

ORIGINAL RESEARCH ARTICLE

Improving mpox identification under imbalanced data using geometric transformation and color augmentation with graph neural networks

Mohammad Reza Faisal^{1*}, Luu Duc Ngo², Radityo Adi Nugroho¹, Fatma Indriani¹, and Muhammad Rafi¹¹Department of Computer Science, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University, Banjarbaru, South Borneo, Indonesia²Department of Computer Science, Faculty of Engineering and Technology, Bac Lieu University, Bac Lieu, Ca Mau, Vietnam

Abstract

Mpox lesion identification from skin images remains challenging because lesion appearances often overlap with other dermatological diseases and available data are commonly imbalanced. This study proposes a graph-based multi-class classification framework that combines class-balancing augmentation, convolutional neural network (CNN)-based feature extraction, and graph neural classification. Experiments were conducted on the Mpox Skin Lesion Dataset Version 2.0 using three data configurations: (i) the original imbalanced data, (ii) a balanced version generated through geometric transformation augmentation, and (iii) a balanced version generated through combined geometric transformation and color-space augmentation. Image features were extracted using eight CNN backbones (EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetV3/InceptionV3, MobileNetV2, and LeNet), then transformed into graphs by selecting k-nearest-neighbor candidates and retaining edges according to cosine-similarity filtering, and finally classified using graph convolutional networks (GCN) and graph attention networks (GAT). Across 48 model combinations, data balancing improved macro-level performance compared with the original imbalanced setting. The best overall result was achieved by GCN with VGG16 on the geometrically and color-augmented balanced data, reaching 92.00% accuracy, 91.55% macro F1-score, 98.81% macro area under the receiver operating characteristic curve (one-versus-rest), 93.54% macro precision, and 90.32% macro recall. The best GAT result was obtained by MobileNetV2 on the geometrically augmented balanced data, with 90.67% accuracy, 89.25% macro F1-score, 98.48% macro area under the receiver operating characteristic curve (one-versus-rest), 89.28% macro precision, and 90.27% macro recall. Although GCN showed a more consistent empirical performance trend than GAT, the difference was not statistically significant. These findings indicate that integrating balanced augmentation, CNN-derived representations, and graph learning is a promising strategy for mpox lesion identification within the scope of the present experiments; however, broader external validation and explainability-oriented analysis are still required before real-world clinical deployment.

Keywords: Mpox; Image classification; Imbalanced data; Data augmentation; Graph convolutional networks; Graph attention networks

***Corresponding author:**
Mohammad Reza Faisal
(reza.faisal@ulm.ac.id)

Citation: Faisal MR, Ngo LD, Nugroho RA, Indriani F, Rafi M. Improving mpox identification under imbalanced data using geometric transformation and color augmentation with graph neural networks. *Artif Intell Health*. doi: 10.36922/AIH026270075

Received: June 29, 2026

Revised: August 10, 2026

Accepted: August 14, 2026

Published online: September 8, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

The disease was formerly referred to as monkeypox. In 2022, the World Health Organization recommended “mpox” as the preferred term for the disease. The mpox outbreak has spread to over 110 nations and has become a significant concern for global public health. According to the report, between 2022 and 2024, 12 African nations were identified as epicenters of mpox transmission, accounting for 45,652 cases and 1,492 fatalities.¹ In contrast, the incidence of mpox in Asia remains comparatively lower than that in Africa, with South Asia reporting 111 cases and Southeast Asia reporting 43 cases.²

Among the strategies to mitigate the spread of infectious disease outbreaks is the implementation of detection protocols. Currently, there exist two primary challenges associated with the execution of mpox detection. One clinically validated technique is polymerase chain reaction; however, its application is hindered by the time required to obtain results.^{3,4} The second challenge arises from the symptomatology of mpox rashes or sores, which can resemble other dermatological conditions such as chickenpox, measles, hand-foot-mouth disease, or cowpox. While healthcare professionals with expertise in these diseases can readily differentiate between them, laypersons may encounter substantial difficulties in making such distinctions.⁵ With advances in artificial intelligence (AI), the diagnostic capabilities of medical practitioners can be supported by AI-based approaches, thereby enabling a broader demographic to participate in the detection process.⁶

Research on the use of AI that employs image data is commonly referred to as computer vision research. Computer vision has been widely implemented in healthcare and medical applications for detecting diseases such as stomach cancer, bladder cancer, lung disease, breast cancer, skin disease, prostate cancer, pneumonia, brain tumor, and osteoarthritis.^{7,8} Research in computer vision has mainly focused on developing preprocessing techniques, feature extraction, and classification algorithms to improve the performance of AI models.^{8,9} In addition to analyzing medical images, machine learning algorithms have also been developed as classification frameworks for monitoring human health conditions, demonstrating the potential of AI to support disease identification and decision-making processes in healthcare.¹⁰ The success of these computer vision studies on medical image data provides the basis for the application of computer vision to mpox identification in the present study.

The development of an image-based mpox rash identification model generally uses the Mpox Skin Lesion

Dataset Version 2.0 (MSLD v2.0).^{11–16} This dataset consists of 755 images from 541 patients across six classes. The challenge of using this dataset lies in its limited size and the unequal distribution of samples across classes, ranging from 55 images for the measles class to 284 images for the mpox category. A model trained directly on the original dataset may therefore be biased toward the majority class and show lower sensitivity to minority classes.

In prior studies, solutions to the limited number of images in the MSLD v2.0 dataset were proposed by increasing the number of images through image augmentation.^{12–16} The image augmentation techniques used in earlier studies can be grouped into two types: geometric augmentation and color space augmentation. Some studies used only geometric augmentation^{12,16}, whereas other studies combined both types of augmentation.^{13–15} Nevertheless, all of these studies increased the number of images without considering the balance of the class distribution. As a result, the resulting models may be biased toward the majority class.

Image-based mpox lesion identification research initially used machine learning algorithms that required convolutional neural network (CNN)-based feature extraction so that images could be represented as numerical data.¹² Subsequent developments have been dominated by deep learning, particularly CNNs^{13,14,16} and transformers^{11,15}, which can perform feature extraction automatically. Alongside these two deep learning approaches, graph neural network (GNN) architectures were also developed, especially graph convolutional networks (GCNs) and graph attention networks (GATs), which have been used in various image classification tasks by modeling images as nodes and similarity relationships as edges.^{17,18} However, based on a review of the literature, the use of GCNs and GATs for mpox lesion classification remains limited.

Given the limited number of samples and class imbalance in the MSLD v2.0 dataset, along with the still-limited adoption of GNNs for mpox lesion classification, this study integrates augmentation-based class balancing, CNN-based feature extraction, feature-similarity-based graph construction, and graph-based classification. Geometric and hue–saturation–value (HSV) color transformations are applied to generate balanced training data. Next, eight CNN architectures—EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetV3/InceptionV3, MobileNetV2, and LeNet—are evaluated as feature extractors. The resulting feature vectors are then used to construct graphs before being classified using GCN and GAT.

This study aims to:

- (i) Investigate the effect of geometric augmentation and HSV-based augmentation with class balancing on the classification of mpox skin lesions.
- (ii) Compare the effectiveness of eight CNN architectures as feature extractors for graph-based classification.
- (iii) Compare the performance of GCN and GAT in classifying skin lesions across six classes.
- (iv) Assess the performance of the proposed framework relative to approaches used in prior studies based on MSLD v2.0.

The main contribution of this study is the integration of augmentation-based class balancing, multiple CNN feature extractors, and graph-based classification for mpox lesion identification. Unlike prior studies that use augmentation only to enlarge the dataset, the proposed strategy uses augmentation to achieve an equal number of training samples across classes. This study also examines the performance of models built using feature extraction from several CNN architectures and two GNN variants, namely GCN and GAT, to determine which combination yields the best performance.

The remainder of this article is organized as follows. Section 2 presents the related work, including a review of the most recent studies on mpox skin lesion classification covering image data augmentation techniques, feature extraction techniques, and classification algorithms for image-based cases. Section 3 describes the data and methods, providing a detailed explanation of the dataset, the research stages, and the proposed method used in this study. Section 4 reports the results, including the performance of 48 models formed from combinations of three dataset variants, eight feature extraction techniques, and two GNN-based classification algorithms. Section 5 discusses the findings, compares them with previously published approaches, and presents implications and limitations. Finally, Section 6 provides the conclusions and directions for future research.

2. Related work

2.1. Deep learning approaches for mpox skin lesion classification

Recent studies have investigated automated mpox identification from skin lesion images using machine learning, deep learning, and transformer-based architectures. Many of these studies have employed the MSLD v2.0, which contains 755 images distributed across six diagnostic classes. Despite its relevance to multi-class skin lesion classification, the dataset is characterized by a relatively limited sample size and substantial class imbalance.^{6,12,13,16}

Existing mpox classification studies can broadly be grouped into three research directions: feature representation, classifier development, and data augmentation. The first direction focuses on extracting discriminative visual representations from skin lesion images. CNNs are commonly employed as feature extractors due to their ability to learn hierarchical visual characteristics, including lesion shape, texture, color distribution, and spatial patterns. The extracted representations are subsequently classified using conventional machine learning methods, such as random forest, support vector machine, LightGBM, Extra Trees, XGBoost, and an adaptive Bayesian-optimized boosting learning machine. One such study reported an F1-score of approximately 82% using this combined feature-extraction and machine-learning framework.¹²

The second research direction concerns the development and comparison of classification architectures. Previous studies have evaluated conventional machine learning algorithms, including logistic regression and K-nearest neighbors (KNNs), alongside deep learning models such as MobileNetV2, VGG16, ResNet50, GoogleNetV3/InceptionV3, and DenseNet201.¹² Transformer-based architectures have also been introduced due to their ability to capture long-range dependencies among image regions. An ensemble combining a vision transformer and ConvMixer achieved an F1-score of 92.97% within its experimental setting, exceeding the results obtained by the individual CNN-based models evaluated in the same study.¹³

Other studies have examined transfer-learning and self-supervised representation-learning approaches, including a vision transformer, a masked autoencoder, DINO, and Swin Transformer.¹¹ Among the architectures evaluated in that study, Swin Transformer produced the highest reported performance, with an F1-score of 93.58%. These findings indicate that hierarchical transformer architectures can provide competitive representations for mpox lesion classification.

More recently, another study evaluated seven pretrained CNN and transformer backbones—EfficientNetV2-S, ConvNeXt-Tiny, ViT-B/16, ResNet-50, DenseNet-121, EfficientNet-B0, and MobileNetV3-Large—for binary mpox classification. The predicted probabilities of these models were combined using a weighted soft-voting ensemble, with the ensemble weights optimized through random search. Within the corresponding experimental setting, the ensemble achieved an average F1-score of 83.34% and an area under the curve (AUC) of 93.88%, whereas ConvNeXt-Tiny was the best-performing individual model with an F1-score of 81.59%.¹⁹

Nevertheless, direct comparisons across studies should be interpreted cautiously because the reported results may be based on different data partitions, augmentation strategies, preprocessing procedures, and evaluation protocols.

2.2. Data augmentation and class-imbalance handling

The third major research direction concerns the use of data augmentation to address the limited number of mpox skin lesion images. Data augmentation is widely adopted in medical image classification because model performance is strongly influenced by the quantity and diversity of training samples. This issue is particularly relevant to MSLD v2.0, which contains only 755 images and exhibits considerable variation in the number of samples available for each class.

Geometric augmentation techniques, including rotation, flipping, translation, cropping, and scaling, have been used to expand the dataset. Previous studies increased the number of images to approximately 2,625 and 7,532 samples through geometric transformations.^{12,13} Other studies combined geometric transformations with zooming and noise injection and subsequently evaluated classifiers such as ConvNeXt, MobileNetV2, DenseNet121, Xception, and Swin Transformer.¹⁶ Within that experimental configuration, the Swin Transformer model achieved the highest F1-score of approximately 91%.¹⁶

More comprehensive augmentation strategies have additionally incorporated color-space transformations. For example, MSMP-Net employed geometric transformations, noise injection, and variations in hue and saturation¹⁵. The proposed architecture used ConvNeXt as its backbone and incorporated multi-scale feature fusion to integrate local lesion information with higher-level semantic representations. This model achieved an F1-score of 90.24%.¹⁵

Another study applied geometric transformations, image-noise augmentation, and color manipulation in the HSV color space.¹⁴ The HSV augmentation was intended to increase model robustness to variations in skin appearance and skin tone across diverse populations. The resulting images were evaluated using transfer-learning architectures, including VGG16, ResNet50, DenseNet121, MobileNetV2, EfficientNetB3, GoogleNetV3/InceptionV3, and Xception, with models pretrained using ImageNet- and HAM10000-based representations. The best-performing configuration combined HAM10000-based pretraining with DenseNet121 and achieved an F1-score of 88%.¹⁴

Although these augmentation strategies increased the number and diversity of images, most previous studies primarily focused on dataset expansion rather than explicit class balancing. Simply generating additional

images does not necessarily ensure an equal distribution among diagnostic classes. Consequently, models trained on augmented but still imbalanced datasets may remain biased toward majority classes and may provide lower sensitivity for minority skin conditions. Therefore, augmentation should not only increase the overall number of samples but should also be designed to achieve a more balanced class distribution.

2.3. Graph neural networks for image classification

Most previous mpox studies have relied on conventional machine learning, CNN-based deep learning, or transformer-based architectures. An alternative approach is offered by GNNs, which represent samples as nodes and their relationships as edges. GCNs and GATs are two widely used GNN variants for classification tasks.^{17,18} GCNs aggregate information from neighboring nodes through graph convolution, whereas GATs assign different attention weights to neighboring nodes according to their relevance.

Graph neural network-based approaches have been successfully applied to various computer-vision and medical-image classification problems. However, unlike CNNs, GCNs and GATs typically do not directly process raw image pixels in their conventional formulation. Images are first converted into numerical feature representations, after which relationships among samples can be constructed based on feature similarity.²⁰ CNN architectures can therefore serve as feature extractors, while GCNs or GATs can exploit relationships among the resulting image representations.

Despite the potential of this approach, the application of GCNs and GATs to multi-class mpox skin lesion identification remains limited. Furthermore, previous studies have not systematically investigated how different CNN feature extractors affect the performance of graph-based mpox classifiers. Architectures such as EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetV3/InceptionV3, MobileNetV2, and LeNet produce feature representations with different dimensions, depths, and levels of abstraction. Their integration with graph-based classifiers may therefore produce substantially different classification outcomes.

Accordingly, the present study addresses two limitations identified in the existing literature. First, it investigates class-balancing strategies based on geometric transformations and a combination of geometric and HSV color transformations rather than merely increasing the overall number of images. Second, it systematically evaluates eight CNN architectures—EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetV3/

InceptionV3, MobileNetV2, and LeNet—as feature extractors for GCN- and GAT-based classification. The experimental framework is designed to assess the effects of class balancing, feature-extraction architecture, and graph-based classification on multi-class mpox skin lesion identification.

3. Data and methods

The procedural framework employed in this investigation is shown in Figure 1.

3.1. Dataset

The dataset used in this study is the MSLD v2.0, a publicly available dermatological image dataset developed for multi-class classification of mpox and visually similar skin diseases. MSLD v2.0 consists of 755 original skin lesion images collected from 541 patients and categorized into six classes: Mpox; Chickenpox; Measles; Cowpox; Hand, Foot, and Mouth Disease (HFMD); and Healthy.^{11–13,16}

Table 1 presents the details of the MSLD v2.0 dataset,

and Figure 2 shows an example image of each class from the MSLD v2.0 dataset.

The dataset exhibits class imbalance, as the number of images varies considerably across classes. The Mpox class contains the largest number of samples, whereas the Measles class has the smallest number of images. This imbalance represents an important challenge in model development because it may cause the classifier to be biased toward the majority classes and reduce its sensitivity in recognizing minority classes.^{21,22}

3.2. Data augmentation

Image data augmentation was applied to increase the variability of the training data and improve the generalization ability of the proposed model. This process generates additional training samples by applying controlled transformations to the original images without changing their class labels.^{23,24} In this study, image augmentation was grouped into two main categories: geometric transformations and color-space transformations.

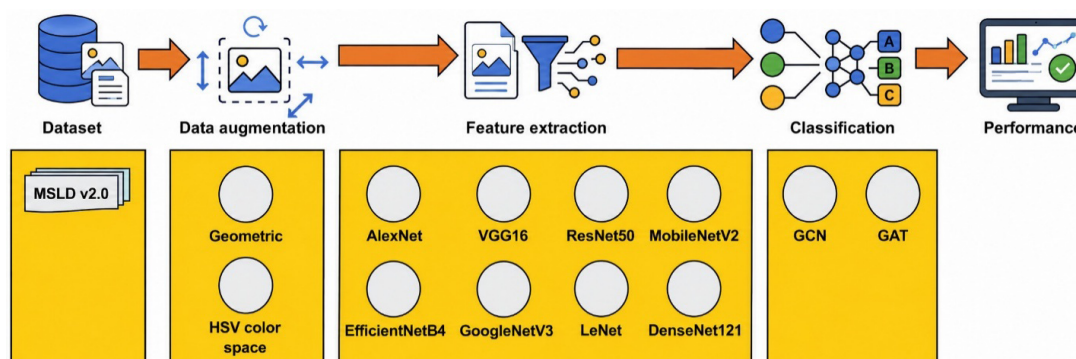


Figure 1. Research flow

Abbreviations: GAT: Graph attention network; GCN: Graph convolutional network; HSV: Hue–saturation–value; MSLD v2.0: Mpox Skin Lesion Dataset Version 2.0.

Table 1. Mpox Skin Lesion Dataset Version 2.0

Class	Number of images	Description
Mpox	284	Skin lesion images associated with mpox infection
Chickenpox	75	Skin lesion images associated with varicella/chickenpox
Measles	55	Rash images associated with measles
Cowpox	66	Skin lesion images associated with cowpox infection
HFMD	161	Skin lesion images associated with hand-foot-mouth disease
Healthy	114	Normal/healthy skin images
Total	755	Original images from 541 patients

Abbreviation: HFMD: Hand, foot, and mouth disease.

Geometric transformations were used to introduce spatial variation in the lesion images. The applied operations included horizontal and vertical flipping, cropping and zooming, and shearing.^{12,13,23} These transformations help the model become more robust to variations in lesion position, orientation, size, and viewpoint, thereby improving its ability to recognize lesion patterns under

different image acquisition conditions. Figure 3 shows examples of the results of the geometric transformations applied in this study. Algorithms 1 and 2 describe the complete geometric transformation procedure employed in this study. The input to this procedure is the imbalanced MSLD v2.0 training data, and the output is a balanced training set containing 250 images per class.

Algorithm 1. Geometric augmentation for dataset balancing

Input: original image dataset D organized by class.

Output: class-balanced dataset $D_{balanced}$ containing 250 images per class, resized to 224×224 pixels.

```

1  procedure BALANCE-DATASET( $D$ )
2    Set target images per class  $T \leftarrow 250$  and image size
       $S \leftarrow 224 \times 224$ 
3    List all valid images and group them according to their class labels
4    Count the number of images  $n_c$  in each class
5    for each class  $c$  do
6       $I_c \leftarrow$  images belonging to class  $c$ 
7      if  $n_c \geq T$  then
8        Randomly select  $T$  images from  $I_c$ 
9        Resize the selected images to  $S$  and add them to  $D_{balanced}$ 
10     else
11       Resize all original images in  $I_c$  to  $S$  and add them to  $D_{balanced}$ 
12        $N_{needed} \leftarrow T - n_c$ 
13       while generated images  $< N_{needed}$  do
14         Select an original image  $x$  and one geometric transformation  $g$ 
15          $x_{aug} \leftarrow$  APPLY-GEOMETRIC-AUGMENTATION( $x, g$ )
16         Add  $x_{aug}$  to class  $c$  in  $D_{balanced}$ 
17       end while
18     end if
19   end for
20   Verify that every class contains exactly  $T$  images
21   return  $D_{balanced}$ 
22 end procedure

```

Algorithm 2. Geometric transformation procedure

```

1  procedure APPLY-GEOMETRIC-AUGMENTATION( $x, g$ )
2    if  $g = \text{rotation}$  then
      Select
3       $\theta \in \{45, 60, 90, 120, 135, 150, 180, 210, 225, 240, 270, 300, 315, 330\}^\circ$ 
      and rotate the image around its center
4    else if  $g = \text{horizontal flip}$  then
5      Reflect the image across the vertical axis
6    else if  $g = \text{vertical flip}$  then
7      Reflect the image across the horizontal axis
8    else if  $g = \text{crop-zoom}$  then
9      Select crop ratio  $r \in [0.80, 0.95]$ , crop randomly, and resize to
        the original dimensions
10   else if  $g = \text{shear}$  then
11     Select horizontal shear factor  $h \in [-0.30, 0.30]$  and apply the
        shear transformation
12   else if  $g = \text{zoom}$  then
13     Select zoom factor  $z \in [0.90, 1.10]$  and scale the image around
        its centre
14   end if
15   Use reflected border padding for rotation, shear, and zoom
        transformations
16   return the transformed image
17 end procedure

```

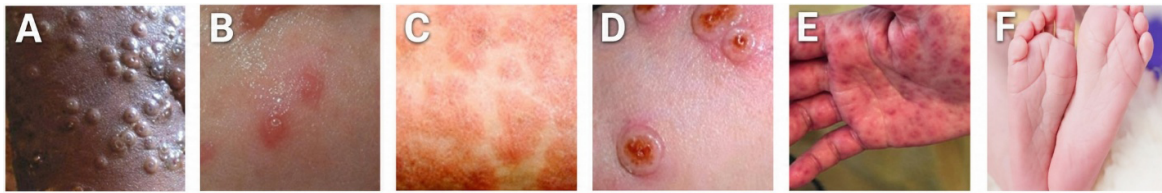


Figure 2. Example of skin lesion images from each class in the Mpox Skin Lesion Dataset Version 2.0: (A) Mpox, (B) Chickenpox, (C) Measles, (D) Cowpox, (E) Hand, Foot, and Mouth Disease, and (F) Healthy

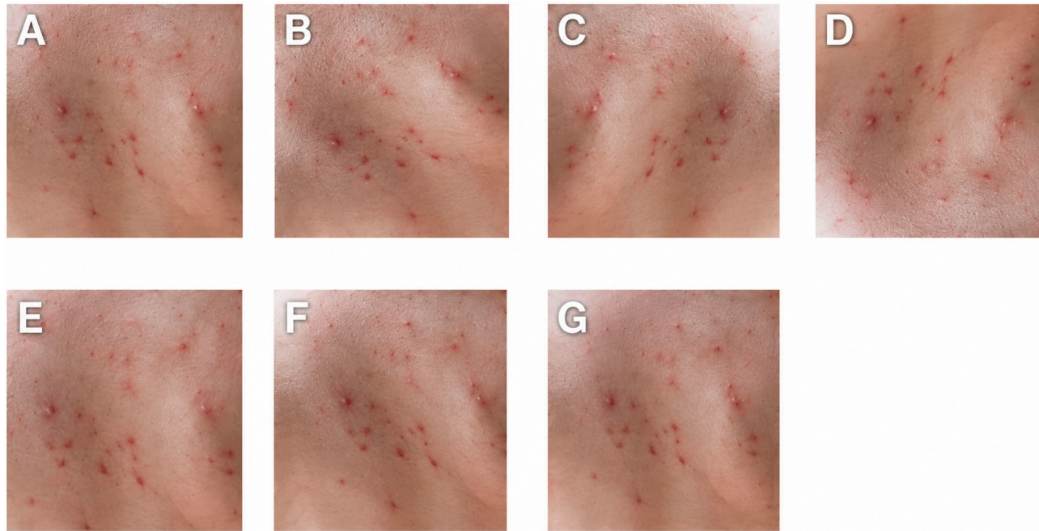


Figure 3. Geometric transformation results: (A) Original, (B) rotation, (C) flip horizontal, (D) flip vertical, (E) crop zoom, (F) shear, and (G) zoom

where n_c denotes the number of images in class c and N_{needed} is the number of augmented images required to reach the target.

Color-space transformations were used to simulate variations in skin tone, illumination, and image acquisition conditions. This augmentation was performed by modifying color-related components, particularly in the HSV color space, where hue represents the dominant color component, saturation represents color intensity, and value represents brightness.^{14,15,25,26} Additionally, image contrast was modified. This technique helps the model learn diverse

color characteristics of skin lesions and enhances its robustness in recognizing lesion patterns across different skin tones and lighting conditions.¹⁵ This technique can also help increase data diversity while accounting for racial diversity.¹⁴ Figure 4 shows examples of the color-space transformation results obtained in this study. The complete color-space transformation procedure is presented in Algorithm 3. The input to this algorithm is the balanced training data generated in the previous stage. Each input image produces 24 augmented images in addition to the original image. The clip operation limits channel values to $[0, 255]$, and hue is wrapped to $[0, 179]$ using modulo 180.

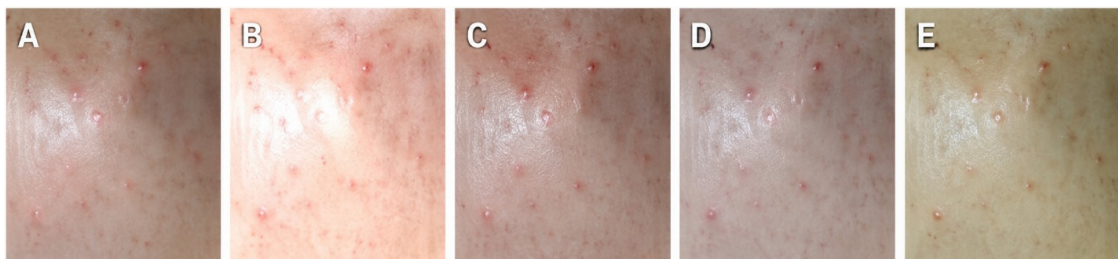


Figure 4. Color-space transformation results: (A) Original, (B) brightness (1.30), (C) contrast (0.84), (D) saturation (0.71), and (E) hue shift (16.0°)

Algorithm 3. HSV color augmentation

Input: an RGB image x .

Output: a set of 24 HSV-augmented RGB images A .

```

1  procedure HSV-COLOR-AUGMENTATION( $x$ )
2    Define hue shifts  $H \leftarrow \{30, 60, 90, 120, 150, 180\}$ 
3    Define saturation factors  $S \leftarrow \{0.5, 1.0\}$ 
4    Define value factors  $V \leftarrow \{0.5, 1.0\}$ 
5     $A \leftarrow \emptyset$ 
6    Convert  $x$  from RGB to HSV and separate the original channels
        $H_0, S_0$ , and  $V_0$ 
7    for each  $h \in H$  do
8      for each  $s \in S$  do
9        for each  $v \in V$  do
10          $H' \leftarrow (H_0 + h) \bmod 180$ 
11          $S' \leftarrow \text{clip}(S_0 \times s, 0, 255)$ 
12          $V' \leftarrow \text{clip}(V_0 \times v, 0, 255)$ 
13         Combine  $H', S'$ , and  $V'$  into an HSV image
14         Convert the augmented HSV image to RGB and add it to  $A$ 
15       end for
16     end for
17   end for
18   Verify that  $A$  contains 24 augmented images
19   return  $A$ 
20 end procedure

```

where H_0 , S_0 , and V_0 denote the original HSV channels, and h , s , and v are the selected hue, saturation, and value parameters.

3.3. Feature extraction

Feature extraction is a process of transforming raw data into meaningful representations that can be used by machine learning or deep learning models for analysis and classification.^{27–29} In image-based tasks, feature extraction aims to capture discriminative visual characteristics, such as color, texture, shape, edges, patterns, and structural

information. In general, feature extraction converts an image into structured data with numerical values because machine learning and deep learning algorithms are fundamentally composed of mathematical operations that require numerical inputs. This representation enables the model to process visual information computationally and learn relevant patterns for classification.³⁰

Currently, one of the most widely used approaches for image feature extraction is the CNN. Compared with traditional handcrafted feature extraction methods, CNNs can automatically learn discriminative features from image data. This capability reduces the need for manual feature engineering and enables the model to extract hierarchical visual representations directly from raw images.⁸ Figure 5 illustrates the general architecture of a CNN.

The CNN architecture employed in this investigation was initiated by incorporating images of skin lesions as the input data. These input images were subsequently processed by convolutional layers to extract localized visual patterns, including edges, colors, textures, lesion morphologies, and skin surface structures. The output from the convolutional layers is represented by feature maps that encapsulate the fundamental characteristics of the image. Moreover, the feature maps were subjected to the rectified linear unit activation function, which introduces nonlinearity, thereby enabling the CNN to capture more intricate visual patterns. Subsequently, pooling layers were implemented to reduce the spatial dimensions of the feature maps while preserving critical information, thereby enhancing computational efficiency and increasing the model's resilience to minor variations in the position or size of the objects depicted in the image.¹⁶

The feature extraction procedure was then repeated across multiple convolutional blocks, comprising convolutional layers, activation functions, and pooling layers, in an iterative manner. At this stage, the CNN extracts more abstract and semantic features, such as rash patterns, lesion geometries, and visual attributes that differentiate one category of skin pathology from another. The feature maps produced by the convolutional blocks were subsequently aggregated via flattening or global average pooling, thereby converting the two-dimensional feature representation into a one-dimensional feature vector. This feature vector is a numerical representation of an image that captures significant visual information resulting from the CNN's feature extraction process.³¹

Subsequently, the feature vector was utilized as input for the fully connected layer to model the relationships between the extracted features and the target classification. Fully connected layers serve as the classification component,

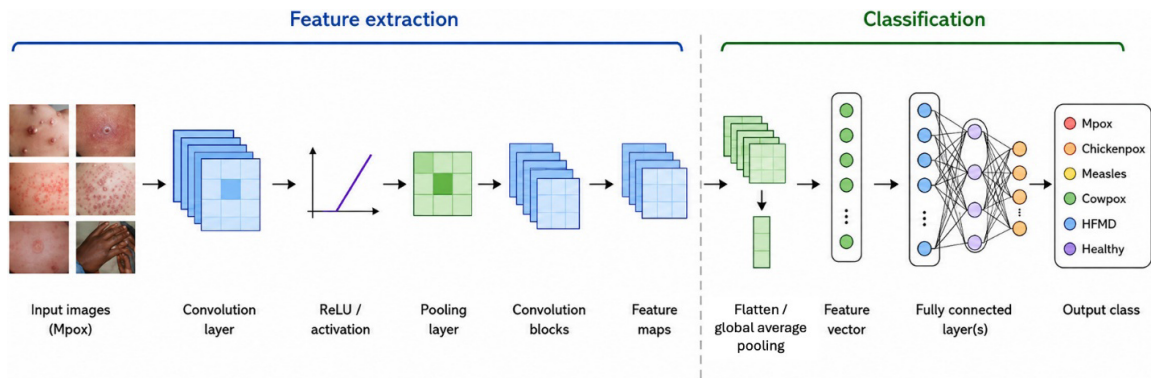


Figure 5. Standard architecture of a convolutional neural network
Abbreviations: HFMD: Hand, Foot, and Mouth Disease; ReLU: Rectified linear unit.

transforming feature representations into probabilistic scores for each class. In the final stage, the model selects the class with the highest score as the prediction. When a CNN is employed as a feature extractor, the output is derived from the flattening or global average pooling layer. This output is then engaged as input for the classification algorithm.³¹

The investigation employed several CNN architectures for image feature extraction, specifically EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetV3/ InceptionV3, MobileNetV2, and LeNet. These architectures represent extensions of standard CNNs, comprising convolutional layers, activation functions, pooling layers, and fully connected layers. The distinctions among these architectures lie in the arrangement of the layers, the depth of the network, the connectivity mechanisms, and the strategies employed for feature extraction.

In terms of their layer configuration and feature extraction methods, LeNet, AlexNet, and VGG16 are categorized as sequential CNNs, given their relatively linear flow of layers from convolution to activation, pooling, and ultimately to fully connected layers. In contrast, ResNet50 and DenseNet121 are CNNs characterized by inter-layer connectivity: ResNet50 employs residual connections, while DenseNet121 uses dense connections to enhance information flow and facilitate feature reuse.^{31,32}

GoogleNetV3, also referred to as InceptionV3, is a CNN architecture that employs multi-scale convolution by using an array of filter sizes concurrently to capture features across a spectrum of scales. In contrast, MobileNetV2 and EfficientNetB4 are characterized by lightweight and computationally efficient CNN architectures. Specifically, MobileNetV2 employs depthwise separable convolution along with an inverted residual block, while EfficientNetB4 leverages MBConv and compound scaling

methodologies to yield robust feature representations that are computationally optimal.^{31,33}

To systematically extract feature representations from skin lesion images using all eight CNN architectures, a comprehensive multi-architecture feature extraction procedure was implemented. This procedure processes the entire dataset (comprising training, validation, and test splits) through each architecture independently, ensuring that discriminative visual information is captured consistently across all models. The extracted features are subsequently prepared as structured datasets for downstream classification tasks. The algorithmic workflow of this multi-architecture feature extraction process is formalized in [Algorithm 4](#), which outlines the sequence of operations from data loading to feature storage.

Algorithm 4. CNN-based image feature extraction

Input: Image dataset $D = \{(x_i, y_i)\}_{i=1}^N$, target image size $S = 224 \times 224$, and the set of CNN architectures M .
Output: Eight structured feature datasets $\mathcal{F} = \{(F^{(m)}, Y^{(m)}) \mid m \in M\}$.

```

1  procedure EXTRACT-CNN-FEATURES(  $D$  )
2
3       $S \leftarrow (224, 224)$ 
4
5       $M \leftarrow \{\text{EfficientNetB4, AlexNet, VGG16, ResNet50,}$ 
6           $\text{DenseNet121, GoogleNetV3, MobileNetV2, LeNet}\}$ 
7
8       $\mathcal{F} \leftarrow \emptyset$ 
9
10     for each  $m \in M$  do
11         Load CNN architecture  $m$ 

```

(Cont'd...)

Algorithm 4. (Continued)

Algorithm 4. CNN-based image feature extraction

```

7   Remove or bypass the final classification layer of  $m$ 
8   Select the designated feature-extraction layer  $L_m$ 
9   Initialize feature matrix  $F^{(m)} \leftarrow \emptyset$ 
10  Initialize label vector  $Y^{(m)} \leftarrow \emptyset$ 
11  for each image-label pair  $(x_i, y_i) \in D$  do
12     $x_i' \leftarrow \text{RESIZE}(x_i, S)$ 
13     $x_i'' \leftarrow \text{PREPROCESS}(x_i', m)$ 
14     $Z_i^{(0)} \leftarrow x_i''$ 
15    for each convolutional block  $l$  in architecture  $m$  do
16       $Z_i^{(l)} \leftarrow \text{CONVOLUTION}(Z_i^{(l-1)}, W_l, b_l)$ 
17       $Z_i^{(l)} \leftarrow \text{ReLU}(Z_i^{(l)})$ 
18       $Z_i^{(l)} \leftarrow \text{POOLING}(Z_i^{(l)})$ 
19    end for
20     $f_i^{(m)} \leftarrow \text{EXTRACT-OUTPUT}(Z_i^{(L_m)})$ 
21    if  $f_i^{(m)}$  is a multidimensional feature map then
22       $f_i^{(m)} \leftarrow \text{FLATTEN-OR-GLOBAL-AVERAGE-POOLING}(f_i^{(m)})$ 
23    end if
24    Append  $f_i^{(m)}$  to  $F^{(m)}$ 
25    Append  $y_i$  to  $Y^{(m)}$ 
26  end for
27  Store  $(F^{(m)}, Y^{(m)})$  in  $\mathcal{F}$  under architecture  $m$ 
28 end for
29 return  $\mathcal{F}$ 
30 end procedure

```

Convolutional feature transformation:

$$Z_i^{(l)} = \sigma(W_l * Z_i^{(l-1)} + b_l) \quad (1)$$

Feature vector generated by architecture m :

$$f_i^{(m)} = \varphi_m(x_i) \in \mathbb{R}^{p_m} \quad (2)$$

Feature matrix and label vector:

$$F^{(m)} = [f_1^{(m)}, f_2^{(m)} \dots, f_N^{(m)}]^T \in \mathbb{R}^{N \times p_m} \quad (3)$$

$$Y = [y_1, y_2 \dots, y_N]^T$$

where:

- (i) N denotes the number of input images.
- (ii) m denotes a CNN architecture.
- (iii) p_m denotes its feature dimension.
- (iv) $\varphi_m(\cdot)$ denotes the feature-extraction mapping.
- (v) $F^{(m)}$ and $Y^{(m)}$ denote the feature matrix and label vector, respectively.

The final CNN classification layer is excluded.

3.4. Classification

This research employed two distinct variants of GNNs, namely GCNs and GATs. Both algorithms were used after the CNN-based image feature extraction phase. The features produced by the CNN, represented as feature vectors, cannot be directly processed by GCN or GAT, since both algorithms require graph-structured data. Consequently, the feature vector corresponding to each image was initially represented as a node, followed by the establishment of relationships between nodes, encoded as edges, based on the degree of feature similarity. In this manner, image data, initially represented as feature vectors, are transformed into a graph structure comprising nodes, edges, and feature representations associated with each node.^{18,34}

Before graph construction, the CNN-derived feature vectors were standardized to reduce differences in their numerical scales. Each image was represented as a node, while candidate relationships between nodes were identified using the KNN method. Cosine similarity was subsequently calculated between neighboring feature vectors, and an edge was retained when its similarity satisfied a percentile-based threshold. If none of a node's candidate connections satisfied the threshold, the node was connected to its most similar candidate to prevent isolation. The retained node pairs were represented as bidirectional

edges, thereby enabling information to propagate between related images during GCN or GAT processing.^{35–37}

Two graph-learning settings are generally distinguished in the literature: transductive and inductive learning. In the transductive setting, the training, validation, and test samples are included as nodes within a single graph. Although the features and graph relationships of all samples are available during model training, only the labels of the training nodes contribute to the loss and parameter updates, with separate node masks distinguishing the three subsets during optimization and evaluation. In the inductive setting, by contrast, feature standardization, similarity estimation, and initial graph construction are performed exclusively using the training samples. Validation and test samples are treated as previously unseen nodes and are introduced only during their respective evaluation phases, each connected to similar training nodes through directed edges oriented from the training nodes towards the unseen node. Unseen nodes therefore receive information from the training graph without influencing the representations of the original training nodes, which more closely reflects practical deployment scenarios in which newly acquired images are not available during model training.^{35,38}

The present study adopted the transductive setting for all reported experiments. This choice reflects the objective of the study, which is to compare the effect of successive data augmentation strategies across three dataset variants under a consistent classification protocol. The transductive setting isolates the influence of the augmentation strategy by allowing the full relational structure of each dataset to be exploited. A systematic evaluation of the inductive setting, together with the additional considerations it raises regarding graph construction from training samples alone, is reserved for future work.

Graph neural networks represent a deep learning methodology designed to analyze data with relational structure. In contrast to CNNs, which focus on local patterns within images, and transformers, which examine global relationships among patches via attention mechanisms, GNNs explicitly investigate relationships among data through graph structures. In image classification, GNNs can effectively model inter-image relationships based on the similarity of visual features extracted by CNNs. This methodology enables the model to analyze not only the individual characteristics of each image but also the interrelationships among images with analogous feature patterns.³⁹ This aspect is particularly relevant to the classification of skin lesions, as numerous disease categories exhibit visual similarities; hence, relational information among samples can augment the model's capacity to differentiate between classes that are inherently challenging to separate.

A GCN aggregates information from adjacent nodes according to the established graph structure. At each layer, the GCN refines a node's representation by combining its features with those of its neighbors. This iterative process facilitates information exchange among nodes, ensuring that each node's final representation is influenced not only by its own image features but also by the features of related images within the graph. Through this mechanism, GCNs can model relational patterns among images and generate more informative node representations to enhance the classification process.¹⁸

Graph attention networks are a class of GNNs that incorporate an attention mechanism during the aggregation of information among nodes. Unlike GCNs, which typically use fixed propagation rules based on graph structure, GAT assigns varying attention weights to each neighboring node. Thus, neighboring nodes deemed more pertinent receive greater weight in updating the node representation. This adaptive mechanism allows GAT to prioritize information from more relevant neighboring nodes.⁴⁰ In the context of classifying skin lesion images, GAT can help the model focus on images with the most relevant visual features, thereby potentially improving classification performance for classes with similar lesion patterns.

To systematically implement these relational classification mechanisms, this study applied a unified workflow encompassing feature standardization, similarity-based graph construction, configuration search, and classification using both GCN and GAT. Graph edges were established by identifying nearest neighbors under cosine similarity and optionally retaining only those connections that exceed a percentile-based threshold. Because the graph topology is selected empirically rather than fixed a priori, the resulting structure adapts to the characteristics of each feature extractor and dataset under the transductive protocol adopted throughout this study. The complete procedure, from graph construction and model training to prediction and performance evaluation, is summarized in [Algorithm 5](#).

Algorithm 5. GNN-based classification using GCN and GAT

Input: CNN feature datasets

$D_m = \{(X_{tr}, y_{tr}), (X_{va}, y_{va}), (X_{te}, y_{te})\}$, neighborhood size k , and similarity quantile q .

Output: Trained GCN and GAT models, predictions, graph metadata, and classification metrics for transductive and inductive settings.

- 1 **procedure** GNN-CLASSIFICATION(D, k, q)
 - 2 $M \leftarrow$ set of CNN feature datasets
-

(Cont'd...)

Algorithm 5. (Continued)

Algorithm 5. GNN-based classification using GCN and GAT

```

3   $R \leftarrow \{\text{transductive, inductive}\}$ 
4   $A \leftarrow \{\text{GCN, GAT}\}$ 
5  for each feature dataset  $m$  in  $M$  do
6    Load training, validation, and test feature-label pairs from  $D_m$ 
7    for each learning setting  $r$  in  $R$  do
8      if  $r = \text{transductive}$  then
9        Fit StandardScaler on  $X_{tr} \cup X_{va} \cup X_{te}$  and transform all features
10        $G \leftarrow \text{BUILD-GRAPH}([X_{tr}; X_{va}; X_{te}], k, q)$ 
11       Define training, validation, and test node masks on  $G$ 
12        $G_{tr} \leftarrow G, G_{va} \leftarrow G, G_{te} \leftarrow G$ 
13     else
14       Fit StandardScaler on  $X_{tr}$  only; transform  $X_{va}$  and  $X_{te}$  with the same scaler
15        $G_{tr}, \tau, I_{tr} \leftarrow \text{BUILD-GRAPH}(X_{tr}, k, q)$ 
16        $G_{va} \leftarrow$  attach validation nodes to  $G_{tr}$  using  $I_{tr}, \tau$ , and Train  $\rightarrow$  Validation edges
17        $G_{te} \leftarrow$  attach test nodes to  $G_{tr}$  using  $I_{tr}, \tau$ , and Train  $\rightarrow$  Test edges
18       If no candidate passes  $\tau$ , connect the unseen node to its most similar training node
19     end if
20    for each graph model  $a$  in  $A$  do
21      Initialize  $a$  with feature dimension  $d$  and  $C$  output classes
22      for epoch = 1 to maximum epochs do
23         $Z_{tr} \leftarrow a(G_{tr})$ 
24        Compute cross-entropy loss using labeled training nodes only
25        Update model parameters with the Adam optimizer
26        Evaluate macro-F1 on validation nodes in  $G_{va}$ 
27        Retain the model state with the highest validation macro-F1
28        Stop early when validation macro-F1 does not improve within the patience limit
29      end for
30      Restore the retained model state and predict test nodes in  $G_{te}$ 
31      Compute accuracy, macro precision, macro recall, macro-F1, and macro ROC-AUC
32      Store the model, predictions, graph metadata, and performance metrics
33    end for
34  end for

```

(Cont'd...)

Algorithm 5. (Continued)

Algorithm 5. GNN-based classification using GCN and GAT

```

35  end for
36  return all experiment results
37  end procedure
38
39  procedure BUILD-GRAPH( $X, k, q$ )
40    Find  $k$  Euclidean nearest-neighbor candidates for each node
41    Compute cosine similarity for every candidate node pair
42     $\tau \leftarrow q$ th percentile of candidate similarities
43    Retain candidate pairs whose cosine similarity is at least  $\tau$ 
44    Connect every isolated node to its most similar candidate
45    Convert retained pairs into bidirectional edges
46    return graph  $G = (V, E, X)$ , threshold  $\tau$ , and neighbor index  $I$ 
47  end procedure

```

Feature standardization:

$$\hat{x}_i = (x_i - \mu) / \sigma \quad (4)$$

Cosine similarity and edge selection:

$$s_{ij} = \left(\hat{x}_i^T \hat{x}_j \right) / \left(\|\hat{x}_i\|_2 \|\hat{x}_j\|_2 \right), E = \{(i, j) \mid j \in N_k(i) \wedge s_{ij} \geq \tau q\} \quad (5)$$

Graph definitions for the two learning settings:

$$\begin{aligned}
G_{trans} &= (V_{tr} \cup V_{va} \cup V_{te}, E_{all}, X_{all}), \\
G_{ind, tr} &= (V_{tr}, E_{tr}, X_{tr}) \\
E_{ind, eval} &= E_{tr} \cup \{(i, j) \mid i \in V_{tr} \wedge j \in V_{eval}\}
\end{aligned} \quad (6)$$

GNN output and training objective:

$$Z = a(G; \theta), L_{train} = -\left(1/|V_{tr}|\right) \sum_{i \in V_{tr}} y_i^T \log(\text{softmax}(z_i)) \quad (7)$$

where:

- (i) X denotes CNN feature matrices.
- (ii) V and E denote graph nodes and edges.
- (iii) $N_k(i)$ is the k -nearest-neighbor set of node i .
- (iv) s_{ij} is cosine similarity.
- (v) τq is the q th-percentile similarity threshold.
- (vi) a is either GCN or GAT.
- (vii) Z contains class logits.
- (viii) V_{tr} , V_{va} , and V_{te} denote training, validation, and test nodes, respectively.

In the transductive setting, all nodes share one graph, but only training labels optimize the model. In the inductive setting, scaling, threshold estimation, and the training graph use training data only; unseen nodes receive directed messages from training nodes during evaluation.

3.5. Performance evaluation metrics

The confusion matrix is a comprehensive evaluation framework used to compare model predictions with the actual categorical classifications.⁴¹ This matrix elucidates the extent to which the data are accurately classified, as well as the degree to which erroneous classifications have occurred among the various categories. Table 2 presents the confusion matrix for the six classes in the MSLD v2.0 dataset.

Accuracy is a critical metric for evaluating the efficacy of a classification model. Accuracy can be computed from the confusion matrix. Specifically, the accuracy for a multi-class classification problem can be determined by dividing the sum of all diagonal values in the confusion matrix by the total number of data instances.⁴¹ Equation 8 delineates the methodology for calculating the relevant accuracy for the classification task associated with the MSLD v2.0 dataset.

$$Accuracy = \frac{C_{11} + C_{22} + C_{33} + C_{44} + C_{55} + C_{66}}{\sum_{i=1}^6 \sum_{j=1}^6 C_{ij}} \quad (8)$$

where:

- (i) C_{ij} represents the element of the confusion matrix, where i denotes the actual class and j denotes the predicted class.
- (ii) i represents the index of the actual class, with $i = 1, 2, 3, 4, 5, 6$.

- (iii) j represents the index of the predicted class, with $j = 1, 2, 3, 4, 5, 6$.
- (iv) $C_{11}, C_{22}, C_{33}, C_{44}, C_{55}$, and C_{66} represent the diagonal elements of the confusion matrix, which indicate the number of correctly classified samples for each class.
- (v) $\sum_{i=1}^6 \sum_{j=1}^6 C_{ij}$ represents the total number of samples in the confusion matrix, including both correctly and incorrectly classified samples.
- (vi) The value “6” represents the total number of classes used in this study.

While accuracy is a useful measure, it has inherent limitations when applied to unbalanced datasets; in such cases, the accuracy metric may yield inflated values due to the predominant correct predictions of the majority class, thereby obscuring the identification of the minority class.

An additional evaluative metric incorporated in this study is the F1-score. The F1-score represents an evaluative measure that amalgamates precision and recall through the application of a harmonic mean, thereby offering a more equitable evaluation of the model's capacity to generate accurate predictions in conjunction with its ability to identify all actual instances from a given class.⁴¹ The operational principle of the F1-score is to balance precision and recall.

Precision is defined as the ratio of true positive predictions to the total number of instances predicted as a given class, with its computational expression detailed in Equation 9⁴¹:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} = \frac{C_{ii}}{\sum_{k=1, k \neq i}^6 C_{ki}} \quad (9)$$

Table 2. Confusion matrix for the Mpox Skin Lesion Dataset Version 2.0 dataset

Actual/predicted	Chickenpox	Cowpox	Healthy	Hand, Foot, and Mouth Disease	Measles	Mpox
Chickenpox	C11 ^a	C12	C13	C14	C15	C16
Cowpox	C21	C22 ^a	C23	C24	C25	C26
Healthy	C31	C32	C33 ^a	C34	C35	C36
HFMD	C41	C42	C43	C44 ^a	C45	C46
Measles	C51	C52	C53	C54	C55 ^a	C56
Mpox	C61	C62	C63	C64	C65	C66 ^a

Note: ^aDiagonal elements of the confusion matrix, representing the number of correctly classified samples for each class.

where:

- (i) $Precision_i$ represents the precision value for class i .
- (ii) TP_i represents the number of true positive samples for class i .
- (iii) FP_i represents the number of false positive samples for class i .
- (iv) C_{ii} represents the diagonal element of the confusion matrix, which indicates the number of correctly classified samples for class i .
- (v) C_{ki} represents the number of samples from actual class k that are predicted as class i .
- (vi) $\sum_{k=1}^6 C_{ki}$ represents the total number of samples predicted as class i .
- (vii) The value “6” represents the total number of classes used in this study.

To obtain an overall precision value across all six classes, macro precision is calculated by averaging the precision values of all classes, as expressed in **Equation 10**:

$$Macro\ precision = \frac{1}{6} \sum_{i=1}^6 precision_i \quad (10)$$

where:

- (i) $Macro\ precision$ represents the average precision across all classes.
- (ii) $precision_i$ represents the precision value for class i .
- (iii) i represents the class index, with $i = 1, 2, 3, 4, 5, 6$.
- (iv) The value “6” represents the total number of classes used in this study.

Recall refers to the ratio of actual instances of a class that have been correctly identified⁴¹. **Equation 11** provides a formulation for computing recall for a specific class.

$$Recall_i = \frac{TP_i}{TP_i + FN_i} = \frac{C_{ii}}{\sum_{k=1, k \neq i}^6 C_{ik}} \quad (11)$$

where:

- (i) $Recall_i$ represents the recall value for class i .
- (ii) TP_i represents the number of true positive samples for class i .
- (iii) FN_i represents the number of false negative samples for class i .
- (iv) C_{ii} represents the diagonal element of the confusion matrix, which indicates the number of correctly classified samples for class i .
- (v) C_{ik} represents the number of samples from actual class i that are predicted as class k .
- (vi) $\sum_{k=1}^6 C_{ik}$ represents the total number of actual samples belonging to class i .

- (vii) The value “6” represents the total number of classes used in this study.

To obtain an overall recall value across all six classes, macro recall is calculated by averaging the recall values of all classes, as expressed in **Equation 12**:

$$Macro\ recall = \frac{1}{6} \sum_{i=1}^6 recall_i \quad (12)$$

where:

- (i) $Macro\ recall$ represents the average recall across all classes.
- (ii) $recall_i$ represents the recall value for class i .
- (iii) i represents the class index, with $i = 1, 2, 3, 4, 5, 6$.
- (iv) The value “6” represents the total number of classes used in this study.

Equation 13 illustrates the procedure for calculating the F1-score for each class i where $i \in \{chickenpox, cowpox, healthy, HFMD, measles, mpox\}$.

$$F1_i = 2 \times \frac{precision_i \times recall_i}{precision_i + recall_i} \quad (13)$$

where:

- (v) $F1_i$ represents the F1-score value for class i .
- (vi) $precision_i$ represents the precision value for class i .
- (vii) $recall_i$ represents the recall value for class i .
- (viii) The F1-score is the harmonic mean of precision and recall, which provides a balanced evaluation of the model performance for each class.

To derive the average F1-score across the six classes, the macro F1-score is calculated as defined in **Equation 14**:

$$Macro\ F1 = \frac{1}{6} \sum_i F1_i \quad (14)$$

where:

- (i) $Macro\ F1$ represents the average F1-score across all classes.
- (ii) $F1_i$ represents the F1-score value for class i .
- (iii) i represents the class index, with $i = 1, 2, 3, 4, 5, 6$.
- (iv) The value “6” represents the total number of classes used in this study.

Accuracy, as defined in **Equation 15**, measures the overall proportion of correctly classified samples, while macro F1-score (**Equation 14**) evaluates the balance between precision and recall by assigning equal importance to each class. As a complementary metric, the area under the receiver operating characteristic curve (ROC-AUC)

was employed to assess the model’s discriminative ability across different classification thresholds. For this six-class problem, the class-wise ROC-AUC was calculated using the one-versus-rest (OvR) strategy as defined in **Equation 15**, and the resulting AUC values were averaged to obtain the macro ROC-AUC using the OvR strategy, as expressed in **Equation 16**.

$$AUC_i^{OvR} = \int_0^1 TPR_i(FPR_i) dFPR_i \quad (15)$$

where:

- (i) AUC_i^{OvR} represents the ROC-AUC value for class i using the OvR strategy.
- (ii) TPR_i represents the true positive rate for class i .
- (iii) FPR_i represents the false positive rate for class i .
- (iv) i represents the class index, with $i = 1, 2, 3, 4, 5, 6$.

$$Macro AUC_{OvR} = \frac{1}{6} \sum_{i=1}^6 AUC_i^{OvR} \quad (16)$$

where:

- (i) $Macro AUC_{OvR}$ represents the average ROC-AUC across all classes using the OvR strategy.
- (ii) AUC_i^{OvR} represents the ROC-AUC value for class i .
- (iii) i represents the class index, with $i = 1, 2, 3, 4, 5, 6$.
- (iv) The value “6” represents the total number of classes used in this study.

4. Results

4.1. Image dataset

The initial phase entails resizing the complete images, followed by the application of the data augmentation procedure. The dimensions for resizing the data are set to 224×224 pixels. The data augmentation procedure was applied exclusively to the training dataset. In this investigation, the dataset was partitioned into training, testing, and validation subsets, comprising 80%, 10%, and 10% of the total data, respectively, as shown in **Table 3**.

The data augmentation methodology employed in this study was implemented using Python and the libraries PyTorch, PyG, timm, sklearn, umap, and OpenCV. The geometric transformation techniques for data augmentation were applied using various methods, including rotation, horizontal and vertical flipping, cropping, zooming, and shearing. The outcomes of data augmentation via geometric transformations are presented in **Table 4**.

The data augmentation process employing HSV transformation is presented in **Table 4**. The parameters

applied during this process include brightness, contrast, and saturation values ranging from 0.7 to 1.3, while the hue is adjusted between -30° and 30° . Consequently, three distinct image datasets were produced, with their specifications detailed in **Table 5**.

4.2. Structured data

The subsequent stage is dedicated to feature extraction, during which the images were transformed into structured data. This process employed eight CNN-based algorithms, each using a distinct architecture: EfficientNetB4, AlexNet, VGG16, ResNet50, DenseNet121, GoogleNetv3/InceptionV3, MobileNetv2, and LeNet. Each architectural framework was characterized by a standardized array of layers, configurations, and parameter values. The resultant structured data were obtained from the designated target layer, as presented in **Table 6**. Furthermore, the number of features in the resulting structured data is shown in **Table 6**.

The three datasets previously referenced in **Table 5** were processed using each CNN architecture detailed in **Table 6** to extract features, resulting in a total of 24 structured datasets.

4.3. Model performance

The third phase encompasses the classification process, which entails examining 24 generated structured datasets to construct a classification model using two distinct classification algorithms, specifically GCN and GAT. However, it is imperative to note that both algorithms accept only graph-structured inputs. Thus, a graph must first be generated for each structured dataset.

Table 3. Composition of train, validation, and test data division

Class	Train data	Validation data	Test data
Chickenpox	60	8	7
Cowpox	53	7	6
Healthy	91	11	12
HFMD	129	16	16
Measles	44	6	5
Mpox	227	28	29
Total	604	76	75

Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

Table 4. Results of data augmentation on train data

Class	Train (original)	Train (geometric transformation)	Train (geometric and HSV transformation)	Validation	Test
Chickenpox	60	250	6,250	8	7
Cowpox	53	250	6,250	7	6
Healthy	91	250	6,250	11	12
HFMD	129	250	6,250	16	16
Measles	44	250	6,250	6	5
Mpox	227	250	6,250	28	29
Total	604	1,500	37,500	76	75

Abbreviations: HFMD: Hand, Foot, and Mouth Disease; HSV: Hue–saturation–value.

Table 5. Three Mpox Skin Lesion Dataset Version 2.0 datasets used in this research

Dataset code	Number of images	Description
MSLD v2.0-O	755	Original imbalanced dataset
MSLD v2.0 -G	1,651	Dataset balanced using geometric augmentation
MSLD v2.0-GC	37,651	Dataset balanced using geometric and color space augmentation

Abbreviation: MSLD v2.0: Mpox Skin Lesion Dataset Version 2.0.

Table 6. Target layer and the number of features from each convolutional neural network architecture

Model	Target layer for feature extraction	Number of features
LeNet	First fully connected layer	120
AlexNet	First fully connected layer (after 6×6 adaptive average pooling)	4,096
VGG16	Global average pooling after final convolutional block	512
GoogleNetV3	Global average pooling before classifier	2,048
ResNet50	Global average pooling/avgpool	2,048
DenseNet121	Global average pooling after final dense block	1,024
MobileNetV2	Global average pooling after final convolutional block	1,280
EfficientNetB4	Global average pooling after final MBConv block	1,792

To facilitate graph creation, the KNN method was employed to identify candidate nodes, and cosine similarity was used both to rank those candidates and to weight the resulting edges. Rather than fixing the graph topology in advance, the number of neighbors was varied over {5, 10, 15, 20}, the similarity percentile filter over {0, 80, 90}, and the mutual KNN constraint over {False, True}, so that each structured dataset yielded 24 candidate graph configurations for every classifier. The most suitable topology was therefore determined empirically for each combination rather than imposed uniformly.

Upon establishing the candidate graphs for each structured dataset, models employing both GCN and GAT were formulated, along with their corresponding configurations and parameters, as presented in [Table 7](#). Every candidate configuration was trained for a maximum of 300 epochs with early stopping after 50 epochs without improvement, and the configuration attaining the highest macro-averaged F1-score on the validation set was retained as the representative model for that combination.

From the 24 structured datasets generated in the preceding phase, which serve as inputs for these two algorithms, a total of 48 models were produced. The development of GCN and GAT models was conducted using the training and validation datasets. After identifying the model that yielded the optimal performance, a final evaluation was conducted on the test dataset. The outcomes of this final performance assessment are presented in [Table 8](#).

4.4. Highest models' performance

The highest-performing models, ranked by macro F1-score, are presented in [Table 9](#). For the original dataset

Table 7. Configuration and parameters for the GCN and GAT algorithms

Component	Parameter	GCN	GAT
Graph construction	Similarity metric	Cosine (exact)	Cosine (exact)
	Neighbor selection	Cosine KNN	Cosine KNN
	Number of neighbors (k)	{5, 10, 15, 20} — searched	{5, 10, 15, 20} — searched
	Similarity percentile filter	{0, 80, 90} — searched	{0, 80, 90} — searched
	Mutual KNN constraint	{False, True} — searched	{False, True} — searched
	Edge weight	Cosine similarity	Cosine similarity
	Edge direction	Undirected	Undirected
	Search space size	24 combinations	24 combinations
Architecture	Convolution operator	GCNConv	GATConv
	Number of layers	2	2
	Hidden dimension	256	16 per head (64 total)
	Attention heads	—	4 (1 in output layer)
	Activation	ReLU	ELU
	Dropout	0.10	0.05
	LayerNorm	No	No
	Residual connection	No	No
Training	Output dimension	6 (number of classes)	6 (number of classes)
	Optimizer	Adam	Adam
	Learning rate	0.001	0.001
	Weight decay	1×10^{-5}	1×10^{-5}
	Loss function	Cross-entropy	Cross-entropy
	Class weighting	No	No
	Label smoothing	0.0	0.0
	LR scheduler	None	None
	Maximum epochs	300	300
	Early-stopping patience	50	50
	Model selection criterion	Best validation F1-macro	Best validation F1-macro
	Learning setting	Transductive	Transductive
	Random seed	42	42

Abbreviations: ELU: Exponential linear unit; GAT: Graph attention network; GCN: Graph convolutional network; KNN: K-nearest neighbor; ReLU: Rectified linear unit.

Table 8. Experiment results

Dataset	Feature extraction	Classification	Accuracy	Macro F1-score	Macro ROC-AUC OvR	Macro precision	Macro recall
MSLD v2.0-O	EfficientNetB4	GCN	76.00	68.91	92.46	69.81	68.57
	AlexNet		80.00	76.73	95.74	80.01	75.70
	VGG16		84.00	80.24	97.88	82.92	79.17
	ResNet50		86.67	80.32	95.74	82.36	79.02
	DenseNet121		84.00	80.44	93.50	91.54	77.02
	GoogleNetV3		70.67	62.89	91.26	65.53	62.34
	MobileNetV2		82.67	81.04	97.69	80.79	82.34
	LeNet		44.00	33.99	74.01	33.47	34.75
	EfficientNetB4	GAT	76.00	65.94	91.18	67.37	67.37
	AlexNet		72.00	66.22	92.88	68.33	65.88
	VGG16		81.33	77.97	97.47	81.27	75.82
	ResNet50		80.00	73.88	95.23	74.87	76.63
	DenseNet121		78.67	74.66	94.28	80.02	71.49
	GoogleNetV3		68.00	62.62	88.01	64.41	63.95
	MobileNetV2		85.33	84.01	96.59	82.35	86.55
	LeNet		42.67	31.77	72.29	34.80	30.81
MSLD v2.0-G	EfficientNetB4	GCN	74.67	68.75	96.99	72.26	67.48
	AlexNet		80.00	77.78	95.87	79.82	76.63
	VGG16		86.67	86.04	98.03	84.52	87.97
	ResNet50		84.00	77.05	97.21	84.82	72.84
	DenseNet121		86.67	82.62	97.01	85.86	81.25
	GoogleNetV3		82.67	81.55	95.58	86.46	78.38
	MobileNetV2		85.33	85.91	97.61	84.03	89.13
	LeNet		26.67	25.94	69.09	27.62	28.62
	EfficientNetB4	GAT	80.00	75.89	95.87	76.37	77.40
	AlexNet		69.33	67.23	92.75	65.51	71.58
	VGG16		82.67	80.64	98.37	80.28	81.82
	ResNet50		82.67	81.37	94.27	79.54	84.06
	DenseNet121		85.33	83.36	97.08	81.50	86.49
	GoogleNetV3		72.00	71.16	92.61	73.41	75.58
	MobileNetV2		90.67	89.25	98.48	89.28	90.27
	LeNet		24.00	24.05	69.66	29.61	26.19

(Cont'd...)

Table 8. (Continued)

Dataset	Feature extraction	Classification	Accuracy	Macro F1-score	Macro ROC-AUC OvR	Macro precision	Macro recall
MSLD v2.0-GC	EfficientNetB4	GCN	80.00	73.69	81.58	91.19	69.65
	AlexNet		80.00	79.25	95.90	81.76	79.81
	VGG16		92.00 ^a	91.55 ^a	98.81 ^a	93.54 ^a	90.32 ^a
	ResNet50		81.33	78.49	90.35	92.76	72.00
	DenseNet121		86.67	85.28	93.37	88.37	83.12
	GoogleNetV3		76.00	75.54	92.40	77.95	74.51
	MobileNetV2		77.33	77.48	96.14	76.58	82.03
	LeNet		30.67	28.17	67.82	29.82	32.22
	EfficientNetB4	GAT	78.67	69.52	87.30	74.52	67.27
	AlexNet		77.33	75.10	95.66	75.05	77.84
	VGG16		88.00	86.78	94.91	87.12	86.50
	ResNet50		84.00	81.68	94.50	90.97	77.05
	DenseNet121		86.67	84.40	95.46	86.30	83.12
	GoogleNetV3		74.67	73.95	89.54	77.12	74.79
	MobileNetV2		76.00	76.61	92.55	75.51	81.46
	LeNet		34.67	35.40	69.71	37.01	46.73

Note: ^aThe performance value of the best-performing model.

Abbreviations: GAT: Graph attention network; GCN: Graph convolutional network; MSLD v2.0: Mpox Skin Lesion Dataset Version 2.0; OvR: One-versus-rest; ROC-AUC: Area under the receiver operating characteristic curve.

Table 9. Highest model's performance

Dataset	Classification	Feature extraction	Accuracy	Macro F1-score	Macro ROC-AUC OvR	Macro precision	Macro recall
MSLD v2.0-O	GCN	MobileNetV2	82.67	81.04	97.69	80.79	82.34
	GAT	MobileNetV2	85.33	84.01	96.59	82.35	86.55
MSLD v2.0-G	GCN	VGG16	86.67	86.04	98.03	84.52	87.97
	GAT	MobileNetV2	90.67	89.25	98.48	89.28	90.27
MSLD v2.0-GC	GCN	VGG16	92.00 ^a	91.55 ^a	98.81 ^a	93.54 ^a	90.32 ^a
	GAT	VGG16	88.00	86.78	94.91	87.12	86.50

Note: ^aRepresents the performance value of the best-performing model.

Abbreviations: GAT: Graph attention network; GCN: Graph convolutional network; MSLD v2.0: Mpox Skin Lesion Dataset Version 2.0; OvR: One-versus-rest; ROC-AUC: Area under the receiver operating characteristic curve.

(MSLD v2.0-O), the best GCN and GAT configurations both used MobileNetV2 as the feature extractor, achieving macro F1-scores of 81.04% and 84.01%, respectively. For the geometrically augmented dataset (MSLD v2.0-G), the best GCN model used VGG16 and achieved a macro F1-score of 86.04%, whereas the best GAT model used MobileNetV2 and achieved a macro F1-score of 89.25%. The overall best-performing configuration was obtained on the MSLD v2.0-GC dataset using GCN with VGG16 feature extraction, achieving an accuracy of 92.00%, a macro F1-score of 91.55%, a macro ROC-AUC OvR of 98.81%, a macro precision of 93.54%, and a macro recall of 90.32%.

For the best models using the MSLD v2.0-O dataset, [Figures 6–11](#) present the confusion matrices for the models employing the MobileNetV2 feature extractor with GCN and GAT.

5. Discussion

5.1. Comparison of GCN and GAT performance

The confusion matrices of the best-performing GAT and

GCN models are presented in [Table 9](#). As shown in [Table 9](#), the GAT model using MobileNetV2 on the MSLD v2.0-G dataset achieved an accuracy of 90.67% and a macro F1-score of 89.25%. The model correctly classified all Cowpox and Measles samples and most HFMD and Mpox cases, although several Chickenpox samples were misclassified as Mpox, with a small number of Healthy and Mpox samples also confused with other classes. In comparison, the GCN model using VGG16 on the MSLD v2.0-GC dataset achieved higher overall performance, with an accuracy of 92.00% and a macro F1-score of 91.55%. This model perfectly classified Cowpox, HFMD, and Measles samples, while the remaining errors were mainly concentrated in Chickenpox and Mpox. Overall, the GCN–VGG16 configuration produced fewer misclassifications and a more consistent class-wise prediction pattern, indicating a stronger ability to distinguish visually similar skin lesion categories.

According to [Figure 12](#), the GCN generally demonstrated better average performance than the GAT across the experimental configurations. GCN achieved an average macro F1-score of 72.49% compared with 70.56%

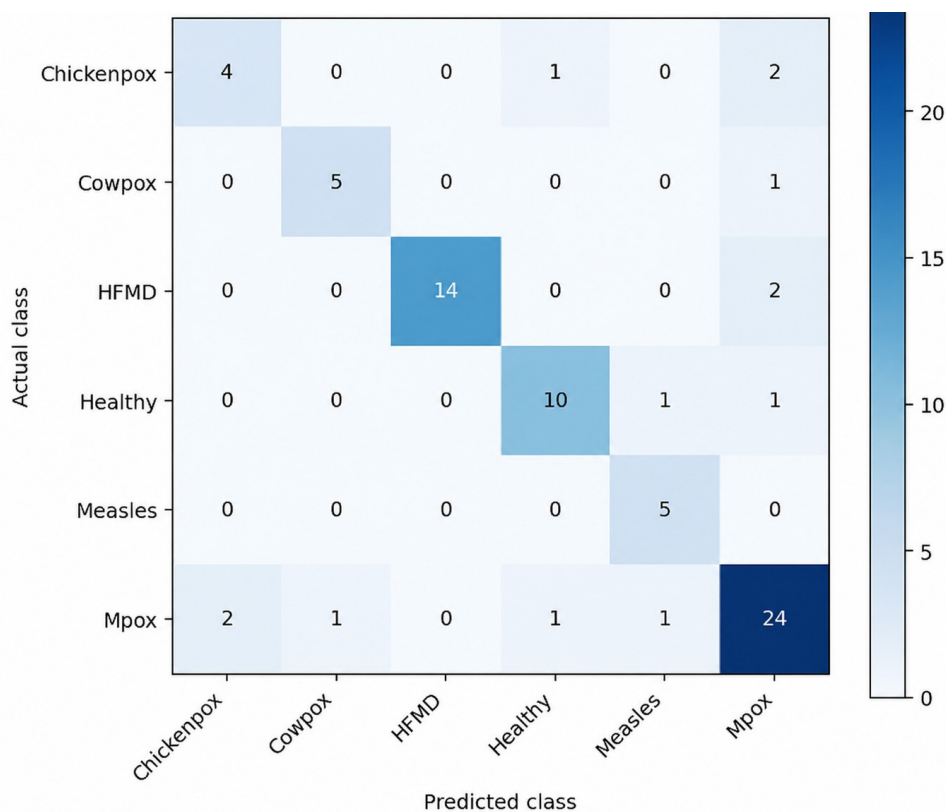


Figure 6. Confusion matrix of the model with MSLD v2.0-O, MobileNetV2, and a graph convolutional network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

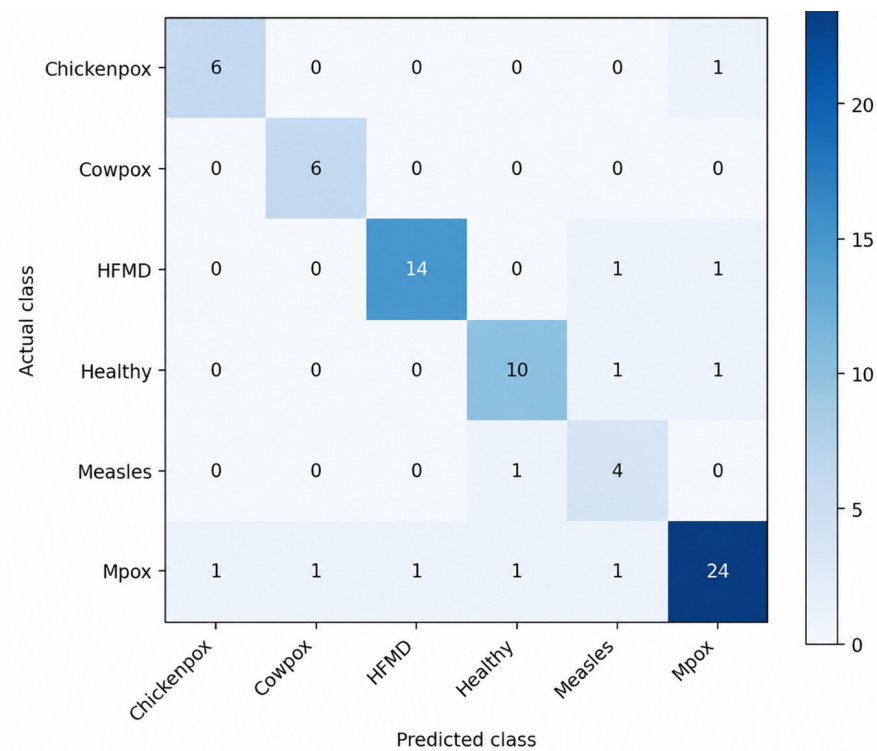


Figure 7. Confusion matrix of the model with MSLD v2.0-O, MobileNetV2, and graph attention network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

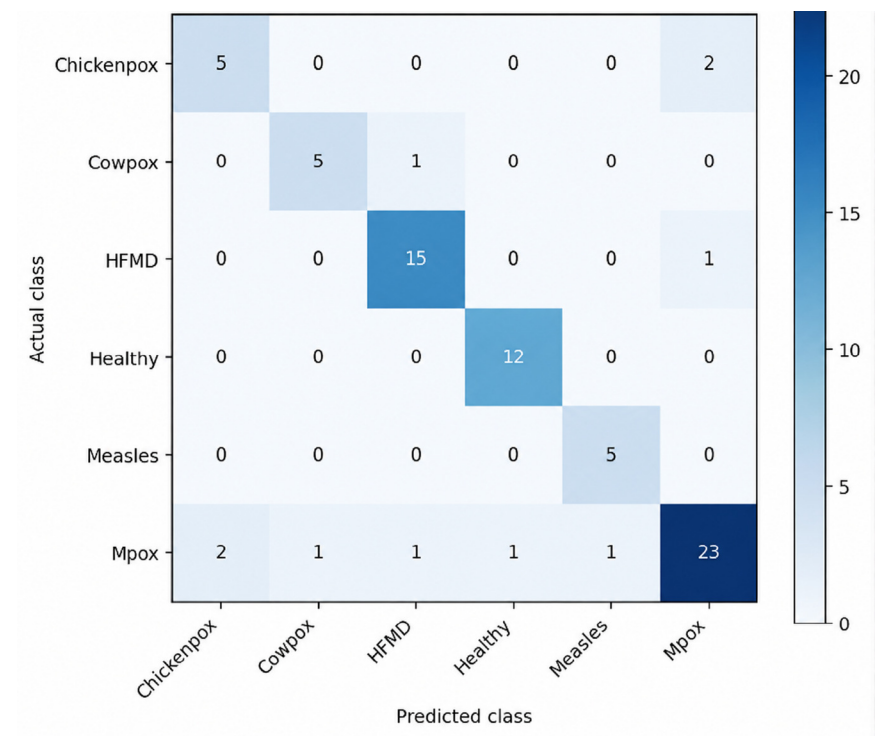


Figure 8. Confusion matrix of the model with MSLD v2.0-G, VGG16, and graph convolutional network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

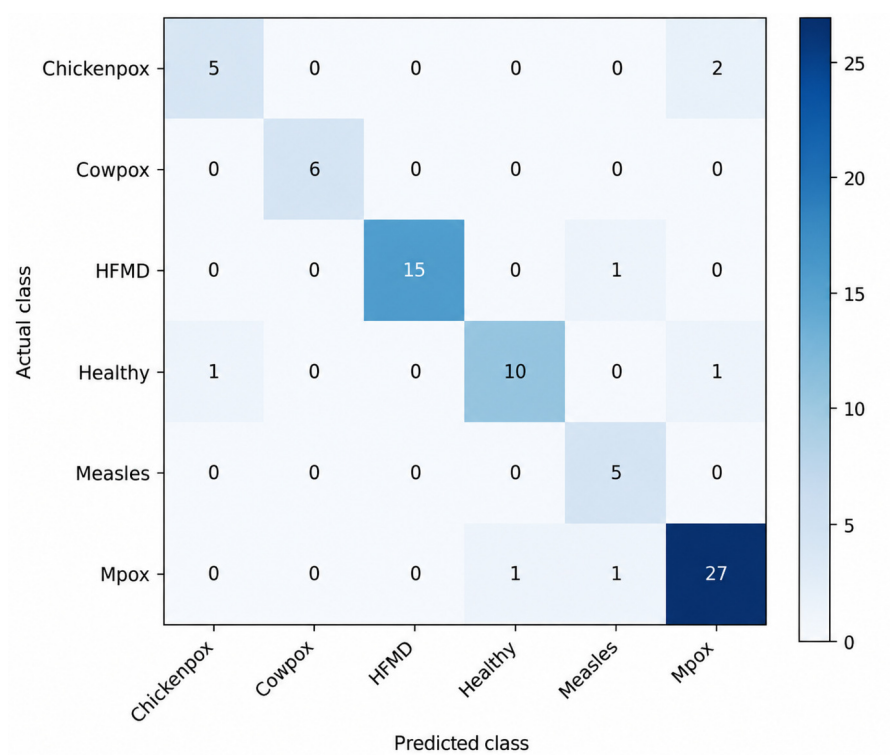


Figure 9. Confusion matrix of the model with MSLD v2.0-G, MobileNetV2, and graph attention network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

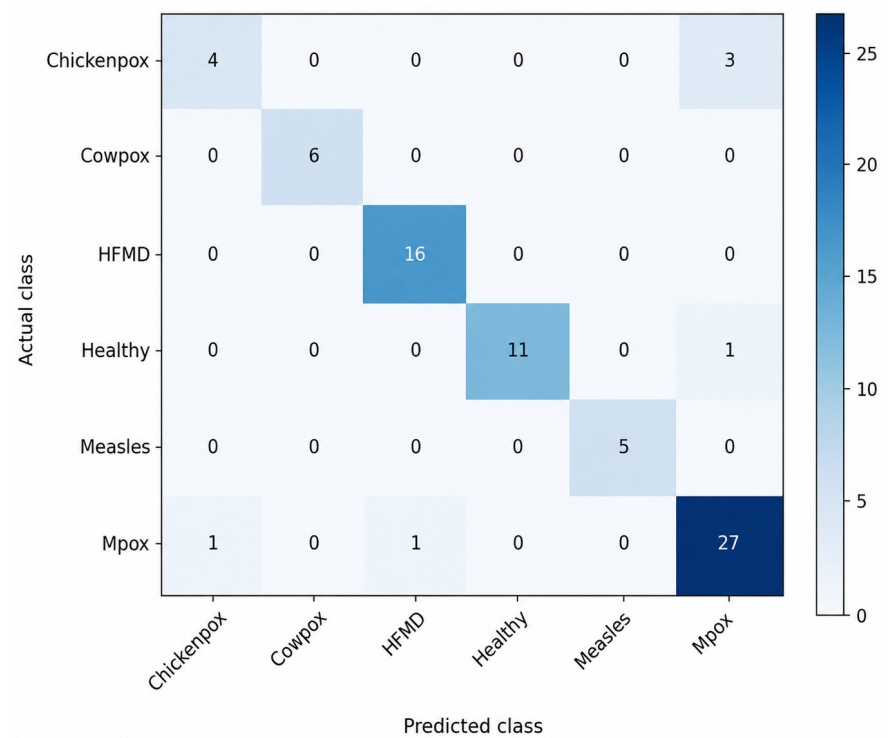


Figure 10. Confusion matrix of the model with MSLD v2.0-GC, VGG16, and graph convolutional network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

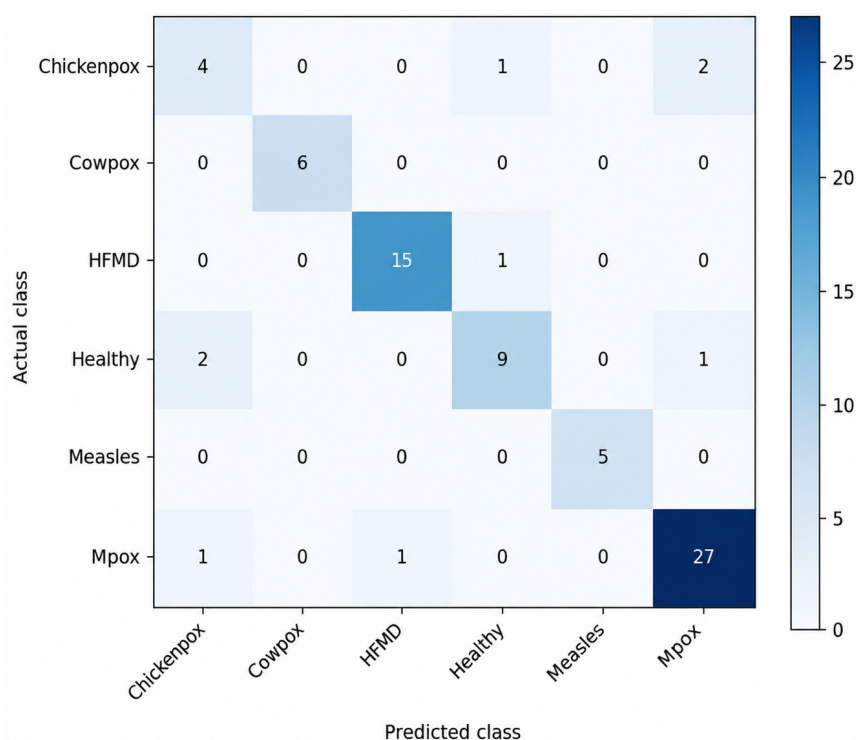


Figure 11. Confusion matrix of the model with MSLD v2.0-GC, VGG16, and graph attention network
Abbreviation: HFMD: Hand, Foot, and Mouth Disease.

for GAT. In addition, GCN obtained a slightly higher average macro ROC-AUC OvR of 91.75%, whereas GAT achieved 91.11%. These results indicate that GCN tended to provide better overall class-wise classification performance and slightly stronger discriminative ability across different decision thresholds.

A similar pattern was observed for macro precision, where GCN achieved an average value of 75.99%, compared with 72.19% for GAT, indicating that GCN produced fewer false-positive predictions across the six classes on average. In contrast, macro recall was nearly identical between the two classifiers, with GAT obtaining 71.94% and GCN 71.87%. This small difference suggests that both graph-based classifiers had comparable abilities to identify samples belonging to each class, although GCN provided a better balance between precision and recall, as reflected by its higher macro F1-score.

To determine whether these performance differences were statistically significant, an unpaired *t*-test was conducted between the 24 GCN and 24 GAT model results. No statistically significant differences were observed for macro F1-score ($t = 0.383$, $p = 0.703$), macro ROC-AUC OvR ($t = 0.253$, $p = 0.801$), macro precision ($t = 0.743$, $p = 0.461$), and macro recall ($t = -0.015$, $p = 0.988$). Since all

p-values were greater than 0.05, the observed advantage of GCN should be interpreted as an empirical performance trend rather than evidence of statistically significant superiority over GAT.

From a methodological perspective, the observed tendency toward better GCN performance may be attributed to its more stable information aggregation mechanisms. The GCN updates the representation of individual nodes by aggregating information from adjacent nodes, using a graph structure derived from the similarity of features extracted by CNNs. This mechanism facilitates the transmission of information among images that exhibit analogous visual characteristics, thereby making node representations more informative for classification.¹⁸ This methodology is particularly suitable for skin lesion imagery, given that certain classes within the MSLD v2.0 exhibit visual patterns that closely resemble one another.

Conversely, the GAT employs an attention mechanism to allocate different weights to each neighboring node. This mechanism is theoretically more adaptable, as the model can determine which neighboring nodes are most relevant for representation updates. However, in datasets characterized by limited sample sizes and unbalanced class distributions, the attention weights may become

increasingly sensitive to feature variation and noise.⁴² This scenario can result in a less stable learning process for GAT than for GCN. Therefore, while GAT possesses adaptive capabilities for selecting relevant neighboring information, the findings of these experiments suggest that GCN may provide more stable utilization of graph-structured information for classifying skin lesion images within the MSLD v2.0 dataset.

5.2. Performance of different convolutional neural network-based feature extractors

Based on Figure 13, VGG16 demonstrated the strongest overall performance among the eight CNN-based feature extraction methods, achieving the highest average macro F1-score of 83.87% and the highest macro ROC-AUC OvR of 97.58%, followed by MobileNetV2 (macro F1-score = 82.38%) and DenseNet121 (macro F1-score = 81.79%). The performance pattern varied across the other macro-averaged metrics. DenseNet121 achieved the highest average macro precision of 85.60%, indicating higher precision in its class-wise predictions, whereas MobileNetV2 achieved the highest macro recall of 85.30%, indicating stronger sensitivity in identifying samples from each diagnostic class. VGG16 maintained a comparatively balanced performance, with a macro precision of 84.94% and a macro recall of 83.60%, which contributed to its highest overall macro F1-score.

Figure 13 presents the average performance comparison of various CNN-based feature extraction architectures across all experimental configurations. Although the figure illustrates variations in performance among the eight CNN architectures, these differences represent descriptive trends based on averaged results and cannot directly determine statistical significance. Therefore, a one-way ANOVA was conducted to statistically evaluate whether the observed performance variations among the feature extraction architectures were significant.

To determine whether the differences among the eight feature extraction architectures were statistically significant, a one-way ANOVA was conducted. Significant differences were observed for macro F1-score ($F = 76.342$, $p < 0.001$), macro ROC-AUC OvR ($F = 59.833$, $p < 0.001$), macro precision ($F = 47.707$, $p < 0.001$), and macro recall ($F = 56.695$, $p < 0.001$). These findings indicate that the CNN architecture used for feature extraction had a significant effect on the downstream GCN- and GAT-based classification performance.

The strong performance of VGG16, DenseNet121, and MobileNetV2 can be associated with their ability to generate discriminative hierarchical feature representations from medical images. VGG16 progressively learns low- to

high-level visual patterns through repeated convolution and pooling operations, allowing it to capture lesion texture, boundaries, shape, and spatial appearance. DenseNet121 employs dense inter-layer connectivity that promotes feature reuse and facilitates information flow across the network, thereby helping preserve subtle visual characteristics. MobileNetV2, despite being designed as a lightweight architecture, uses depth-wise separable convolutions and inverted residual blocks that can retain informative feature representations while reducing computational complexity. Recent studies on skin-lesion classification have also demonstrated the effectiveness of VGG16 and DenseNet121 as transfer-learning backbones, while MobileNetV2 has shown strong performance in automated skin-disease classification.⁴³

These architectural characteristics are particularly relevant to mpox lesion classification because mpox and several visually similar dermatological conditions may share overlapping characteristics, while differing only in relatively subtle visual cues such as texture, lesion boundaries, morphology, color distribution, and local spatial patterns. In this context, VGG16 may benefit from its progressive hierarchical feature extraction, DenseNet121 from the preservation and reuse of earlier features, and MobileNetV2 from compact but discriminative representations. The complementary strengths of these architectures may therefore explain why they produced the most effective CNN-derived node representations for subsequent GCN and GAT processing in this study. This interpretation is also consistent with recent dermatological-image studies in which DenseNet121, VGG16, and MobileNet-based architectures achieved competitive classification performance.⁴³

In contrast, LeNet exhibited the lowest performance, with an average macro F1-score of only 29.89%, a macro ROC-AUC OvR of 70.43%, a macro precision of 32.06%, and a macro recall of 33.22%. This comparatively poor performance may be associated with the limited representational capacity of the classical LeNet architecture. LeNet has a relatively shallow structure with fewer convolutional and feature-transformation stages than modern CNN architectures, which may restrict its ability to learn sufficiently rich hierarchical representations for complex medical images. Recent studies have emphasized that effective medical and dermatological image classification requires feature representations capable of capturing subtle variations in texture, lesion boundaries, morphology, and spatial patterns.^{44,45} Consequently, the shallow feature hierarchy of LeNet may be insufficient for representing the complex visual characteristics of 224×224 mpox and other skin-lesion images. Therefore, its poor performance in this study

should be interpreted as a limitation of the classical LeNet architecture for this particular feature-extraction task rather

than as evidence that shallow neural networks are inherently unsuitable for image classification.⁴⁴

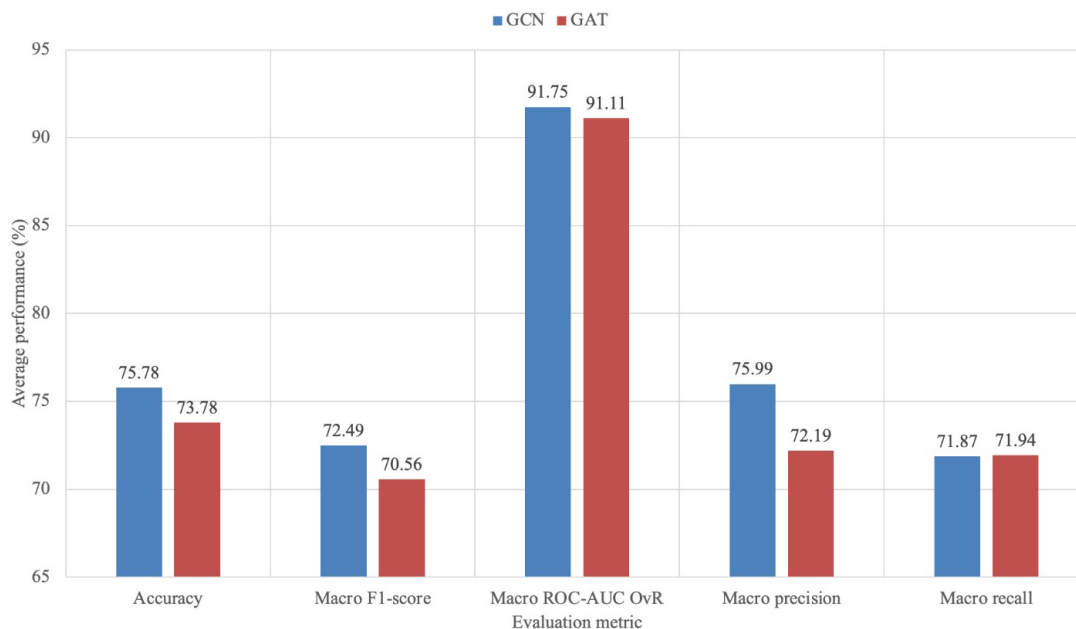


Figure 12. Average performance comparison between GCN and GAT

Abbreviations: GAT: Graph attention network; GCN: Graph convolutional network; OvR: One-versus-rest; ROC-AUC: Area under the receiver operating characteristic curve.

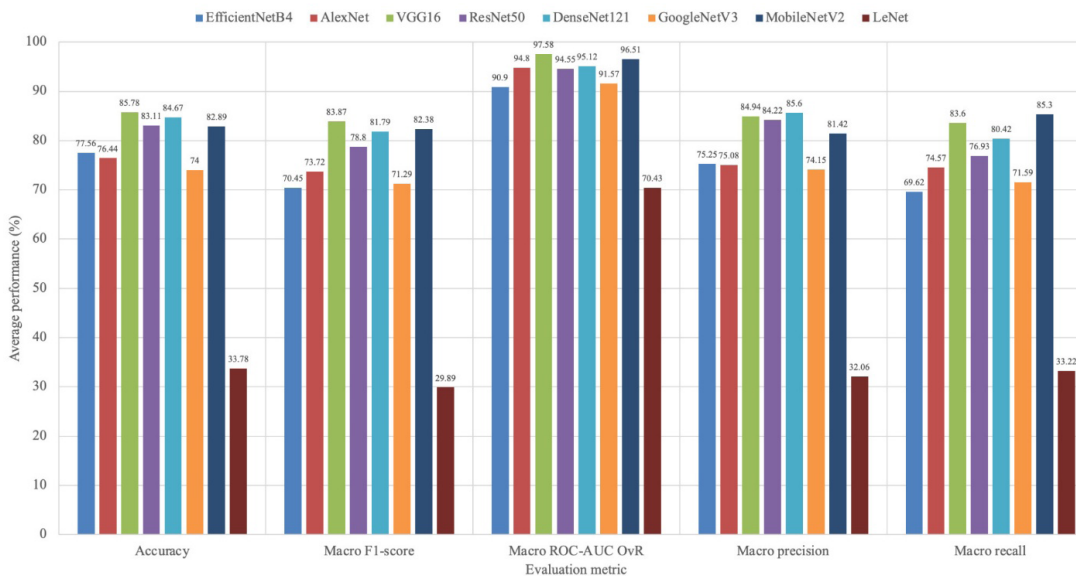


Figure 13. Average performance comparison between various CNN-based feature extraction methods

Abbreviations: OvR: One-versus-rest; ROC-AUC: Area under the receiver operating characteristic curve.

5.3. Effect of data balancing and augmentation

The confusion matrices illustrate the class-wise classification performance of the best-performing GCN and GAT configurations across the original and balanced datasets. On the original imbalanced MSLD v2.0-O dataset, both models showed relatively good recognition of the majority and better-represented classes but exhibited errors across several classes. The GCN–MobileNetV2 model correctly classified 4 out of 7 Chickenpox samples, 5 out of 6 Cowpox samples, 14 out of 16 HFMD samples, 10 out of 12 Healthy samples, all 5 Measles samples, and 24 out of 29 Mpox samples. The GAT–MobileNetV2 model performed better than GCN for Chickenpox and Cowpox, correctly classifying 6 out of 7 and all 6 samples, respectively. However, misclassifications remained for HFMD, Healthy, Measles, and Mpox, with two HFMD, two Healthy, one Measles, and five Mpox samples incorrectly classified.

Following geometric balancing in MSLD v2.0-G, the best-performing GAT–MobileNetV2 model showed improved recognition of several classes, particularly HFMD and Mpox, correctly classifying 15 out of 16 HFMD samples and 27 out of 29 Mpox samples. Recognition of Cowpox and Measles was also maintained at 100%, while Chickenpox remained more challenging, with 5 out of 7 samples correctly classified. After applying the combined geometric and HSV/color balancing strategy in MSLD v2.0-GC, the best-performing GCN–VGG16 model further improved class-wise recognition, correctly classifying all Cowpox, HFMD, and Measles samples, as well as 11 out of 12 Healthy samples and 27 out of 29 Mpox samples. However, Chickenpox remained the most challenging class, with only 4 out of 7 samples correctly classified, including three misclassified as Mpox.

Overall, balancing the training data was associated with improved recognition, particularly for HFMD, Cowpox, and Mpox, and reduced errors across several minority classes, although visual confusion between Chickenpox and Mpox remained evident. These class-wise improvements are also reflected in the aggregated performance across all CNN–GNN configurations, as summarized in Figure 14.

Consistent with the class-wise patterns observed in the confusion matrices, Figure 14 illustrates the average classification performance obtained across the three dataset configurations. The original imbalanced dataset, MSLD v2.0-O, produced the lowest average macro F1-score (68.85%) and macro recall (68.59%). After class balancing using geometric augmentation, MSLD v2.0-G increased the macro F1-score to 72.41% and macro recall to 73.48%. The combination of geometric and HSV/color augmentation in MSLD v2.0-GC produced the highest average macro

F1-score (73.31%), macro precision (77.22%), and macro recall (73.65%). These findings indicate that balancing the class distribution through augmentation generally improved the observed classification performance, particularly in terms of macro F1-score, macro precision, and macro recall, compared with the original imbalanced dataset.

The effect of augmentation differed across the evaluation metrics. MSLD v2.0-G achieved the highest macro ROC-AUC OvR of 92.91%, compared with 91.64% for MSLD v2.0-O and 89.75% for MSLD v2.0-GC. Thus, although the additional color space transformations were associated with higher average macro F1-score and macro precision, they did not further improve the overall class-discrimination ability measured by ROC-AUC. This finding suggests that geometric augmentation alone provided stronger separability across classification thresholds than the combined geometric and color space strategy, whereas the combined strategy tended to improve the balance between precision and recall.

Figure 14 presents the average performance comparison of the models across the three dataset configurations, namely MSLD v2.0-O, MSLD v2.0-G, and MSLD v2.0-GC. Although the figure illustrates performance variations among the three datasets, these differences represent descriptive trends based on averaged results and cannot directly determine statistical significance. Therefore, paired *t*-tests were conducted using the individual model performance values across the same CNN–GNN configurations to evaluate whether the observed improvements were statistically significant.

Paired *t*-tests across the same CNN–GNN configurations showed that the increase in macro F1-score from MSLD v2.0-O to MSLD v2.0-G was close to, but did not reach, statistical significance ($t = -2.110$, $p = 0.052$). In contrast, the difference between MSLD v2.0-O and MSLD v2.0-GC was statistically significant ($t = -2.830$, $p = 0.013$). No significant difference in macro F1-score was observed between MSLD v2.0-G and MSLD v2.0-GC ($t = -0.564$, $p = 0.581$). These findings indicate that balancing the original dataset through augmentation was associated with improved macro F1-score, with statistical significance observed for the combined geometric and HSV/color augmentation strategy but not for geometric augmentation alone.

A similar pattern was observed for macro recall and macro precision. Macro recall increased significantly from MSLD v2.0-O to both MSLD v2.0-G ($p = 0.012$) and MSLD v2.0-GC ($p = 0.012$), supporting the effectiveness of class balancing in improving recognition across the six classes. Macro precision was significantly higher for MSLD

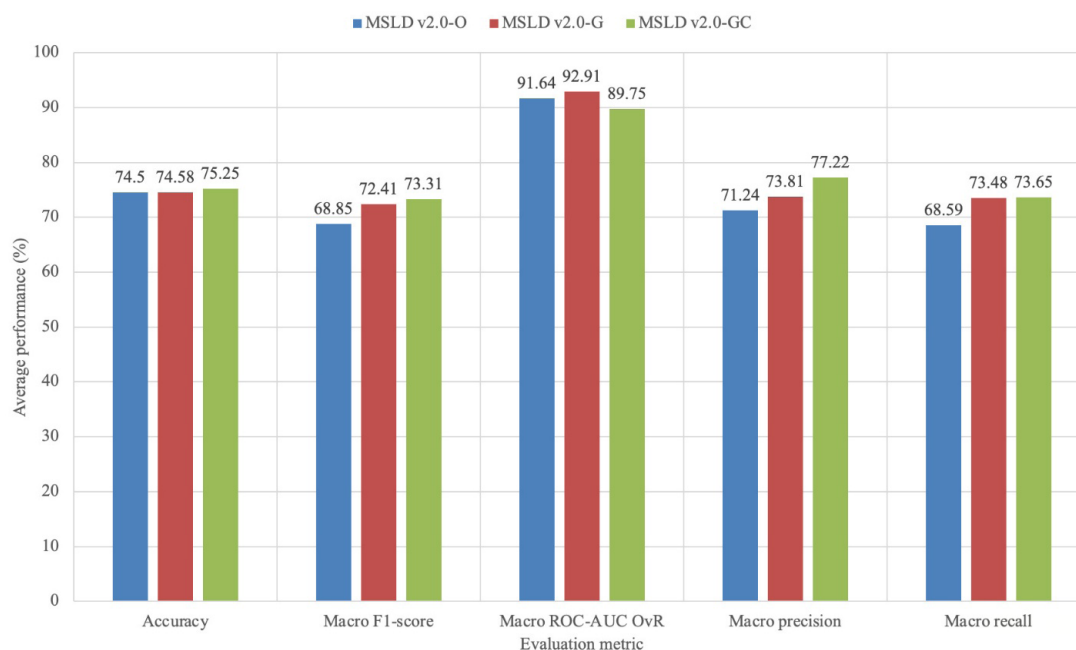


Figure 14. Average performance comparison by dataset: original (MSLD v2.0-O) and datasets from the data augmentation process (MSLD v2.0-G and MSLD v2.0-GC)

Abbreviations: MSLD v2.0: Mpox Skin Lesion Dataset 2.0; OvR: One-versus-rest; ROC-AUC: Area under the receiver operating characteristic curve.

v2.0-GC than for MSLD v2.0-O ($p = 0.009$), whereas the difference between MSLD v2.0-G and MSLD v2.0-GC was not statistically significant ($p = 0.117$). For macro ROC-AUC OvR, MSLD v2.0-G significantly outperformed MSLD v2.0-GC ($p = 0.012$), indicating that the additional color space transformations did not consistently improve discrimination across decision thresholds.

Taken together, these results suggest that geometric augmentation was particularly beneficial for class separability, whereas the combination of geometric and color space augmentation provided the strongest balance among macro F1-score, macro precision, and macro recall.

5.4. Comparison of performance between existing research

Table 10 compares the performance of the proposed model with previous studies that employed the MSLD v2.0 dataset for mpox classification. Since the previous studies primarily reported F1-score, the comparison in this section focuses on F1-based performance. The best-performing model in the present study was the GCN-VGG16 configuration using the MSLD v2.0-GC dataset, achieving a macro F1-score of 91.55%. The best GAT configuration, using MobileNetV2 features on the geometrically balanced MSLD v2.0-G dataset, achieved a macro F1-score of 89.25%.

Compared with previously reported deep-learning

approaches, the best GCN model achieved a numerically higher reported F1-score than several existing configurations. Vuran *et al.*¹¹ reported F1-scores of 93.58% for Swin Transformer, 92.75% for ViT, 89.89% for DINO, and 84.29% for MAE. Therefore, the proposed GCN model achieved a higher reported F1-score than DINO and MAE but a lower reported F1-score than Swin Transformer and ViT. In the study by Pal *et al.*¹³, a ViT-based approach using geometric and HSV augmentation achieved an F1-score of 92.97%, which was slightly higher than the 91.55% obtained by the proposed model. In the study by Huan *et al.*¹⁵, MSMP-Net achieved an F1-score of 90.24%, while Aboshosha *et al.*¹⁹ reported an F1-score of 83.30% for a weighted ensemble. Thus, the proposed GCN model achieved a higher reported F1-score than both MSMP-Net and the weighted ensemble approach.

A similar comparison can be made with the transfer-learning configurations reported in the study by Ali *et al.*¹⁴ The ImageNet-pretrained models achieved F1-scores ranging from 77% to 86%, with DenseNet121 providing the highest value of 86%, while the HAM10000-pretrained models achieved F1-scores ranging from 70% to 88%, with DenseNet121 again providing the highest value of 88%. The proposed GCN model, with a macro F1-score of 91.55%, therefore had a higher reported F1-score than all configurations reported in the study by Ali *et al.*¹⁴

Table 10. Performance comparison of the proposed models with existing studies

Reference	Year	Method			F1-score
		Data augmentation	Feature extraction	Classification	
11	2025	–	–	Swin Transformer	93.58
				ViT	92.75
				DINO	89.89
				MAE	84.29
12	2026	Geometric transformation	CNN	W-Random Forest	90.81
				W-L-SVM	95.92
				W-LightGBM	95.1
				W-Extra Trees	93.25
				W-XGBoost	93.01
				LDSCNN–ABBLM	98.28
13	2025	Geometric and HSV transformation	–	ViT	92.97
14	2024	Geometric and HSV transformation	–	IMAGENET + VGG16	77
				IMAGENET + ResNet50	82
				IMAGENET + DenseNet121	86
				IMAGENET + MobileNetV2	80
				IMAGENET + EfficientNetB3	78
				IMAGENET + InceptionV3	77
				IMAGENET + Xception	80
				HAM10000 + VGG16	70
				HAM10000 + ResNet50	79
				HAM10000 + DenseNet121	88
				HAM10000 + MobileNetV2	82
				HAM10000 + EfficientNetB3	87
				HAM10000 + InceptionV3	80
				HAM10000 + Xception	81
15	2024	Geometric and HSV transformation	–	MSMP-Net	90.24

(Cont'd...)

Table 10. (Continued)

Reference	Year	Method			F1-score
		Data augmentation	Feature extraction	Classification	
19	2026	Geometric transformation	—	Weighted Ensemble	83.3
				ConvNeXt-Tiny	81.6
				MobileNetV3-Large	80.5
				ViT-B/16	80.2
				DenseNet-121	80.0
				ResNet-50	79.5
				EfficientNet-B0	77.4
Present study		Geometric transformation with balancing data	MobileNetV2	GAT	89.25
			VGG16	GCN	91.55

Abbreviations: CNN: Convolutional neural network; GAT: Graph attention network; GCN: Graph convolutional network; HSV: Hue–saturation–value.

For machine-learning approaches using CNN-derived features, Gabriel *et al.*¹² reported F1-scores of 90.81% for weighted random forest, 95.92% for weighted linear support vector machine, 95.10% for weighted LightGBM, 93.25% for weighted Extra Trees, and 93.01% for weighted XGBoost. The LDSCNN–ABBLM framework in the same study achieved the highest reported F1-score of 98.28%. Accordingly, the proposed GCN model had a higher reported F1-score than weighted random forest but lower reported F1-scores than weighted linear support vector machine, weighted LightGBM, weighted Extra Trees, weighted XGBoost, and LDSCNN–ABBLM. The difference between LDSCNN–ABBLM and the best model in the present study was approximately 6.73 percentage points.

Although these comparisons indicate that the proposed graph-based framework is competitive with several existing approaches, they should be interpreted cautiously because the studies may differ in data partitioning, augmentation procedures, preprocessing strategies, model configurations,

and evaluation protocols. In addition, the present study reports macro F1-score, whereas the averaging strategy used for the F1-score was not consistently specified across all previous studies. The main contribution of this study lies not only in its predictive performance, but also in the systematic integration of explicit class balancing, multiple CNN feature extractors, and GCN/GAT-based relational classification within a unified multi-class mpox identification framework.

5.5. Limitations

This study has several limitations that should be considered when interpreting the findings. First, the experiments were conducted on a single data source (MSLD v2.0) with a relatively limited number of images; therefore, the model's generalizability across different populations, acquisition devices, and clinical settings remains uncertain. Second, the primary evaluation used a transductive graph learning setup, which may yield more optimistic performance

than real-world deployment scenarios where truly unseen samples must be handled. Third, the evaluation relied on a single 80:10:10 ratio split with a fixed seed; thus, the stability and robustness of the performance across different data partitions have not been comprehensively assessed.

In addition, methodological limitations remain at the model level. Although GAT is theoretically more adaptive due to its attention mechanism, it did not consistently outperform GCN in this study. Furthermore, the best GCN and GAT results still did not consistently exceed the performance reported by all prior studies across all metrics, indicating that there is still room for improvement. This study also did not incorporate explainable AI (XAI); thus, the model's predictions cannot yet be interpreted transparently in a clinical decision-support context. Future work should prioritize multi-source external validation, stricter inductive evaluation protocols, repeated experiments, and XAI integration to improve model reliability and practical applicability.

6. Conclusions

This study proposed and evaluated a graph-based multi-class mpox lesion identification framework that integrates class-balanced augmentation, CNN feature extraction, and graph neural classification on MSLD v2.0. Across the three dataset settings (MSLD v2.0-O, MSLD v2.0-G, and MSLD v2.0-GC), the results show that balancing the data through geometric and geometric plus HSV/color augmentation was associated with improved overall macro-level performance compared with the original imbalanced setting. Among all evaluated combinations, the best result was obtained by GCN with VGG16 features on MSLD v2.0-GC, achieving 92.00% accuracy and 91.55% macro F1-score, with a strong balance between macro precision and recall. The best GAT result was achieved by MobileNetV2 on MSLD v2.0-G with 89.25% macro F1-score. Overall, GCN showed a more consistent empirical trend than GAT in this study, although the difference was not statistically significant.

These findings indicate that combining balanced augmentation with relational learning over CNN-derived features is a promising strategy for mpox lesion classification, particularly for handling class imbalance and inter-class visual similarity. However, this conclusion should be interpreted within the current experimental scope. The study was conducted on a single relatively small dataset, primarily under a transductive graph-learning setup, and evaluated using a single split with a fixed seed; therefore, broader generalizability and performance stability are not yet fully established. In addition, the observed performance of GAT did not consistently demonstrate an advantage over GCN, and the best results do not universally surpass

all prior studies across all metrics. The absence of XAI analysis also limits clinical interpretability. Future work should focus on multi-source external validation, stricter inductive and patient-level evaluation protocols, repeated runs for robustness estimation, more extensive graph-construction and hyperparameter optimization, and XAI integration to improve transparency, reliability, and practical readiness for real clinical deployment.

Acknowledgments

This research was conducted as part of a collaborative research activity between Lambung Mangkurat University and Bac Lieu University.

Funding

This research was financially supported by the Lambung Mangkurat University Research Grant for 2025, with grant number 1842/UN8.2/PG/2025.

Conflict of interest

The authors declare that they have no conflict of interest.

Author contributions

Conceptualization: Mohammad Reza Faisal, Luu Duc Ngo, Radityo Adi Nugroho, Fatma Indriani

Data curation: Mohammad Reza Faisal, Luu Duc Ngo, Muhammad Rafi

Formal analysis: Mohammad Reza Faisal, Luu Duc Ngo, Radityo Adi Nugroho, Fatma Indriani

Funding acquisition: Mohammad Reza Faisal, Luu Duc Ngo

Investigation: Mohammad Reza Faisal, Luu Duc Ngo, Radityo Adi Nugroho

Methodology: Mohammad Reza Faisal, Luu Duc Ngo, Radityo Adi Nugroho, Fatma Indriani

Software: Muhammad Rafi

Supervision: Mohammad Reza Faisal, Luu Duc Ngo

Validation: Mohammad Reza Faisal, Luu Duc Ngo, Radityo Adi Nugroho

Visualization: Mohammad Reza Faisal, Radityo Adi Nugroho

Writing–original draft: Mohammad Reza Faisal, Luu Duc Ngo

Writing–review & editing: Mohammad Reza Faisal, Luu Duc Ngo

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

The dataset used in this study is publicly available and can be accessed at <https://www.kaggle.com/datasets/joydippaul/mpox-skin-lesion-dataset-version-20-msld-v20>

Further disclosure

During the preparation of this manuscript, the authors used generative artificial intelligence (AI) and AI-assisted technologies, including a large-language model, to improve the readability and language of the text and to assist in drafting portions of the related-work text that summarize existing, published literature. These tools were not used to generate scientific insights, design the study, generate, analyze, or interpret data, or draw scientific conclusions; all such intellectual work, including all reported methods, experiments, results, and conclusions, was performed by the authors. All AI-assisted text was produced under full human supervision, and the authors reviewed, verified, and edited it, including confirming that every cited references were accurate and correctly characterized. The authors take full responsibility for the integrity and accuracy of the published work. No AI or AI-assisted technology is listed as an author or co-author.

References

1. Ndembi N, Folayan MO, Komakech A, *et al.* Evolving Epidemiology of Mpox in Africa in 2024. *New Engl J Med.* 2025;392(7):666–676.
doi: 10.1056/NEJMoa2411368
2. Wangdi K, Adhikary R, Liu Z, *et al.* Monkeypox in South Asia: a systematic review. *Emerg Microbes Infect.* 2025;14(1).
doi: 10.1080/22221751.2025.2572677
3. Mills MG, Juergens KB, Gov JP, *et al.* Evaluation and clinical validation of monkeypox (mpox) virus real-time PCR assays. *J Clin Virol.* 2023;159:105373.
doi: 10.1016/j.jcv.2022.105373
4. Bunse T, Ziel A, Hagen P, *et al.* Analytical and clinical evaluation of a novel real-time PCR-based detection kit for Mpox virus. *Med Microbiol Immunol.* 2024;213(1).
doi: 10.1007/s00430-024-00800-4
5. Rahi M, Joy S, Sharma A. Public health challenges in the context of the global spread of Mpox infections. *Am J Trop Med Hyg.* 2023;108(4):641–645.
doi: 10.4269/ajtmh.22-0596
6. Asif S, Zhao M, Li Y, Tang F, Ur Rehman Khan S, Zhu Y. AI-based approaches for the diagnosis of Mpox: challenges and future prospects. *Arch Computat Methods Eng.* 2024;31(6):3585–3617.
doi: 10.1007/s11831-024-10091-w
7. Tsuneki M. Deep learning models in medical image analysis. *J Oral Biosci.* 2022;64(3):312–320.
doi: 10.1016/j.job.2022.03.003
8. Kshatri SS, Singh D. Convolutional neural network in medical image analysis: a review. *Arch Computat Methods Eng.* 2023;30(4):2793–2810.
doi: 10.1007/s11831-023-09898-w
9. Chlap P, Min H, Vandenberg N, Dowling J, Holloway L, Haworth A. A review of medical image data augmentation techniques for deep learning applications. *J Med Imaging Radiat Oncol.* 2021;65(5):545–563.
doi: 10.1111/1754-9485.13261
10. Kurdi SZ. Machine learning–based classification framework for human health care monitoring. *Int J Theor Appl Comput Intell.* 2026;2026:1–15.
doi: 10.65278/IJTACI.2026.1
11. Vuran S, Ucan M, Akin M, Kaya M. Multi-Classification of Skin Lesion Images Including Mpox Disease Using Transformer-Based Deep Learning Architectures. *Diagnostics.* 2025;15(3):374.
doi: 10.3390/diagnostics15030374
12. Gabriel CIO, Kumar R. An explainable deep learning and machine learning model for Mpox classification using dermatologic images. *Digit Transform Soc.* 2026;5(3):365–385.
doi: 10.1108/DTS-08-2025-0267
13. Pal M, Branda F, Alkhedaide AQ, *et al.* Early detection of human Mpox: A comparative study by using machine learning and deep learning models with ensemble approach. *Digit Health.* 2025;11.
doi: 10.1177/20552076251344135
14. Ali SN, Ahmed MT, Jahan T, *et al.* A web-based mpox skin lesion detection system using state-of-the-art deep learning models considering racial diversity. *Biomed Signal Process Control.* 2024;98:106742.
doi: 10.1016/j.bspc.2024.106742
15. Huan E, Dun H. MSMP-net: a multi-scale neural network for end-to-end monkeypox virus skin lesion classification. *Appl Sci.* 2024;14(20):9390.
doi: 10.3390/app14209390
16. Elhadidy MS, Elgohr AT, Mousa A, Safwat A, Abdelfatah RI, Kasem HM. Benchmarking pre-trained CNNs and vision transformers for Mpox-related dermatological image classification on MSLD v2. 0. *Results Eng.* 2025;28:108071.
doi: 10.1016/j.rineng.2025.108071
17. Duan S, Huang P, Chen M, *et al.* Semi-supervised classification of fundus images combined with CNN and GCN. *J Appl Clin Med Phys.* 2022;23(12).

- doi: 10.1002/acm2.13746
18. Shi C, Liao D, Wang L. Hybrid CNN-GCN network for hyperspectral image classification. *IEEE J Sel Top Appl Earth Observations Remote Sensing*. 2025;18:10530-10546.
doi: 10.1109/JSTARS.2025.3548571
19. Aboshosha BW, Rehman SU, Mahmoud LN, Sadek I. AI-driven diagnosis of mpox using deep learning models. *PLoS ONE*. 2026;21(7):e0352161.
doi: 10.1371/journal.pone.0352161
20. Zhao X, Ma J, Wang L, Zhang Z, Ding Y, Xiao X. A review of hyperspectral image classification based on graph neural networks. *Artif Intell Rev*. 2025;58(6).
doi: 10.1007/s10462-025-11169-y
21. Altalhan M, Algarni A, Turki-Hadj Alouane M. Imbalanced data problem in machine learning: A review. *IEEE Access*. 2025;13:13686-13699.
doi: 10.1109/access.2025.3531662
22. van Zyl JP, Engelbrecht AP. Analysis of classification metric behaviour under class imbalance. *Egypt Inform J*. 2025;31:100711.
doi: 10.1016/j.eij.2025.100711
23. Goceri E. Medical image data augmentation: techniques, comparisons and interpretations. *Artif Intell Rev*. 2023;56(11):12561-12605.
doi: 10.1007/s10462-023-10453-z
24. Kumar T, Brennan R, Mileo A, Bendeache M. Image data augmentation approaches: A comprehensive survey and future directions. *IEEE Access*. 2024;12:187536-187571.
doi: 10.1109/ACCESS.2024.3470122
25. Sengupta P, Mehta A, Rana PS. Enhancing performance of deep learning models with a novel data augmentation approach. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. New Delhi, India: IEEE; 2023:1-7.
doi: 10.1109/icccnt56998.2023.10308298
26. Velastegui R, Yang L, Han D. The importance of color spaces for image classification using artificial neural networks: a review. In: *Lecture Notes in Computer Science*. Cham: Springer International Publishing; 2021:70-83.
doi: 10.1007/978-3-030-86960-1_6
27. Shuai Z, Xiaolin D, Jing Y, et al. Comparison of different feature extraction methods for applicable automated ICD coding. *BMC Med Inform Decis Mak*. 2022;22(1).
doi: 10.1186/s12911-022-01753-5
28. Liu H, Li I, Liang Y, Sun D, Yang Y, Yang H. Research on deep learning model of feature extraction based on convolutional neural network. In: *2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA)*. Shenyang, China: IEEE; 2024:810-816.
doi: 10.1109/ICIPCA61593.2024.10709168
29. Shabana M, Kumar JP, Masthan K. Image Forgery Detection Using Sobel Edge Detection, Feature Extraction. In: *Disruptive Technologies in Computing and Communication Systems*. Boca Raton, FL, USA: CRC Press; 2024:253-258.
doi: 10.1201/9781032665535-42
30. Archana R, Jeevaraj PSE. Deep learning models for digital image processing: a review. *Artif Intell Rev*. 2024;57(1).
doi: 10.1007/s10462-023-10631-z
31. Zhao X, Wang L, Zhang Y, Han X, Deveci M, Parmar M. A review of convolutional neural networks in computer vision. *Artif Intell Rev*. 2024;57(4).
doi: 10.1007/s10462-024-10721-6
32. Mahmood SH, Farhan AK. An evaluation of convolutional neural network architectures: AlexNet, LetNet, VGG-16, and fingerprint-net for fingerprint classification. In: *AIP Conference Proceedings*. Vol 3224. AIP Publishing; 2025:030021.
doi: 10.1063/5.0255956
33. Zhang H, Yang S, Zhang X. Automatic recognition and classification of microalgae using an inception-v3 convolution neural network model. *Int J Environ Sci Technol*. 2023;21(4):4625-4634.
doi: 10.1007/s13762-023-05209-9
34. Lachaud G, Conde-Cespedes P, Trocan M. Mathematical Expressiveness of Graph Neural Networks. *Mathematics*. 2022;10(24):4770.
doi: 10.3390/math10244770
35. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst*. 2021;32(1):4-24.
doi: 10.1109/TNNLS.2020.2978386
36. Parisot S, Ktena SI, Ferrante E, et al. Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease. *Med Image Anal*. 2018;48:117-130.
doi: 10.1016/j.media.2018.06.001
37. Wang X, Liang Z. Hybrid network model based on 3D convolutional neural network and scalable graph convolutional network for hyperspectral image classification. *IET Image Process*. 2022;17(1):256-273.
doi: 10.1049/ipr2.12632
38. Ciano G, Rossi A, Bianchini M, Scarselli F. On inductive-transductive learning with graph neural networks. *IEEE Trans Pattern Anal Mach Intell*. 2022;44(2):758-769.
doi: 10.1109/TPAMI.2021.3054304

39. Vrahatis AG, Lazaros K, Kotsiantis S. Graph Attention Networks: A Comprehensive Review of Methods and Applications. *Future Internet*. 2024;16(9):318.
doi: 10.3390/fi16090318
40. Sun C, Li C, Lin X, *et al*. Attention-based graph neural networks: a survey. *Artif Intell Rev*. 2023;56(S2):2263–2310.
doi: 10.1007/s10462-023-10577-2
41. Diallo R, Edalo C, Awe OO. Machine learning evaluation of imbalanced health data: a comparative analysis of balanced accuracy, MCC, and F1 score. In: *Practical Statistical Learning and Data Science Methods (STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics&Health)*. Cham, Switzerland: Springer Nature; 2024:283–312.
doi: 10.1007/978-3-031-72215-8_12
42. Zhou X, Zeng Y, Hao Z, Wang H. A robust graph attention network with dynamic adjusted graph. *Eng Appl Artif Intell*. 2024;129:107619.
doi: 10.1016/j.engappai.2023.107619
43. Bello A, Ng SC, Leung MF. Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121. *Appl Sci*. 2024;14(17):7707.
doi: 10.3390/app14177707
44. Mohammed FA, Tune KK, Assefa BG, Jett M, Muhie S. Medical Image Classifications Using Convolutional Neural Networks: A Survey of Current Methods and Statistical Modeling of the Literature. *Mach Learn Knowl Extr*. 2024;6(1):699–736.
doi: 10.3390/make6010033
45. A RS, Chamola V, Hussain Z, Albalwy F, Hussain A. A novel end-to-end deep convolutional neural network based skin lesion classification framework. *Expert Syst Appl*. 2024;246:123056.
doi: 10.1016/j.eswa.2023.123056