLAHE and Adaptive Gamma correction (SD-HACAG) preprocessing. Simultaneously, the enriched images are processed through two parallel feature extraction backbones: ConvNeXt, which extracts fine-grained local spatial features, and

Vision Transformer (ViT), which captures long-range global dependencies and semantic relationships. Subsequently, the Bidirectional Cross-Attention module performs mutual interaction between CNN-based and transformer-based representations to refine and enhance discriminative feature learning. This is followed by the integration of the complementary features into a more expressive and unified representation by an adaptive fusion module. In order to optimize hyperparameters and enhance model stability and performance, the BNCO-based metaheuristic optimization algorithm is implemented to enhance convergence and generalization capabilities. After optimization, the classification layer receives the optimized feature representation and generates the final prediction for four classes: monkeypox, measles, chickenpox, and normal. The remainder of this section is structured to supply the details of each component in the subsequent subsections.

3.1 Data Acquisition and Preparation

The dataset used for this work is the Monkeypox Skin Images Dataset (MSID), a publicly accessible multiclass benchmark proposed by Bala et al. and available at Kaggle [29]. The dataset includes 770 clinical skin images distributed across four diagnostic categories, consisting of 279 Monkeypox images, 107 Chickenpox images, 91 Measles images, and 293 Healthy skin images, with representative samples from each category illustrated in Fig. 2. Diseases that are often misdiagnosed because of their visual similarities were the primary focus of the sample collection, which included medical literature, dermatological atlases, and publicly available picture archives. To resolve the class imbalance in the MSID dataset, an augmentation based oversampling method was utilized to balance the distribution of all classes, as shown in Table 2, resulting in a balanced dataset of 1200 images across all classes. Overfitting is more likely to occur, and the model's generalizability is reduced due to the MSID dataset's limited size. Data augmentation was used to boost dataset variety and strengthen the suggested model to overcome this restriction. This is achieved by generating additional training samples using a range of geometric and intensity-based transformation techniques. The implemented data augmentation techniques include rotations of up to 20°, width and height shifts of up to 10%, zooming up to 15%, shear range up to 10%, and horizontal and vertical flips, with missing pixels filled using nearest-neighbor interpolation. To illustrate the data augmentation techniques in action, Fig. 3 shows both an original image from the dataset

and multiple augmented versions created using the specified data augmentation procedures.

3.2 Image Preprocessing

In real-world datasets, images of skin diseases often have big differences in brightness, contrast, skin tone, and acquisition circumstances. These issues are especially apparent in datasets of viral skin infections such as Monkeypox, Chickenpox, and Measles, where lesions may exhibit a variety of color, texture, and luminosity features. To address these challenges, we present a novel adaptive image preprocessing pipeline called SD-HACAG (Self-Dynamic Hybrid Adaptive CLAHE and Adaptive Gamma correction) designed to improve lesion-related visual features while maintaining the natural appearance of the image. Unlike traditional preprocessing methods that use fixed parameters, SD-HACAG adapts contrast and brightness based on each image's characteristics. Specifically, the luminance channel is first enhanced using adaptive CLAHE, where the clip limit is dynamically adjusted according to the image's global contrast to improve local lesion characteristics. This is followed by adaptive gamma correction, which adjusts the global brightness based on the mean luminance of the contrast enhanced channel, enhancing lesion visibility without causing over-saturation. By combining these steps, the pipeline improves lesion visibility while maintaining natural skin color and texture. This makes SD-HACAG well suited for preprocessing Monkeypox, Chickenpox, and Measles images, as it avoids over enhancement, maintains natural lesion characteristics, and offers high quality inputs for CNN-based classification models. The pipeline consists of five steps which are discussed in detail below.

3.2.1 Image resizing

The MSID dataset's images vary in size and resolution because they are acquired from various sources. Therefore, before applying any enhancement steps, each raw RGB image I_{raw} is resized to a fixed size of $(224 \times 224 \times 3)$ using bicubic interpolation in RGB color mode, defined as:

$$I_{RGB} = Resize(I_{raw}, 224 * 224 * 3) \quad (1)$$

3.2.2 Color space Transformation

The LAB color space separates lightness (L) from color information (a, b), which makes it possible to enhance contrast only in the intensity component without affecting the image colors. Consequently, each RGB image is converted to the LAB color space according to the equation below.



Figure 2 Representative sample images from each category in the MSID dataset

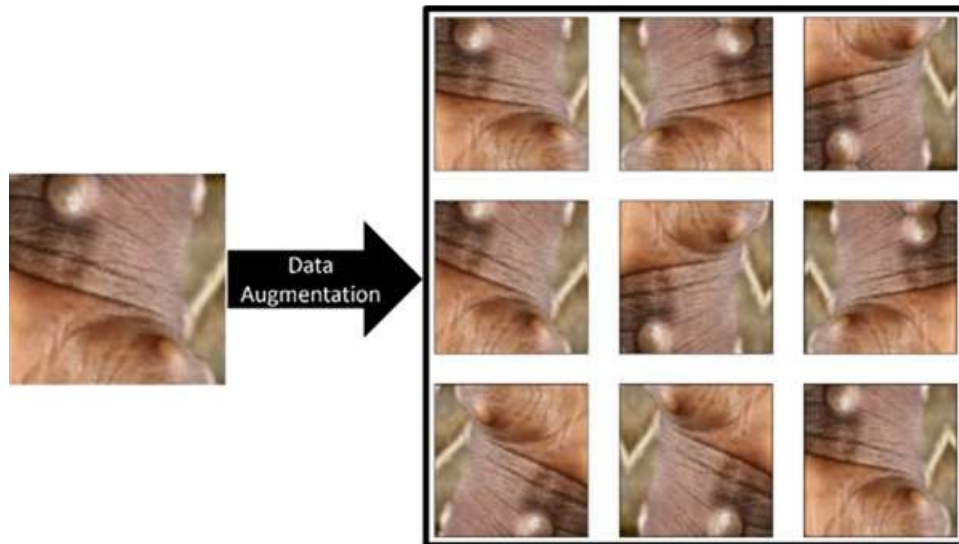


Figure 3 Sample images after data augmentation

Table 2 Distribution of MSID Dataset Images before and after applying the oversampling technique

Classes	chickenpox	Measles	Monkeypox	normal	Total
Data size before data balancing	107	91	279	293	770
Data size after data balancing	300	300	300	300	1200

$$I_{LAB} = (L, a, b) = RGB2LAB(I_{RGB}) \quad (2)$$

3.2.3 Adaptive CLAHE

For the luminance channel, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used to improve local contrast. In contrast to traditional CLAHE, SD-HACAG uses each image's global contrast to dynamically modify the clip limit. The overall contrast of the luminance channel is measured using its standard deviation, which helps guide the adaptive enhancement process. The

following equation calculates the contrast of the image.

$$C_{overall} = \sigma_L = \sqrt{\frac{1}{N} \sum_{x,y} (L(x,y) - \mu_L)^2}, \quad (3)$$

$$\mu_L = \frac{1}{N} \sum_{x,y} L(x,y)$$

Where N is the total number of pixels. To ensure numerical stability, the contrast value is normalized

and limited to the range $[0, 1]$ using a clipping function.

$$C_{norm} = \text{clip}\left(\frac{C_{overall}}{C_{max}}, 0, 1\right) \quad (4)$$

Where C_{max} is the maximum expected contrast, which is set to 64.

Next, the clip limit is dynamically adjusted based on the normalized contrast value. Images with low contrast are given higher clip limits to enhance details, while those with sufficient contrast receive lower clip limits to prevent over enhancement and reduce noise. The adaptive clip limit is then calculated using the equation below.

$$CL_{Adaptive} = CL_{max} - (CL_{max} - CL_{min}) * C_{norm} \quad (5)$$

Here, $CL_{Adaptive}$ represents the adaptive clip limit, while the maximum and minimum clip limits are denoted as CL_{max} and CL_{min} respectively. Subsequently, CLAHE is performed on the $L(x, y)$ channel with the adaptive clip limit, as described in the equation below.

$$L_{CLAHE} = \text{CLAHE}(L(x, y), CL_{Adaptive}) \quad (6)$$

3.2.4 Adaptive Gamma

Although Adaptive CLAHE enhances local contrast, it may not fully address global brightness variations. To tackle this, an adaptive gamma correction step is applied to adjust the overall luminance, which consists of three equations. First the mean intensity of the improved CLAHE $L_{CLAHE}(x, y)$ is calculated as:

$$\mu_{CLAHE} = \frac{1}{N} \sum_{x,y} L_{CLAHE}(x, y) \quad (7)$$

After that, an adaptive gamma parameter $\gamma_{Adaptive}$ is calculated based on the normalized mean luminance.

$$\gamma_{Adaptive} = \gamma_{max} - (\gamma_{max} - \gamma_{min}) * \frac{\mu_{CLAHE}}{255} \quad (8)$$

where the lower and upper bounds of gamma ranges are denoted by γ_{min} , γ_{max} respectively. Finally, the gamma corrected luminance channel $L_\gamma(x, y)$ is computed as follows.

$$L_\gamma(x, y) = 255 * \left(\frac{L_{CLAHE}(x, y)}{255}\right)^{1/\gamma_{Adaptive}} \quad (9)$$

This formulation allows the gamma parameter to adapt dynamically to the overall brightness of each image, improving illumination balance while preserving important lesion details.

3.2.5 Channel Reconstruction

Finally, the enhanced luminance channel merged with the original color channels and converted back to the RGB color space. Keeping the original color information helps SD-HACAG preserve important diagnostic cues, such as erythema, vesicles, and pigmentation patterns.

$$I_{final} = \text{LAB2RGB}(L_\gamma, a, b) \quad (10)$$

Fig. 4 shows representative samples from the MSID dataset before and after applying the proposed SD-HACAG preprocessing technique. As shown, the enhanced images display better contrast and clearer lesion boundaries compared to the original images, while maintaining the natural color of the skin. Overall, the adaptive CLAHE step effectively enhances local texture details, making lesion structures more distinct from the surrounding skin. Meanwhile, the adaptive gamma correction balances overall brightness and reduces illumination inconsistencies. Consequently, the proposed enhancement avoids over-saturation and artificial artifacts, resulting in visually realistic images.

3.3 Parallel ConvNeXt-ViT Feature Extraction

The suggested Conv²iT-BCAF architecture uses a parallel feature extraction technique that integrates the complementing qualities of ConvNeXt and the Vision Transformer (ViT) to efficiently capture the various features of skin lesions. The improved input image is processed concurrently by two separate branches rather than depending on a single backbone. Through hierarchical convolutional operations, the ConvNeXt branch concentrates on the extraction of fine-grained local texture patterns, lesion boundaries, and spatial structures. In contrast, the ViT branch employs self-attention mechanisms to model long-range dependencies and global contextual relationships. By using this supplementary learning approach, the network is able to acquire more detailed and discriminative feature representations prior to the next steps.

3.3.1 ConvNeXt Branch

ConvNeXt (Convolutional Network for the Next Generation) is a convolutional neural network architecture proposed by Liu et al. (2022) that updates the original ResNet design with certain



Figure 4 Representative samples from the MSID dataset before and after SD-HACAG preprocessing

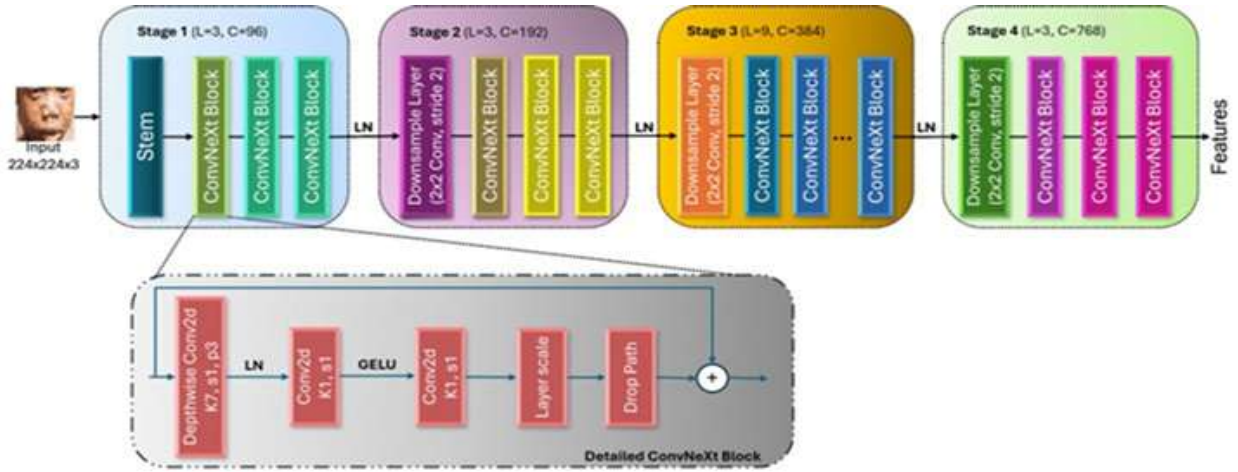


Figure 5 ConvNeXt-T Hierarchical Architecture Overview

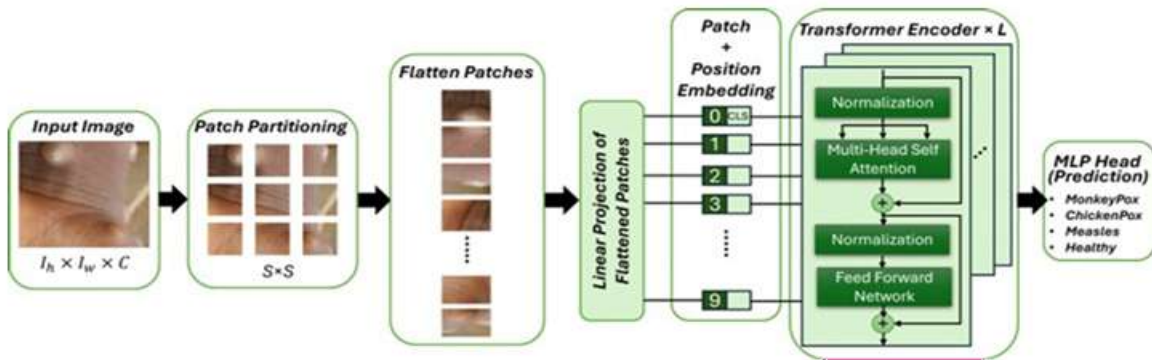


Figure 6 Vision Transformer Architecture

fundamental ideas from Vision Transformers while preserving the efficiency and inductive biases of CNNs [30]. The architecture uses GELU activation functions, an inverted bottleneck block design, depthwise separable convolutions with big 7x7 kernels, and Layer Normalization instead of Batch

Normalization. The local feature extraction backbone of the suggested framework is the ConvNeXt-Tiny (ConvNeXt-T) variation. With four hierarchical stages and a total of almost 28 million parameters, ConvNeXt-T is the lightest member of the ConvNeXt family. The block depths are 3, 3, 9,

3 and the channel dimensions are 96, 192, 384, 768, respectively. This compact architecture provides an efficient trade-off between the representational capacity and computational efficiency and is very suitable for medical image classification applications with little training data. The model was initialized with pretrained ImageNet weights, enabling the transfer of generalized visual representations to the domain of pox-related skin lesion classification. Fine-tuning is subsequently performed on the MSID dataset to adapt the pretrained representations to the specific visual characteristics of monkeypox, chickenpox, measles, and healthy skin images. The architecture of ConvNeXt-T is illustrated in Fig. 5.

3.3.2 Vision Transformer

The first use of the Transformer design was in natural language processing (NLP), where it showed outstanding performance by modeling long-range relations using the self-attention mechanism. Motivated by its success in NLP, Dosovitskiy et al. introduced the Vision Transformer (ViT) in 2021 as an alternative to traditional convolutional neural networks by employing the self-attention mechanism to sequences of image patches, as illustrated in Fig. 6. In order to enhance the ConvNeXt branch, the suggested Conv²iT-BCAF architecture incorporates the Vision Transformer (ViT) branch, which extracts global contextual representations from clinical skin pictures. Instead of extracting features using sliding convolutional filters, ViT first divides the input image into a sequence of fixed-size, non-overlapping patches. Each patch is then transformed into an embedding vector through a linear projection. To retain the spatial arrangement of the patches, positional embeddings are added, while a learnable classification token is appended to the sequence for image-level classification. Transformer encoder layers with residual connections and Layer Normalization process the subsequent series of patch embeddings. Each layer consists of a multi-head self-attention and feed-forward network sublayer [31]. The ViT architecture consists of two main components: a patch embedding layer and a stack of Transformer encoder blocks. In the initial stage of the patch embedding process, the input image of size $I_h \times I_w \times C$ is partitioned into a grid of M non-overlapping patches, each with spatial dimensions of $S \times S$ pixels. The total number of patches is calculated as:

$$M = \frac{I_h \times I_w}{S^2} \quad (11)$$

Where the image height and width denoted by I_h and I_w , respectively, C represents the number of

image channels, S is the patch size, and M is the total number of extracted patches. Each patch is flattened and linearly projected into a D -dimensional embedding vector. A learnable classification token (CLS) and positional embeddings are then added to the sequence to enable image-level classification while preserving the spatial arrangement of the patches. The resulting sequence of patch embeddings is processed by a stack of L Transformer encoder layers, each comprising a multi-head self-attention sublayer and a feed-forward network sublayer. The multi-head self-attention mechanism enables each patch to attend to all other patches, capturing long-range dependencies and global contextual information. Layer Normalization and residual connections are applied around each sublayer to improve training stability and facilitate efficient information flow. In the proposed framework, the input image is divided into 16×16 non-overlapping patches and processed through 8 Transformer encoder layers with 4 attention heads. Unlike the pretrained ConvNeXt-T branch, the ViT branch is trained from scratch on the MSID dataset to learn task-specific global representations. The contextualized patch embeddings produced by the Transformer encoder are directly passed to the proposed Bidirectional Cross-Attention module, where they interact with the local features extracted by the ConvNeXt branch.

3.4 Bidirectional Cross-Attention

After extracting complementary local and global features, the ConvNeXt and ViT branches forward their respective feature representations to the proposed Bidirectional Cross-Attention (BCA) module. The BCA module facilitates bidirectional information exchange between the two branches, enabling each branch to selectively enhance its feature representations by attending to the complementary information learned by the other branch. Specifically, the ConvNeXt features attend to the ViT features to incorporate global contextual information, while the ViT features simultaneously attend to the ConvNeXt features to enhance local structural awareness. Consequently, the fused features capture both fine-grained local details and rich global contextual information, leading to more discriminative representations for skin disease classification. The local feature sequence extracted by the ConvNeXt branch is denoted by $F_C \in \mathbb{R}^{N \times d}$, while the global feature sequence generated by ViT branch is represented by $F_V \in \mathbb{R}^{N \times d}$, where N denotes the number of feature tokens and d represents the embedding dimension. To enrich the local representations with global contextual information, the ConvNeXt features are used as the query Q_C ,

whereas the ViT features provide the key K_V and value V_V matrices. The corresponding linear projections are computed as:

$$Q_C = F_C W_Q, \quad (12)$$

$$K_V = F_V W_K, \quad (13)$$

$$V_V = F_V W_V \quad (14)$$

Where W_Q , W_K , W_V denote the learnable projection matrices. The refined ConvNeXt representation is then obtained using multi-head cross attention:

$$\hat{F}_C = \text{MultiHeadAttention}(Q_C, K_V, V_V) \quad (15)$$

Similarly, the ViT branch attends to the ConvNeXt feature sequence by treating the ViT representation as the query Q_V while the ConvNeXt features serve as the key K_C and value V_C matrices. The projections are computed as:

$$Q_V = F_V W_Q, \quad (16)$$

$$K_C = F_C W_K, \quad (17)$$

$$V_C = F_C W_V \quad (18)$$

The refined global representation is subsequently obtained as:

$$\hat{F}_V = \text{MultiHeadAttention}(Q_V, K_C, V_C) \quad (19)$$

Consequently, The Bidirectional Cross-Attention module outputs two refined feature representations, $\hat{F}_C$ and $\hat{F}_V$, each incorporating complementary local and global information learned from the opposite branch. These enhanced representations are then passed to the next phase, where they are adaptively integrated into a unified discriminative representation for final classification.

3.5 Adaptive Feature Fusion

After the Bidirectional Cross-Attention module refines the feature representations, the enhanced ConvNeXt and ViT features are forwarded to the proposed Adaptive Feature Fusion module, where they are adaptively combined to generate a unified feature representation. Rather than combining features using predefined fusion strategies, such as concatenation or fixed-weight averaging, the proposed Adaptive Feature Fusion module learns the contribution of each feature stream during training. By adaptively weighing the ConvNeXt and ViT representations, the module generates a unified feature representation that better captures the complementary strengths of both branches for pox-related skin disease classification.

The Adaptive Feature Fusion module introduces two scalar learnable parameters, α and β , which are initialized to equal values of 0.5 to ensure a balanced starting point with no initial bias toward either feature stream. Both parameters are declared as trainable weights, allowing the model to update them through backpropagation during training. To ensure that the learned importance weights form a valid probability distribution that sums to unity, a Softmax normalization is applied jointly over α and β before fusion:

$$[\hat{\alpha}, \hat{\beta}] = \text{Softmax}([\alpha, \beta]) \quad (20)$$

This Softmax normalization is a critical design choice that prevents the model from assigning arbitrarily large or negative weights to either feature stream, ensuring stable and interpretable fusion throughout training. The normalized weights $\hat{\alpha}$ and $\hat{\beta}$ directly reflect the learned relative importance of the ConvNeXt and ViT representations respectively, providing an implicit measure of which branch contributes more discriminative information for the classification task. The final fused feature representation, denoted as F_f , is obtained by computing a weighted linear combination of the two refined feature maps:

$$F_f = \hat{\alpha} \cdot \hat{F}_C + \hat{\beta} \cdot \hat{F}_V \quad (21)$$

Finally, the fused feature representation is subsequently passed through a fully connected (FC) layer, which performs a learnable linear projection to refine the fused features and preserve the desired embedding dimension before the final classification stage, the final adaptive feature fusion F_{AFF} is calculated as follows:

$$F_{AFF} = W \cdot F_f + b \quad (22)$$

Where W and b denote the learnable weight matrix and bias vector of the fully connected layer, respectively.

3.6 BNCO-Based Hyperparameter Optimization

The performance of deep learning models is highly dependent on the selection of appropriate hyperparameters. Key hyperparameters, including the batch size, dropout rate, unfreeze ratio (which determines the proportion of ConvNeXt layers fine-tuned), attention dimension of the Bidirectional Cross-Attention module, and optimizer, have a significant impact on the model's convergence, classification accuracy, and generalization performance. The search space of the optimized hyperparameters considered by the BNCO algorithm is presented in Table 3. Traditionally, these

hyperparameters are selected manually or optimized using exhaustive search methods, such as grid search, which are computationally expensive and may fail to identify the optimal configuration. To overcome these limitations, the proposed Conv²iT-BCAF framework employs the recently proposed Black-Necked Crane Optimization (BNCO) algorithm to automatically search for the optimal hyperparameter combination [32].

Table 3 Hyperparameter Search space used by BNCO

Hyperparameter	Candidate values	Description
Optimizer type	Adam, Nadam, RMSprop, Adagrad, SGD	Training optimizer
Batch size	8,16,32	Number of samples per batch
Dropout rate	0.2, 0.3, 0.4, 0.5, 0.6	Classifier dropout probability
Unfreeze ratio	0.1, 0.2, 0.3, 0.4, 0.5	Fraction of ConvNeXt layers fine-tuned
Attention dimension	64,128	Embedding dimension of the Bidirectional Cross-Attention module

BNCO is a population-based metaheuristic algorithm inspired by the natural foraging, migration, and survival behaviors of black-necked cranes. By effectively balancing global exploration and local exploitation, BNCO demonstrates rapid convergence while avoiding premature convergence toward local optima. Consequently, BNCO is well suited for searching the high-dimensional hyperparameter space of the proposed Conv²iT-BCAF architecture. The BNCO optimization process is positioned immediately before the network training stage. During each optimization iteration, BNCO generates a candidate hyperparameter configuration, trains the proposed Conv²iT-BCAF using the selected parameters, evaluates its classification performance on the validation dataset, and subsequently updates the search population according to the BNCO evolutionary strategy. This iterative process continues until the maximum number of optimizations iterations is reached or the convergence criterion is satisfied. The optimal hyperparameter set is then adopted for the final model training and testing.

3.6.1 Hyperparameter Encoding

Within BNCO, each search agent represents one complete hyperparameter configuration of Conv²iT-BCAF. Suppose that the optimization problem consists of d hyperparameters. The position of the i^{th} black-necked crane is represented as:

$$X_i = [x_{i1}, x_{i2}, \dots, x_{id}] \quad (23)$$

where each element corresponds to one optimized hyperparameter. According to the proposed framework, each solution is encoded as:

$$X_i = [BS, DR, UR, AD, OP] \quad (24)$$

Where, BS denotes the batch size, DR represents the dropout rate, UR is the ConvNeXt unfreeze ratio, AD denotes the attention dimension value, and OP indicates the optimizer type.

3.6.2 Population Initialization

Initially, BNCO randomly generates a population of candidate solutions within predefined lower and upper bounds. The initial position of each crane is determined by:

$$x_i^{(j)} = lb^{(j)} + r(ub^{(j)} - lb^{(j)}) \quad (25)$$

Where $lb^{(j)}$ and $ub^{(j)}$ denote the lower and upper limits of the i^{th} hyperparameter, respectively, while $r \in [0,1]$ is a uniformly distributed random number.

3.6.3 Fitness Function

The quality of each candidate solution is evaluated after training the proposed model using the corresponding hyperparameter set. Since the objective is to maximize the classification performance, the optimization process minimizes the following fitness function:

$$\text{Fitness} = 1 - \text{Accuracy}, \quad (26)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP (True Positives) denotes the number of correctly classified positive samples, TN (True Negatives) represents the correctly classified negative samples, FP (False Positives) indicates the negative samples incorrectly classified as positive, and FN (False Negatives) corresponds to the positive samples incorrectly classified as negative.

3.6.4 BNCO Optimization Mechanism

Following population initialization, BNCO updates each candidate solution using four biologically inspired behavioral phases.

1) Searching for Prey Phase

During the early optimization stage, BNCO performs extensive global exploration by updating

candidate solutions according to the following equation:

$$x_{i,new}^{(j)} = x_i^{(j)} + |x_{rand1} - x_{rand2}|r_1 \quad (27)$$

Where x_{rand1} and x_{rand2} denote two randomly selected individuals from the population and r_1 is a uniformly distributed random number. This mechanism increases population diversity and enables the optimizer to investigate unexplored regions of the hyperparameter space.

2) Pouncing on Prey Phase

As optimization proceeds, BNCO gradually shifts from exploration toward local exploitation by refining promising candidate solutions around the globally best solution according to

$$x_{i,new}^{(j)} = x_{best} + x_i^{(j)} \times CF \times WL(D) \quad (28)$$

Where x_{best} denotes the best solution obtained so far, CF is the adaptive control factor, and WL(D) represents the weighted levy flight function. This stage enables precise exploitation around highly promising hyperparameter configurations.

3) Migration Phase

To maintain population diversity and prevent premature convergence, BNCO periodically performs a migration operation inspired by the seasonal movement of black-necked cranes.

$$x_{i,new}^{(j)} = \begin{cases} C_1 & \text{if } fitness_i < fitness_{rand} \\ C_2, & \text{else} \end{cases} \quad (29)$$

$$C_1: x_i^{(j)} + (x_i^{(j)} - x_L^{(j)}) * C(0,1),$$

$$C_2: x_i^{(j)} + (x_L^{(j)} - 2 * \sin(r_1 + \pi/2) * x_i^{(j)}) * C(0,1).$$

Where x_L represents the current leader of the population, C(0,1) denotes the standard Cauchy distribution and $fitness_{rand}$ is the fitness value corresponding to the position of a randomly selected agent at the current iteration. This adaptive migration mechanism encourages broader exploration while preserving convergence stability.

4) Mutation phase

Finally, the mutation phase is applied primarily to poorly performing individuals, enabling them to escape local optima and explore previously unexplored regions of the search space. By periodically replacing or perturbing low-quality solutions, the mutation strategy preserves population diversity throughout the optimization process, thereby reducing the risk of premature convergence and improving the algorithm's ability to discover globally optimal hyperparameter configurations. The mutation operation is mathematically expressed as follows:

$$x_{i,new}^{(j)} = x_i^{(j)} + 0.1r_2(ub^{(j)} - lb^{(j)}) \quad (30)$$

Where r_2 is a uniformly distributed random number within the interval [0,1].

The overall BNCO optimization workflow is summarized in Algorithm 1, and the corresponding fitness evaluation process of the proposed Conv²iT-BCAF model is presented in Algorithm 2.

3.7 Classification Stage

After the optimization process converges, BNCO outputs the optimal hyperparameter configuration that minimizes the objective function while maximizing accuracy of the proposed model. These optimized hyperparameters are subsequently employed during the final training stage, enabling the proposed architecture to achieve improved convergence behavior, enhanced feature learning capability, and superior skin disease classification performance. The optimized feature representations produced by the Bidirectional Cross-Attention module are first integrated through the Adaptive Feature Fusion module to obtain a unified feature vector. This fused representation is then passed through a dense layer with ReLU activation, followed by batch normalization and dropout regularization, before being fed into the final Softmax classification layer. The classifier predicts one of the four diagnostic categories: Monkeypox, Chickenpox, Measles, or Healthy skin.

Algorithm 1: Pseudocode of BNCO-based Hyperparameter Optimization

- 1: **Input:** D= (X_{arr}, Y_{arr}) Preprocessed and balanced MSID dataset images and labels, (N) Population Size, (T_{max}) Maximum number of optimization operation, Hyperparameter Search Space Ranges: {(BS)batch size, (DR)dropout ratio, (UR)unfreeze ratio, (AD)attention dimension, (OP)optimizer type}.
- 2: **Output:** x_{best} the optimal configuration yielding the highest accuracy, F_{best} best fitness.
- 3: **begin**
- 4: Split the dataset D into Training, Validation, and Test sets.
- 5: Initialize the population of N cranes randomly. // $x_i = \{BS_i, DR_i, UR_i, AD_i, OP_i\}$, $i = 1, 2, \dots, N$
- 6: **For** each candidate x_i **do**. /*Evaluate each solution using fitness function*/

```

15:   | Scores.push ← ComputeFitness( $x_i$ )
16: End For
17: Select the best candidate solution  $x_{best}$ 
18: For  $t = 1$  to  $T_{max}$  do
19:   | Update BNCO Adaptive parameters
20:   For each crane  $x_i$  in population do
21:     | Generate random numbers
22:     If Exploration condition is satisfied then /*Global exploration phase*/
23:       |  $x_i^{new} \leftarrow$  update  $x_i$  using BNCO exploration Eq. (27).
24:     Else /*local exploitation phase*/
25:       |  $x_i^{new} \leftarrow$  update  $x_i$  using BNCO exploitation Eq. (28).
26:     End
27:     /*Boundary Control & Re-evaluation*/
28:     Clamp updated  $x_i$  values to ensure they do not exceed upper bound or fall below lower bound
29:     New_score ← ComputeFitness( $x_i^{new}$ )
30:     If new_score <  $scores_i$  then
31:       |  $x_i \leftarrow x_i^{new}$ 
32:       |  $scores_i \leftarrow$  new_score
33:     End
34:     If  $scores_i < scores_{rand}$ 
35:       |  $x_i^{new} \leftarrow$  update  $x_i$  using migration C1 in Eq.(29).
36:     Else
37:       |  $x_i^{new} \leftarrow$  update  $x_i$  using migration C2 in Eq.(29).
38:     End
39:     /*Boundary Control & Re-evaluation*/
40:     Clamp updated  $x_i$  values to ensure they do not exceed upper bound or fall below lower bound
41:     New_score ← ComputeFitness( $x_i^{new}$ )
42:     If new_score <  $scores_i$  then
43:       |  $x_i \leftarrow x_i^{new}$ 
44:       |  $scores_i \leftarrow$  new_score
45:     End
46:   End For
47:   /*Mutation stage*/
48:   Select the worst 20% of candidate solutions based on their fitness.
49:   Apply the BNCO mutation strategy (Eq. 30) to the selected candidates.
50:   Re-evaluate the mutated candidates using ComputeFitness( $x_i^{new}$ )
51:   Update  $x_{best}$  and  $F_{best}$  if a superior solution is obtained
52: End

```

Algorithm 2: Pseudocode of the Fitness Function

```

1: Function ComputeFitness( $X_i$ ):
2:   | Decode  $x_i$  into its corresponding hyperparameters
3:   | Build the proposed model. // using ConvNext, ViT, Bidirectional Cross-Attention, and adaptive feature fusion
4:   | Fine tune ConvNeXt using  $UR_i$ .
5:   | Set the Bidirectional Cross-Attention embedding dimension to  $AD_i$ 
6:   | Set the classifier dropout rate to  $DR_i$ 
7:   | Compile the model using optimizer  $OP_i$ 

```

8:	Train the model using batch size BS_i
9:	Compute the accuracy of the model
10:	Return (1- accuracy)
11:	End Function

4 Simulation results

A. Experimental setup

All experiments were conducted on the Kaggle Notebook cloud computing platform, which provides a preconfigured environment for deep learning development with GPU acceleration. The Monkeypox Skin Image Dataset (MSID) was obtained directly from the Kaggle repository and used throughout the training and evaluation stages. The proposed Conv²iT-BCAF framework was implemented in Python using the TensorFlow deep learning framework with the Keras high-level API, which facilitated model construction, training, and performance evaluation. The dataset was randomly divided into training, validation, and testing subsets using a 70%, 15%, and 15% split, respectively. To reduce the computational complexity of the BNCO-based hyperparameter optimization process, the maximum number of optimization iterations was set to 10, while each candidate solution was trained for only 5 epochs during fitness evaluation. This configuration provides a balance between optimization efficiency and computational cost, enabling BNCO to assess candidate solutions within a reasonable execution time. After identifying the optimal hyperparameter combination, the proposed model was retrained using the selected hyperparameters for 50 epochs to fully exploit its learning capability and obtain the final classification performance. The experimental configuration of the proposed framework is summarized in Table 4.

Table 4 Experimental Configuration Of The Proposed Conv²iT-BCAF Framework

Key	Value
Dataset	MSID
Image size	(224 × 224 × 3)
Categories	“monkeypox”, “chickenpox”, “measles” and “normal”
Split ratio	70:15:15 (train:validation:test)
Working environment	Kaggle(using its GPU)
Scripting language	python
Apply Data augmentation	yes
Rotation	±20°
Width shift	±10%
Height shift	±10%
Zoom	15%
Shear	10%

Horizontal flip	Enabled
Vertical flip	Enabled
Fill mode	Nearest
Training epochs	50
BNCO population size	5
BNCO maximum number of iterations	10
Training epochs per BNCO evaluation	5
Output activation function	Softmax
Hyperparameter search ranges	Table 3
Solution dimension	5

B. Results and Discussion

To evaluate the effectiveness of the BNCO algorithm in optimizing the proposed model hyperparameters, the optimization process was monitored over 10 iterations. Table 5 summarizes the selected hyperparameter configurations and the corresponding accuracies obtained at each iteration. As optimization progressed, BNCO consistently identified increasingly effective hyperparameter combinations, leading to a steady improvement in model accuracy. The optimization began with an accuracy of 0.72 at the first iteration using the RMSprop optimizer with a batch size of 8. The accuracy gradually increased as BNCO explored the search space and balanced exploration with exploitation. At iteration 7, the algorithm identified the optimal hyperparameter configuration, consisting of the Adam optimizer, a batch size of 16, a dropout rate of 0.3, an unfreeze ratio of 0.4, and an attention dimension of 128, achieving a validation accuracy of 0.89. This configuration remained unchanged during the remaining iterations, indicating that the algorithm had successfully converged to a stable solution. The convergence at iteration 7, well before reaching the maximum of 10 iterations, demonstrates the ability of BNCO to efficiently navigate the hyperparameter search space while reducing the computational cost associated with exhaustive search methods. After identifying

the optimal hyperparameter configuration using the BNCO algorithm, the proposed Conv²iT-BCAF model was retrained for 50 epochs using the selected hyperparameters. Fig. 7 presents the training and validation accuracy curves of the proposed Conv²iT-BCAF model using the optimal hyperparameters selected by the BNCO algorithm. Both curves increase rapidly during the initial training epochs before gradually stabilizing, indicating effective learning and successful convergence of the model. Training was terminated after 35 epochs using the EarlyStopping criterion, as no further improvement in the validation metric was observed. The close agreement between the training and validation accuracy curves indicates good generalization and minimal overfitting. Likewise, Fig. 8 shows that both the training and validation loss decreased steadily throughout the training process and eventually converged to stable values, confirming the effectiveness of the optimization process. The absence of a noticeable increase in the validation loss further indicates that the proposed model maintained stable learning without overfitting. Using the optimal hyperparameter configuration selected by BNCO, the proposed Conv²iT-BCAF model achieved a final test accuracy of 98.89%, demonstrating its effectiveness for pox-related skin disease classification. To further evaluate the effectiveness of the proposed Conv²iT-BCAF framework, its performance was compared with eight widely used deep learning models under identical experimental conditions. All models were trained and evaluated using the same dataset partition, preprocessing strategy, and evaluation protocol to ensure a fair comparison. Table 6 summarizes the classification results in terms of accuracy, precision, recall, F1-score, specificity, AUC, and Dice coefficient (DCI). As shown, the proposed model consistently achieved the best performance across all evaluation metrics, attaining a classification accuracy of 98.89%, an F1-score of 98.89%, a precision of 98.94%, and a recall of 98.89%. Furthermore, the model achieved a specificity of 99.63%, an AUC of 0.9972, and a Dice coefficient of 98.89%, demonstrating excellent discriminative capability and reliable classification performance. To further investigate the influence of the CNN backbone on the proposed framework, additional experiments were conducted by replacing the ConvNeXt branch with three widely used pretrained CNN architectures: VGG19, ResNet50, and EfficientNetB0. In all experiments, the Vision Transformer branch, Bidirectional Cross-Attention (BCA) module, Adaptive Feature Fusion (AFF)

module, and training configuration were kept unchanged to ensure a fair comparison. As summarized in Table 7, the ConvNeXt backbone consistently achieved the best performance across all evaluation metrics, outperforming the alternative CNN backbones. These results demonstrate that ConvNeXt provides more discriminative local feature representations, which effectively complement the global contextual information extracted by the Vision Transformer, leading to superior classification performance within the proposed framework. Moreover, the confusion matrices presented in Fig. 8 further support these findings by demonstrating that the proposed ConvNeXt-based framework produced the highest number of correctly classified samples and the fewest misclassifications among all evaluated backbone configurations. To further evaluate the impact of both the fine-tuning strategy and data enhancement, three ConvNeXt training configurations were investigated: freezing all ConvNeXt layers, unfreezing all layers, and partial fine-tuning with an unfreeze ratio of 0.4, which was automatically selected by the BNCO algorithm. Fig. 9 compares the classification accuracy achieved by each fine-tuning strategy using the original and enhanced datasets. As illustrated, the enhanced dataset consistently improved the classification performance across all three fine-tuning strategies, highlighting the effectiveness of the proposed preprocessing technique. Among the evaluated configurations, the BNCO-selected partial fine-tuning strategy achieved the highest accuracy on both datasets, outperforming the freeze all and unfreeze all approaches. Freezing all ConvNeXt layers limited the model's ability to adapt the pretrained features to the target domain, whereas unfreezing all layers increased the risk of overfitting. In contrast, partially unfreezing 40% of the ConvNeXt layers provided an effective balance between preserving pretrained knowledge and adapting the model to the characteristics of pox-related skin images. These results further demonstrate the capability of the BNCO algorithm to identify an effective fine-tuning strategy while also confirming the positive contribution of the enhanced dataset to the overall performance of the proposed framework.

C. Ablation study

To further validate the effectiveness of the proposed architectural components, an ablation study was conducted by progressively enhancing the proposed

Table 5 Hyperparameter optimization results obtained by BNCO algorithm over 10 iterations

Iteration	Optimizer	Batch size	Dropout	Unfreeze ratio	Attention-Dimension	Accuracy
-----------	-----------	------------	---------	----------------	---------------------	----------

1	RmsProp	8	0.2	0.2	128	0.72
2	RmsProp	16	0.3	0.3	128	0.77
3	Nadam	8	0.2	0.4	128	0.78
4	RMSprop	16	0.3	0.4	128	0.8
5	RMSprop	16	0.3	0.4	128	0.8
6	RMSprop	16	0.3	0.4	128	0.8
7	Adam	16	0.3	0.4	128	0.89
8	Adam	16	0.3	0.4	128	0.89
9	Adam	16	0.3	0.4	128	0.89
10	Adam	16	0.3	0.4	128	0.89

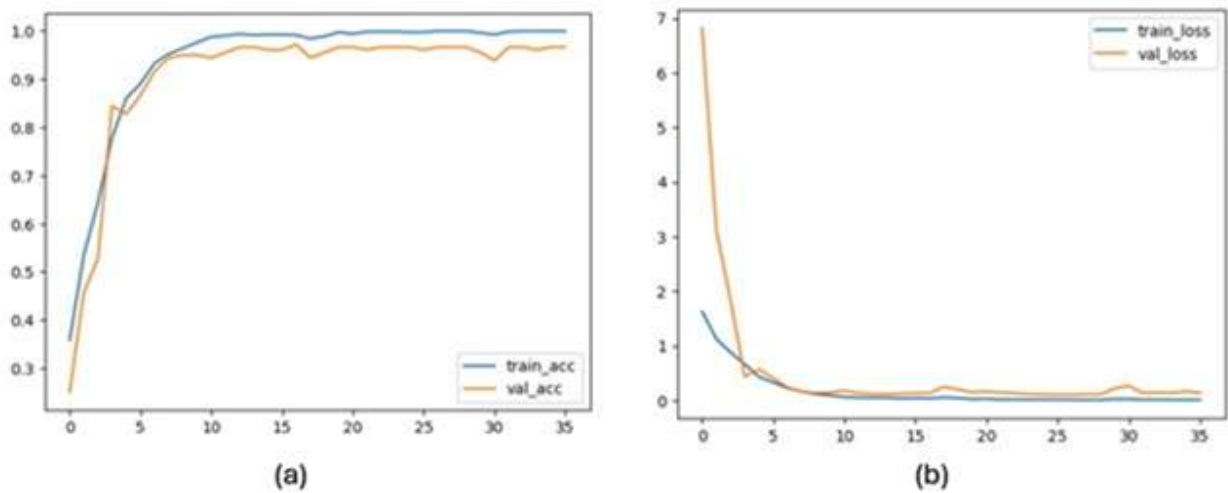


Figure 7 Learning curves of the proposed Conv²iT-BCAF model using the optimal hyperparameters selected by the BNCO algorithm: (a) training and validation accuracy and (b) training and validation loss

Table 6 Performance comparison of the proposed Conv²iT-BCAF model with state-of-the-art deep learning models on the MSID dataset

Model	Accuracy	F1	Precision	Recall/Sensitivity	Specificity	AUC	DCI
ConvNeXt	92.22	92.17	92.34	92.22	97.41	0.9889	92.17
ViT scratch	61.11	60.1	63.74	61.11	83.70	0.7676	60.10
ResNet-50	91.33	91.23	91.53	91.33	96.78	0.9870	91.23
Inception-v3	73.33	72.61	72.62	73.33	87.78	0.8156	72.61
Efficient net	90.00	89.84	89.89	90.00	96.67	0.9826	89.84
MobileNet	76.11	76.28	77.21	76.11	88.70	0.8951	76.28
VGG16	82.78	82.72	82.73	82.78	94.26	0.9663	82.72
VGG19	86.11	86.14	86.24	86.11	95.37	0.9743	86.14
Proposed	98.89	98.89	98.94	98.89	99.63	0.9972	98.89

Table 7 performance comparison of different CNN backbones within the proposed framework on the MSID dataset

Framework Configuration	Accuracy	F1 score	Sensitivity	specificity	Precision	AUC
VGG19+ViT+BCA+AFF	95.56	0.9553	95.56	98.52	0.9561	0.9921
EfficientNet-B0+ViT+BCA+AFF	96.67	0.9666	96.67	98.70	0.9684	0.9968
ResNet50+ViT+BCA+AFF	97.22	0.9722	97.20	98.50	0.9728	0.9969
Proposed Conv ² iT-BCAF	98.89	0.9889	98.94	98.89	0.9963	0.9972

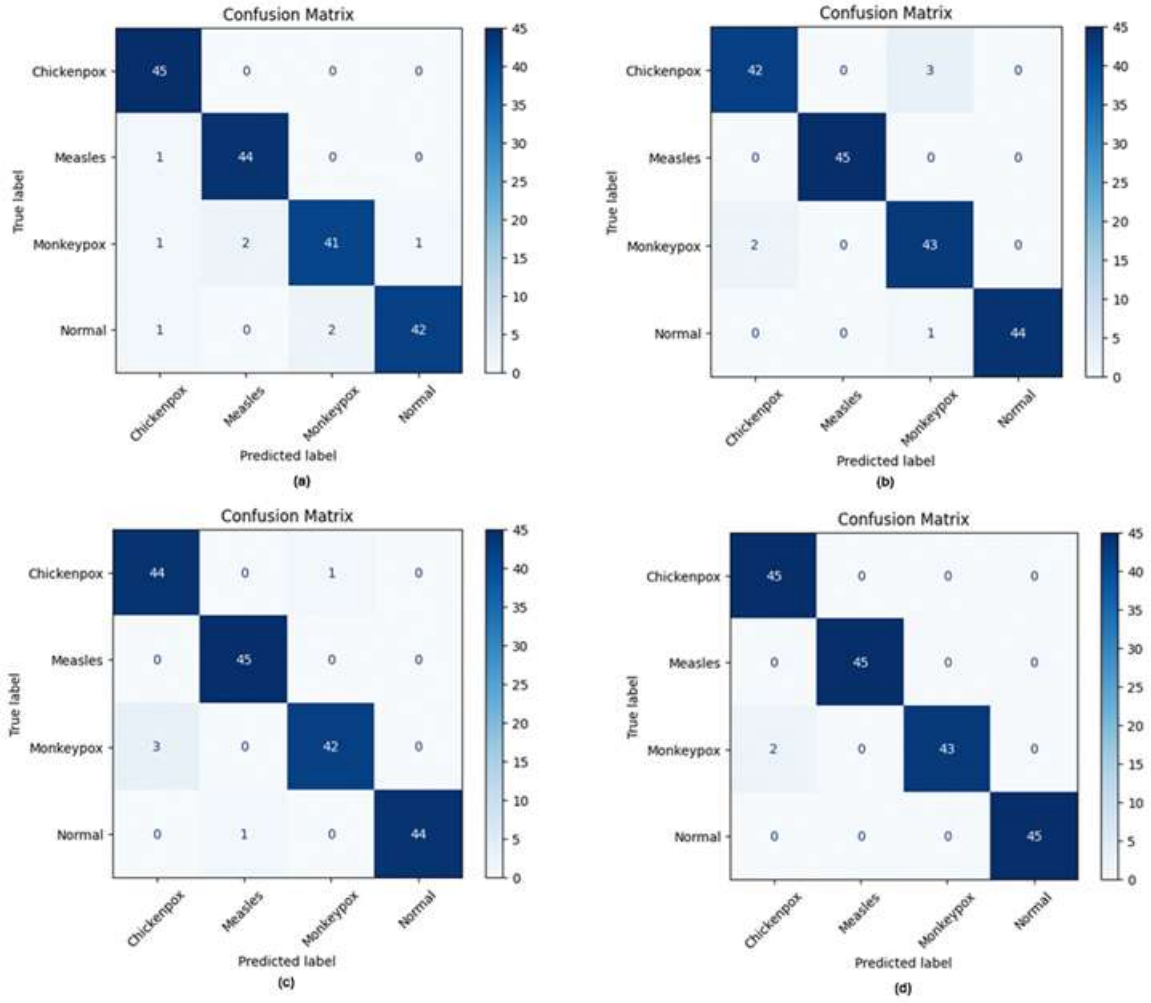


Figure 8 Confusion matrices of different CNN backbones within the proposed framework: (a) VGG19 + ViT + BCA + AF, (b) EfficientNet-B0 + ViT + BCA + AF, (c) ResNet50 + ViT + BCA + AF, and (d) the proposed Conv²iT-BCAF (ConvNeXt-T + ViT + BCA + AF).

framework. The baseline configuration consisted of parallel ConvNeXt and ViT branches, where the extracted feature representations were directly combined using feature concatenation. Subsequently, a unidirectional cross-attention mechanism was incorporated to enable one-way feature interaction between the two branches while retaining the concatenation based fusion strategy. In the third configuration, the unidirectional attention mechanism was replaced with the proposed Bidirectional Cross-Attention module to facilitate mutual information exchange between the local and global feature representations. Finally, the conventional concatenation operation was replaced by the proposed Adaptive Feature Fusion module, which adaptively learns the relative contribution of each feature stream during feature integration. Table 8 summarizes the progressive enhancement in classification performance that resulted from each architectural enhancement. The parallel ConvNeXt–ViT baseline was enhanced by the introduction of the unidirectional cross-attention mechanism, which facilitated the exchange of information between the

ConvNeXt and ViT branches. Further performance improvements were achieved by substituting it with the proposed Bidirectional Cross-Attention module, which enabled both branches to mutually refine their feature representations. The proposed Adaptive Feature Fusion module achieved the highest performance across all evaluation metrics by substituting the conventional concatenation operation. This demonstrates that adaptive feature weighting is more effective than fixed feature concatenation for integrating complementary local and global representations. These findings confirm the effectiveness of the proposed Conv²iT-BCAF framework and demonstrate that each architectural enhancement contributes to its overall classification performance.

D. Comparison with State-of-the-Art

A comparison with several recently published state-of-the-art methods reported in the literature was conducted to assess the effectiveness of the proposed Conv²iT-BCAF framework. Fig. 10 illustrates the classification accuracy achieved by the proposed

framework and the selected methods. As shown, the proposed Conv²iT-BCAF framework achieved the highest classification accuracy of 98.89%, outperforming all competing approaches. This superior performance can be attributed to the effective integration of local and global feature representations through the parallel ConvNeXt and ViT branches, together with the proposed architectural design, which enables more discriminative feature learning for pox-related skin disease classification.

5 Conclusion and Future work

This study proposed Conv²iT-BCAF framework for pox-related skin disease classification. The framework incorporates a novel image preprocessing technique called Self-Dynamic Hybrid Adaptive CLAHE and Adaptive Gamma correction (SD-HACAG) technique to enhance image quality and improve lesion visibility prior to feature extraction. Subsequently, parallel ConvNeXt

and ViT branches were employed to capture complementary local and global features, while the proposed Bidirectional Cross-Attention (BCA) and Adaptive Feature Fusion (AFF) modules facilitated effective feature interaction and integration. Furthermore, the Black-Necked Crane Optimization (BNCO) algorithm was utilized to automatically determine the optimal hyperparameter configuration. Experimental results on the MSID dataset demonstrated the effectiveness of the proposed framework, achieving a classification accuracy of 98.89% and outperforming existing methods in comparative and ablation studies. In order to further evaluate the suggested framework's robustness and generalizability, future work will focus on validating the proposed framework on larger and more diverse clinical datasets. In addition, lightweight model optimization and explainable artificial intelligence techniques will be investigated to facilitate real-time deployment and improve the interpretability of the proposed framework in clinical practice.

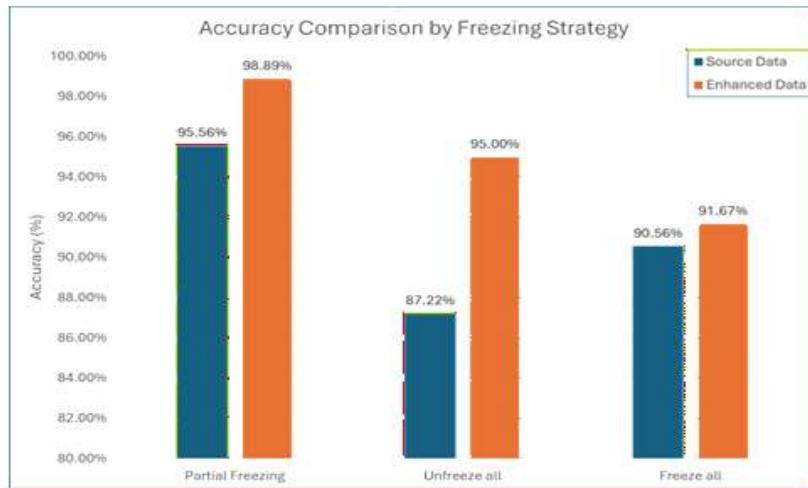


Figure 9 Comparison of ConvNeXt Fine-Tuning Strategies on the Source and Enhanced MSID Dataset

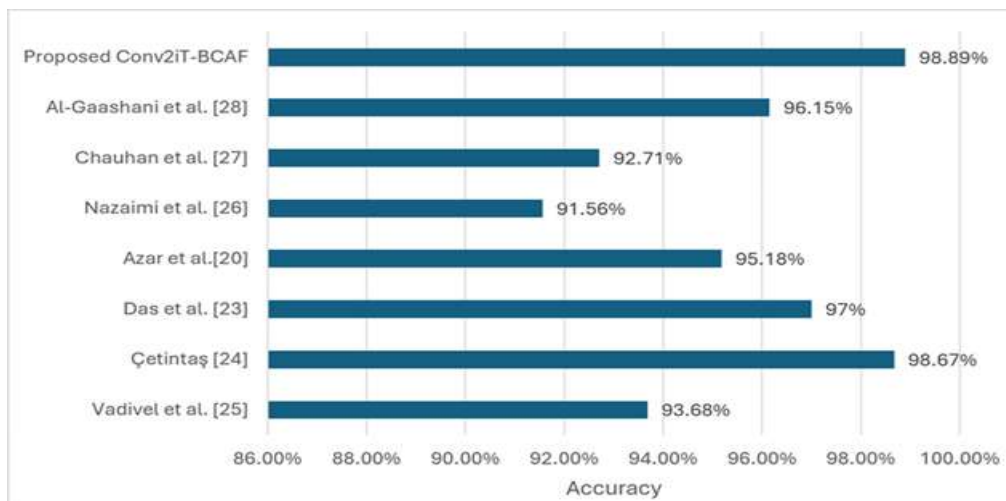


Figure 10 Comparison of the classification accuracy of the proposed Conv²iT-BCAF framework with recently published state-of-the-art methods

Table 8 Ablation study of the proposed Conv²iT-BCAF framework

Configuration	Interaction mechanism	Fusion strategy	Accuracy	F1-score	Recall	Precision	Loss crossentropy
Parallel (ConvNeXt-ViT)	None	Concatenation	95.67%	0.9568	0.9567	0.9579	0.1861
Parallel (ConvNeXt-ViT)	Unidirectional Cross-Attention	Concatenation	96.67%	0.9668	0.9667	0.9676	0.1763
Parallel (ConvNeXt-ViT)	Bidirectional Cross-Attention	Concatenation	97.22%	0.9724	0.9722	0.9726	0.1468
Proposed model	Bidirectional Cross-Attention	Adaptive Feature Fusion	98.89%	0.9889	0.9889	0.9963	0.0951

Author Statements:

- **Ethical Approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal rights on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** the dataset generated during and/or analyzed during the current study are publicly available at <https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset>

References

- [1] S. Krishna et al., "Insights into the emergence and evolution of monkeypox virus: Historical perspectives, epidemiology, genetic diversity, transmission, and preventative measures," Jun. 01, 2024, Elsevier B.V. doi: 10.1016/j.imj.2024.100105.
- [2] Y. A. Tajudeen, H. J. Oladipo, A. O. Muili, and J. G. Ikebuaso, "Monkeypox: A review of a zoonotic disease of global public health concern," 2023, Tabriz University of Medical Sciences. doi: 10.34172/hpp.2023.01.
- [3] G. Fatima et al., "MONKEYPOX OUTBREAKS: A COMPREHENSIVE REVIEW OF EPIDEMIOLOGY, CLINICAL MANAGEMENT, AND PUBLIC HEALTH RESPONSES," Jul. 08, 2025, Saint Petersburg Pasteur Institute. doi: 10.15789/2220-7619-MOA-17773.
- [4] M. Aslam Rind et al., "Global Expansion of Mpox: Addressing the Threat to Maternal Health and Healthcare Systems in Pakistan," Journal of Clinical Question, vol. 2, no. 4, p. e84, Aug. 2025, doi: 10.69854/jcq.2025.0025.
- [5] M. Soheili et al., "Monkeypox: Virology, Pathophysiology, Clinical Characteristics, Epidemiology, Vaccines, Diagnosis, and Treatments," 2022. [Online]. Available: www.cspsCanada.org
- [6] S. H. Hussein, Z. Abd Al-Azim Dakil, and B. Abed Alhassan Nasser, "CENTRAL ASIAN JOURNAL OF MEDICAL AND NATURAL SCIENCES Diagnosis and Treatment of Monkeypox Patients a Review," 2024. [Online]. Available: <https://cajmns.centralasianstudies.org/index.php/CAJMNShttps://cajmns.centralasianstudies.org/index.php/CAJMNS>
- [7] S. Sukhdeo, S. Mishra, and S. Walmsley, "Human monkeypox: a comparison of the characteristics of the new epidemic to the endemic disease," Dec. 01, 2022, BioMed Central Ltd. doi: 10.1186/s12879-022-07900-7.
- [8] A. K. Gupta, M. Talukder, T. Rosen, and V. Piguet, "Differential Diagnosis, Prevention, and Treatment of mpox (Monkeypox): A Review for Dermatologists," Am. J. Clin. Dermatol., vol. 24, no. 4, pp. 541–556, 2023, doi: 10.1007/s40257-023-00778-4.
- [9] T. P. and S. R. and M. M. E. and T. K. and S. R. and L. P. and L. S. and R. A. Jain Nityanand and Umar, "Monkeypox Diagnosis in Clinical Settings: A Comprehensive Review of Best Laboratory Practices," in Poxviruses, N. Rezaei, Ed., Cham: Springer Nature Switzerland, 2024, pp. 253–271. doi: 10.1007/978-3-031-57165-7_16.
- [10] S. Jefferson, R. Da Silva, A. Kohl, L. Pena, and K. Pardee, "iScience Clinical and laboratory diagnosis of monkeypox (mpox): Current status and future directions," iScience, vol. 26, p. 106759, 2023, doi: 10.1016/j.isci.
- [11] S. N. Ali et al., "Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study," Jul. 2022, [Online]. Available: <http://arxiv.org/abs/2207.03342>
- [12] M. Rana and M. Bhushan, "Machine learning and deep learning approach for medical image analysis: diagnosis to detection," Multimed. Tools Appl., vol.

- 82, no. 17, pp. 26731–26769, Jul. 2023, doi: 10.1007/s11042-022-14305-w.
- [13] P. Jayakanth and G. R. N. Kumari, “Dermatological image-based skin disease classification model using full resolution residual networks with attention mechanism,” *Signal Image Video Process.*, vol. 20, no. 5, p. 310, 2026, doi: 10.1007/s11760-026-05327-0.
- [14] A. S. Narasimha Raju, S. Gurunathan, R. Kumar Gatla, S. Ankalaki, and G. S. Hukkeri, “XAI-SkinCADx: A Six-Stage Explainable Deep Ensemble Framework for Skin Cancer Diagnosis and Risk-Based Clinical Recommendations,” *IEEE Access*, vol. 13, pp. 176583–176621, 2025, doi: 10.1109/ACCESS.2025.3616738.
- [15] B. and C. N. and L. P. and W. J. and Y. L. Ding Mingyu and Xiao, “DaViT: Dual Attention Vision Transformers,” in *Computer Vision – ECCV 2022*, G. and C. M. and F. G. M. and H. T. Avidan Shai and Brostow, Ed., Cham: Springer Nature Switzerland, 2022, pp. 74–92.
- [16] A. B. Siddick et al., “Hybrid Approach of Classification Monkeypox Disease: Integrating Transfer Learning with ViT and Explainable AI Urgench State pedagogical institute,” 2024. [Online]. Available: <https://www.kaggle.com/datasets/mdmokshedurrahman/>
- [17] Sanjida Akter, Mohammad Rasel Mahmud, Md Ariful Islam, Md Ismail Hossain Siddiqui, and Anamul Haque Sakib, “Efficient and interpretable monkeypox detection using vision transformers with explainable visualizations,” *International Journal of Science and Research Archive*, vol. 15, no. 1, pp. 1811–1822, Apr. 2025, doi: 10.30574/ijrsra.2025.15.1.1162.
- [18] A. M. Zbrzezny and T. Krzywicki, “Artificial Intelligence in Dermatology: A Review of Methods, Clinical Applications, and Perspectives,” Jul. 01, 2025, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/app15147856.
- [19] A. A. Adekunle, I. Fofana, P. Picher, E. M. Rodriguez-Celis, O. H. Arroyo-Fernandez, and R. Zemouri, “Optimizing deep learning predictive models: A comprehensive review of RNN and its variant architectures,” Dec. 01, 2025, Elsevier Ltd. doi: 10.1016/j.asoc.2025.114015.
- [20] A. Sorayaie Azar, A. Naemi, S. Babaei Rikan, J. Bagherzadeh Mohasefi, H. Pirnejad, and U. K. Wiil, “Monkeypox detection using deep neural networks,” *BMC Infect. Dis.*, vol. 23, no. 1, Dec. 2023, doi: 10.1186/s12879-023-08408-4.
- [21] M. Altun, H. Gürüler, O. Özkaraça, F. Khan, J. Khan, and Y. Lee, “Monkeypox Detection Using CNN with Transfer Learning,” *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23041783.
- [22] G. Yolcu Oztel, “Vision transformer and CNN-based skin lesion analysis: classification of monkeypox,” *Multimed. Tools Appl.*, vol. 83, no. 28, pp. 71909–71923, Aug. 2024, doi: 10.1007/s11042-024-19757-w.
- [23] B. Das, H. A. Dagdogan, M. O. Kaya, and R. Das, “A novel hybrid model combining Vision Transformers and Graph Convolutional Networks for monkeypox disease effective diagnosis,” *Information Fusion*, vol. 117, May 2025, doi: 10.1016/j.inffus.2024.102858.
- [24] D. Çetintaş, “Efficient monkeypox detection using hybrid lightweight CNN architectures and optimized SVM with grid search on imbalanced data,” *Signal Image Video Process.*, vol. 19, no. 4, Apr. 2025, doi: 10.1007/s11760-025-03915-0.
- [25] R. Vadiel, R. T. T. Lakshmi, B. V. S. Rao, V. G. Krishnan, N. Sivakumar, and M. Bhuvaneswari, “MLF-Net: Hybrid CNN–Transformer with Lesion-Focused Fusion for Trustworthy Multi-Class Mpx Diagnosis on MSID,” *Advances in Engineering and Intelligence Systems*, vol. 5, no. 1, pp. 59–74, Dec. 2026, doi: 10.22034/aeis.2026.569044.1406.
- [26] N. K. Binti Nazaimi, H. Mansor, T. S. Gunawan, and N. Md. Yusoff, “Machine Learning Approach for Monkeypox Detection System from Medical Images,” in *2025 IEEE 11th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2025, pp. 44–49. doi: 10.1109/ICSIMA66552.2025.11233484.
- [27] N. Chauhan and A. S. Chauhan, “A Hybrid Deep-Learning and Evolutionary Feature-Selection Framework for Skin Lesion Classification: Application to Monkeypox Detection,” *Advance Sustainable Science, Engineering and Technology*, vol. 8, no. 1, Jan. 2026, doi: 10.26877/asset.v8i1.2786.
- [28] M. S. A. M. Al-Gaashani, A. S. B. Mahel, M. M. Khayyat, and A. Muthanna, “A novel multi-scale context aggregation and feature pooling network for Mpx classification,” *Biomed. Signal Process. Control*, vol. 111, Jan. 2026, doi: 10.1016/j.bspc.2025.108254.
- [29] D. Bala et al., “MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification,” *Neural Networks*, vol. 161, pp. 757–775, 2023, doi: <https://doi.org/10.1016/j.neunet.2023.02.022>.
- [30] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” Mar. 2022, [Online]. Available: <http://arxiv.org/abs/2201.03545>
- [31] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [32] N. El Menbawy, E. M. El-Gendy, and M. M. Abdelsalam, “Black-necked crane optimizer: A novel metaheuristic algorithm for engineering applications,” *Cluster Comput.*, vol. 29, no. 5, Jun. 2026, doi: 10.1007/s10586-026-06043-7.