Al-Nahrain Journal of Science
ANJS, Vol.28 (4), December, 2025, pp. 207-225

Evaluating Felzenszwalb and Quick Shift Integrated Graph Neural Networks for Monkeypox Blister Segmentation

Ashutosh Satapathy *, Duda Neelima, Naresh Dukka

[1]Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada - 520007, India

Article's Information

Received: 28.02.2025
Accepted: 08.10.2025
Published: 15.12.2025

Keywords:
Monkeypox disease
Skin lesion segmentation
Spectral image clustering
GNN
RAG

Abstract

The Deep Neural Networks (DNNs) are increasingly leveraged in medical image segmentation, particularly for complex skin conditions like Monkey pox. Due to the distinctive visual features of Monkey pox, such as rashes, blisters, and scabs, distinguishing these features from those of Smallpox and Chickenpox may be difficult, thereby complicating diagnosis. We propose a comparison study with Graph Neural Networks (GNNs) integrated with both Felzenszwalb's and Quick Shift segment algorithms for Monkey pox blister segmentation. Our GNN and Felzenszwalb's algorithm combination reached a Mean Structural Similarity Index Metric (SSIM) of 0.85 ± 0.091 and an accuracy of 85.03%, outperforming the other traditional methods. Most distinctly, the combination of Felzenszwalb and GNN were more accurate in segmentation than Quick Shift, which achieved a lower performance SSIM of 0.63 ± 0.080. Correspondingly, GNN using Felzenszwalb's algorithm reached a lower average test loss of 0.0082 compared to the other approaches. This further establishes that GNNs combined with traditional segmentation methods may greatly enhance accuracy and speed of Monkey pox blister segmentation for improved diagnostic applications. It supports automated lesion analysis for early Monkey pox screening in digital dermatoscopy and teledermatology systems.

http://doi.org/10.22401/ANJS.28.4.16
*Corresponding author: ashutosh@vrsiddhartha.ac.in

1. Introduction

Monkey pox virus is an uncommon infectious disease caused by a Zoonotic Orthopoxvirus. Regarding traits, it has close aspects to cowpox and Smallpox in the Poxviridae family within the Orthopoxvirus genus [1]. It is acquired primarily through monkeys and rodents. However, the transmission from human to human is also very wary. In 1958, at a laboratory in Denmark, scientists noticed this Virus inside a monkey's body. Then In 1970, the first human Monkey pox case was recorded in the Democratic Republic of Congo, while Smallpox prevention campaigns were ramped up [2]. This disease is commonly witnessed in the central and western regions of Africa, and most of the cases are among people close to the tropical rainforest.

The virus is transmitted when a person comes into close contact with another infected person, animal, or material. It transmits through direct body contact, animal bites, respiratory droplets, eye, nose, or mouth mucosa and sometimes even consumption .Some early-stage symptoms of patients infected with Monkey pox include fever, body aches, and fatigue; the long-term effect is a red bump on the skin. As per Saxena et al., Monkey pox symptoms usually last 2 to 4 weeks, and severe cases can occur. According to a WHO report, the recent case-death ratio has been between 3% and 6% [3]. The treatment was initially observed for managing the Smallpox virus across two periods but has since been adapted for treating the Monkey pox virus [4]. During the invasion period, the patients had back pain, fever, swelling of lymph nodes, intense headaches, muscle aches, and a lack of energy. In the second period, within 1–3 days of fever, the skin lesions begin. In 95% of cases, the face is affected. In 75% of them, the palms of the hands and soles of the feet are affected. Additionally, human oral mucosa, genitalia, and conjunctiva are affected in varying percentages [5].

The infection rate has increased significantly, from 50 cases in 1990 to over 5000 in 2020 [6], and by June 2024, 97,281 cases were reported globally across 118 nations [7, 8]. While some treatments, such as Brincidofovir and Tecovirimat, have been authorized for Monkey pox, there is no broad public diagnostics test available, thus relying mainly on clinical observation of cutaneous lesions and patient history of exposure, as indicated in Figure 1 [4].
Therefore, a greater demand for automated image-based diagnostic tools to support quick preliminary detection. Image segmentation methods assist in separating lesions from the surrounding skin by partitioning images into significant segments, thus simplifying the analysis of the affected areas [9]. In the present work, a new framework combines Graph Neural Networks (GNNs) with Felzenszwalb's and Quick Shift algorithms for effective Monkey pox blister segmentation to enhance accuracy and aid in timely diagnosis.

(a) (b)

Figure 1. (a) Monkeypox-induced patient; (b) non-Monkeypox (Measles) induced patient.

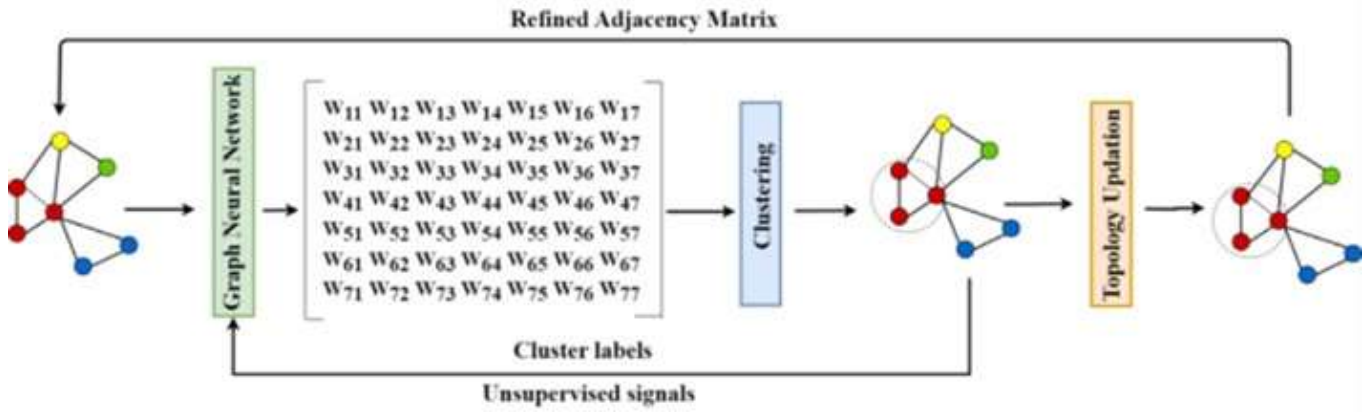


Figure 2. A Graph Neural Network architecture

GNNs are a subtype of neural network that operates on graph-structured data of interrelated nodes and edges. GNNs are useful for graph structure tasks, such as nodes' classification, links' prediction, graphs' clustering, and segmentation. GNNs are developed as enhanced forms of recursive neural networks, random walk models, and their characteristics [10]. It applies neural networks to learn hidden patterns from data organized in a graph format. GNNs are important because they help extract and capture the essential information from a graph. Most graph classifications utilize GNNs using these graph-structured representations, making them much more effective. Whereas Convolutional. Neural Networks (CNNs) cannot conduct graph-based tasks well and are limited, GNNs have broadened the scope of neural network applications by adding node, edge, and graph-level tasks that involve predictions [11]. Furthermore, Zhu et al. implemented a cluster-aware method that enables unsupervised graph representation to improve clustering accuracy by modifying the graph structure [12]. The architecture shown in Figure 2 is a GNN system specifically designed for clustering purposes.

2. Related work

Research by Castro et al. used a hard C-means technique with landmark-based registration to define the amount of disease caused by the Monkey pox virus before injection as well as after its progression [13]. The accuracy of the automated estimation relative to the manual segmentation proves the effectiveness of the method employed. They were the first to provide partial volumes of meningoencephalitis associated with the Monkey pox virus. This was done by establishing the level set Volume of Interest (VoI) of the normal lung

image to assist in differentiating the lungs from the other parts of the lungs. A hard c-means clustering algorithm, which was highly validated on different classes of images, was then used to perform image segmentation within that area. This clustering algorithm proceeded to classify pixels iteration until each pixel had been classified and a lower cost function was reached. The data set for this literary survey includes CT data containing CereTorn. The partial volume accuracy for seven categories stood at 69.3%. McNeil et al. developed a deep-learning method for monkey pox lesion localization and counting in-patient photos. In 20 photos from 12 patients with monkey pox, they manually annotated all visible lesions and trained a U-Net network with an Inceptionv4 encoder to localize and 13 count the lesions [14]. In a leave-one-out evaluation, they got promising results for lesion segmentation, with a median Dice of 0.74. In 2022, Muñoz-Saavedra et al. of the University of Seville designed, implemented and evaluated several diagnostic aid systems for Monkey pox. Images related to Monkey pox were obtained using official governmental websites and from a public dataset called the Monkey pox Skin Dataset for this classification [15]. The classifiers developed and evaluated are based on Convolutional Neural Network models, and some ensembles comprises of a combination of those models, obtaining automatic classification results between healthy, Monkey pox, and other skin damages given a close skin tissue image. The use of the AI Grad CAM technique for an ensemble network improved the results whereby the system accuracy was above 93.33% when using a single CNN model such as VGG-19 and ResNet50 and above 98.33% when using a CNN ensemble made up of ResNet50, EfficientNet-B0 and MobileNet-V2, even though these classifier systems were applied on a reduced number of samples and in prospective uses of classification there exist chances of getting the wrong classifications.

Ahsan et al. analyzed six deep-learning convolutional models, VGG16, InceptionResNetV2, ResNet50, ResNet101, MobileNetV2, and VGG19, for assisted detection of Monkey pox disease in images using transfer learning [16]. As observed in the research, the design steps entailed building deep learning models based on hybrid data and developing a mobile application with faster loading speeds, which would help diagnose the Monkey pox disease. They also justified and interpreted model performance through Local Interpretative Model-agnostic Explanations (LIME), which is an external interpretive agnostic approach, but took an already trained model and fed it with data through layers that were capped off by a dense layer fine-tuned to Monkey pox. No changes were made to the convolution's weight during the training stage. Regarding the diagnosis of Monkey pox disease infection, the InceptionResNetV2 and MobileNetV2 attained high and respectable accuracy rates of 93% and 99%, respectively. The best part of their work was that different pre-trained deep neural networks and their ensemble were trained, and various performance measures were compared. ResNet50 still performed best, with accuracy and precision of 82.96% and 87%, respectively. Due to the data shortage, the developed model's accuracy was low.

Chakroborty discussed a transfer learning method to predict Monkey pox disease using the Monkey pox Skin Lesion Dataset (MSLD) in 2022. They employed VGG16, ResNet50, Inception v3, and an ensemble model of the neural networks [17]. The optimized weights of the network were derived using many runs incorporating Adam optimization, and the evaluation was done concerning the cross-entropy loss. One of the advantages common to all of the studies was the ability to perform such assessments on different deep learning neural networks where the weights were fine-tuned, and an ensemble of these models performed well. The ResNet50 model achieved the best overall accuracy with 82.96% and a precision of 87% among all models developed. However, there was evidence that the limitations on the volume of data affected the efficiency of the model, which could have a big impact on determining the generalizability of the output.

Alhasson et al. suggested models to identify the disease known as monkey pox. MSID was created in 2022 using web scraping to contain Monkey pox illnesses and other diseases. Some pretrained models were proposed including ResNet50, DenseNet121, Inception-V3, SqueezeNet, MnasNet-A1, MobileNet-V2 and ShuffleNet-V2. Performance parameters for these models, including precision, accuracy, etc., were examined [18]. Due to the trade-off between the training sample size and the number of trainable parameters in a model, deep learning models frequently over fit or under fit. ShuffleNet-V2 outperformed the other models on the referred dataset, with an average accuracy of 79%.

Yadav et al. discussed and evaluated the DenseNet-121 model, a deep learning technique to diagnose monkeypox using ML techniques through a dataset of Monkeypox skin images [19]. The primary objective of using DenseNet-121 was to help improve feature reusability and reduce the vanishing

gradient problem or lower parameter usage, which came in handy while training deep learning models. Nonetheless, they did not conduct more research trials on other deep learning models that efficiently identify monkey pox from large datasets with many other skin diseases such as chickenpox, syphilis, eczema, etc. In the end, they improved the testing accuracy to 93.7% after fine-tuning the hyper-parameters of DenseNet-121.

Sitaula et al.compared thirteen different pre-trained DL models such as VGG16, VGG19, ResNet50, ResNet101, InceptionV3, Xception, EfficientNet-B0, EfficientNet-B1, EfficientNet-B2, DenseNet-121, IncepResnetv2, MobileNetV2, and DenseNet-169 with the help of transfer learning on the Monkeypox image dataset [20]. They chose Xception and DenseNet-169 based on the empirical study, fine-tuned the parameters for improving the accuracy, and showed the explainability using Grad-CAM and LIME of the best-performing deep learning model. Since the approach was based on pre-trained DL models, it could be problematic if the model was deployed in a memory-constrained setting. Finally, an assessment of the model was performed with the help of different performance metrics, such as accuracy and precision, and the model recorded an accuracy of 87.13% and a precision of 85.44%.

Yang et al. developed AICOM-MP, a monkey pox detector based on AI systems that perform exceptionally well – particularly with images acquired from low-end devices. While comparing AICOM-MP with other popular AI-based monkey pox detectors, AICOM-MP achieved a decisively better state-of-the-art (SOTA) performance [21]. They used the U2-Net model and its pre-trained weights for object and background removal, the FCNResNet10 model for region-based skin detection and segmentation, and EfficentNet in the classification. AICOM-MP had evolved as a web service so that this high throughput monkey pox screening system could be accessible to everyone without restrictions. AICOM-MP was thoroughly validated against MSLD datasets and showed a precision of 0.9102.

Sahin et al. suggested that new-generation smartphone devices can diagnose the diseases caused by Monkey pox through an Android mobile application. It was found that MobileNetv2 was suitable for the application implementation [4]. At the same time, the model was tested against other pre-trained deep learning models, which include ResNet18, GoogleNet, EfficientNet, NasNet-Mobile, and ShuffleNet, using the MSLD dataset. Several performance measures, including accuracy and precision, were also carried out to validate the proposed approach. The proposed model achieved 91.11% accuracy in diagnosing Monkey pox, which is commendable and underlines the model's efficacy. Even with such results, the issues related to the network system's stability were part of the aspects that had to be improved.

Bamini et al. created a deep learning pipeline using a hybrid CoAtNet architecture, which utilized convolutional and transformer layers, for Monkey pox lesion segmentation and classification. After advanced image augmentation and attention-aware segmentation, their model achieved an accuracy of 95.42% on the MSLD v2.0 dataset [22]. Srinivasan et al. proposed an Enhanced Spatial-Awareness Capsule Network (ESACN) that was tested on a dataset of 659 images from four different classes: Normal, Measles, Chickenpox, and Monkeypox. The ESACN's accuracy for detecting Monkeypox was 92.68%, with an F1 score of 96.73% [23].

Pratama et al. created a deep learning model based on a hybrid of Xception and InceptionV3. Features were fused, and augmentation was through Albumentation. Overall, the accuracy of the model achieved 85.96% on the MSLD dataset. An ablation study was conducted, and it was confirmed that the augmentation led to improved model accuracy and the model's sensitivity towards variation in lesions [24]. Vandana et al. presented MRpoxNet, which is an extension of ResNet50 that increases depth from 177 to 182 layers by adding 10 additional convolutional, ReLU, dropout, and batch normalization layers. MRpoxNet outperformed ResNet50, VGG16, AlexNet, and GoogleNet on the same dataset, achieving a diagnostic accuracy of 98.1% after being trained on the MSID expanded via data augmentation to 6116 images. Their findings demonstrate the model's resilience and applicability for incorporation into clinical diagnostic procedures [25].

Yue et al. utilized ShuffleNetV2-X20 on the MSLD dataset and achieved an accuracy of 72.33%, indicating that lightweight CNNs could be a viable method for analyzing medical images [26]. Several other architectures performed likewise, such as EfficientNet-B1, which achieved an accuracy of 71.02%, and MobileOne-S0, which achieved an accuracy of 73.58%. The transformer-based models, MobileViT-XS with an accuracy of 74.92%, FastVIT-t8 with an accuracy of 75.00%, and Swin transformer-T with an accuracy of 77.63%, provide evidence of a rising efficacy of using vision transformers for lesion classification. Custom models like PoxNet22 and MpoxNet achieved

74.50% and 74.94% accuracy, respectively, while MonketNet demonstrated a strong accuracy score of 79.44%. The authors' MpoxMamba light-weight hybrid architecture had the highest accuracy at 82.47%. These results show a clear move towards small, high-performance architectures that can be used in real time in healthcare settings with limited resources.

3. Proposed Methodology

3.1. Proposed System Architecture

The system architecture for Monkey pox image segmentation optimization in Figure 3 outlines a sound approach to improve the quality of segmenting Monkey pox images. The first step in the process is the input image. The input image is converted to a monochrome image to help with the following tasks. The next step includes segmenting the image using the labeled areas obtained through Felzenszwalb's or the Quick-shift segmentation algorithm, which is segmented based on image intensities, thus yielding an over-segmented image. After this, a GNN employs regions of over-segmented images as nodes in a Region Adjacency Graph (RAG) to efficiently study interactions between adjacent regions [27]. This methodology provides important practical consequences. The Minimum Cut Algorithm and the Min Cut Pool determine the best way to partition a graph, but reduce the edges connecting the different sets without obtaining an optimal graph partition [28]. Compared to the first over-segmentation output of the image segmentation process, this leads to a more accurate and significant segmented image. The proposed methodology gives a structured approach incorporating sophisticated methods to improve image segmentation of Monkey pox lesions, which is relevant in accurately diagnosing and evaluating patients in health facilities.

3.2. Image Segmentation

1. Felzenszwalb's segmentation algorithm

Felzenszwalb's segmentation algorithm is one of the popular techniques used in image segmentation; it is named after its author, Pedro Felzenszwalb.

It is good at dividing an image into coherent regions or segments with similar color and texture while keeping meticulous with the object boundaries. It begins its process by initializing each image pixel as a separate segment. Then, inspecting a sorted list of edges, the algorithm checks the fulfillment of a similarity criterion for each edge.

Each criterion considers the properties of color similarity and spatial proximity between the two segments to which an edge is attached; when such a similarity criterion is met, the algorithm merges the two connected segments. This integration is a process that repeatedly merges segments progressively based on their similarity until no further merges are possible or until a predetermined stopping criterion is achieved [29].

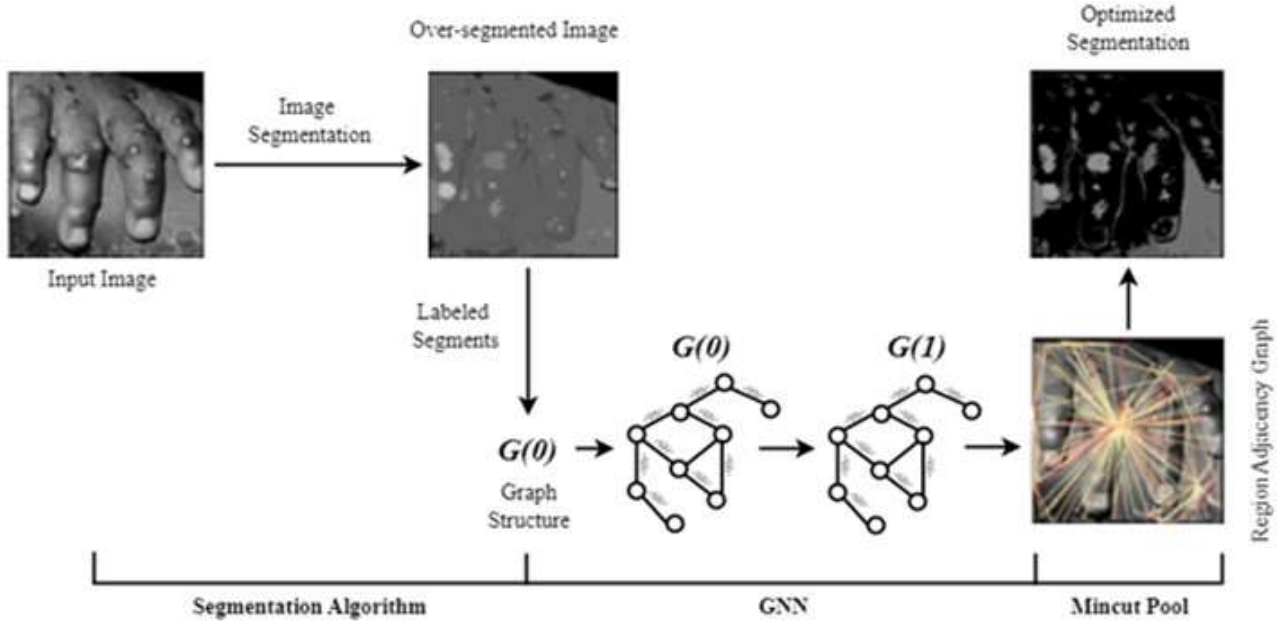


Figure 3. Proposed system architecture of Monkeypox blisters segmentation

The final result is a set of segments representing meaningful partitions of the original image and capturing local and global patterns. Felzenszwalb's segmentation algorithm is fairly effective and is appreciated for its visually good segmentation. The algorithm results in meaningful partitions of the original image through color and spatial characteristics and takes the form of a set of segments. Felzenszwalb's algorithm, Algorithm 1, segments an image into spatial properties, where a gray-scale image is divided into partitions.

Algorithm 1 Felzenszwalb's Segmentation Algorithm

Input: A grayscale image, denoted as I.
Output: Segments

1. *Vertices:* Each pixel p_i in the image I is considered as a vertex.
2. *Edges:* Connect neighboring pixels with weights based on a measure of similarity (color proximity and intensity difference).
3. *Edge Weight Calculation:* The weight $w(p_i, p_j)$ between two neighboring pixels p_i and p_j can be calculated using a similarity measure.
$$w(p_i, p_j) = exp\left(-\frac{|I(p_i) - I(p_j)|^2}{\sigma^2}\right). exp\left(-\frac{d(p_i,\ p_j)^2}{\rho^2}\right)$$
/* $I(p_i)$ and $I(p_j)$ are intensities at pixels p_i and p_j. d(p_i, p_j) is the distance between p_i and p_j. σ and ρ controls sensitivity to intensity differences and spatial proximity. */
4. *Segment Initialization:* Start with each pixel as its segment.
5. *Merging Criterion:* Merge segments S_i and S_j if the edge weight $w(p_i, p_j)$ is below the adaptive threshold τ.
$$\text{if } w(p_i, p_j) < \tau, \text{ merge } S_i \text{ and } S_j$$
$$\tau = min\left(Int(S_i), Int(S_j)\right) + \frac{k}{min(|S_i|, |S_j|)}$$
/* $Int(S_i)$ and $Int(S_j)$ are the maximum edge weight within clusters S_i and S_j. $|S_i|$ and $|S_j|$ are the number of pixels in segments S_i and S_j. */
6. *Stopping Iteration:* Stop merging when no edges satisfy the merging condition (i.e., all remaining edges have weights aboveτ).

Felzenszwalb's segmentation algorithm proceeds by building a graph G = (V, E) whose vertices V are the image pixels p_i. Each vertex p_i will be linked by edges E with weights given by a similarity measure $w(p_i, p_j)$between adjacent vertices p_j. This generally includes parameters σ and ρ, representing color closeness and spatial proximity. σ is usually between 0.5 and 1.0, and usual values of ρ are between 0.5 and 2.0. every pixel initially corresponds to a segment. The algorithm repeatedly merges pairs of adjacent segments S_i and S_j for which the edge weight $w(p_i, p_j)$ is smaller than the adaptive threshold τ . This merging process continues until all the remaining edges have weights greater than τ. The segments merge based on significant similarity cues while preserving boundaries where differences between pixels are strong. It depends on the application and the desired detail segmentation level to achieve. τ can be between 0.5 and 2.0 or more, depending upon the characteristics of images as well as segmentation goals.

2. Quick Shift Algorithm

Algorithm 2 attempts to achieve image segments with automatic adjustment of the quick shift based on invariant features for determining objects in detection and recognition [30]. First, we eliminate image acquisition factors like shadow and highlight, which is possible by applying an invariant method. The concept of "similarity" among pixels is one of the major components of the quick-shift algorithm and is often described with a Gaussian kernel in feature space. The similarity measure takes into account both spatial proximity and feature similarities. Thus, pixel groups can be formed based on the similarity of their characteristics.

Algorithm 2 Quick Shift Segmentation Algorithm

Input: A gray-scale Image, denoted as *I.*
Output: Segments

1. Convert 1-channel monochrome image to 3-channel gray-scale image.
2. *Parameters - kernel_size:* Set the kernel size for density estimation.
3. *max_dist:* Set the maximum distance for merging segments. *ratio:* Set the ratio for balancing color and spatial distances.
4. Transform each pixel's color and spatial coordinates into a feature space.
5. For each pixel *p*, calculate its density as the sum of kernel values for neighboring pixels *q*:
$$Density(p) = \sum Kernel(p, q),\ for\ all\ q$$
6. Identify high-density points in the feature space as potential segment centers.
7. For each pixel p, assign it to the nearest high-density point based on feature space distance:
$$Distance\ (p, q) = \sum(wcolor * Color\_Distance(p, q) + wspatial * Spatial\_Distance(p, q))$$
8. Check if two segments should merge based on distance and density:

$$Merge\ (p,q) = ((Distance(p,q) < max\_dist)\ AND\ (\frac{Density(p)}{Density(q)} < ratio))$$

Quick-shift algorithm iteration updates cluster assignments of pixels based on similarity with neighboring pixels. The process continues until convergence is achieved, and then each pixel belongs to a different cluster, which falls under a distinct image segmentation region. After the pre-processed images, the quick-shift algorithm is called with relevant parameters based on the nature of the invariant images. Quick shift is a non-parametric clustering algorithm for similar pixels of feature space. This allows for effectively segmenting the image into regions of coherent visual content. The outcome of segmentation with the Quick Segmentation Algorithm provides excellent information regarding the accuracy and reliability the method poses in delineating Monkey pox blisters from surrounding healthy tissue. Higher values of SSIM, as well as lower computational complexity, indicate higher quality and efficiency for the segmentation, which is superior to diagnostic and research performance in medical image analysis.

3.3. Spectral Image Clustering

1. Graph Construction

Spectral image clustering involves the graph-based approach with which the gray-scale image is clustered into spectral regions [31]. Initially, the image segmentation results in N segments, represented as nodes in the graph. The feature vectors that describe these segments define an initial feature matrix X. Construct an adjacency matrix A such that the non-zero entries of the adjacency matrix encode relationships between segments. The clustering process commences by normalization of the adjacency matrix A to obtain the normalized adjacency matrix $\hat{A}$. The degree matrix D is also added to reflect the graph's structure. Using iterative spectral processing, which can be denoted as repeated layers of graph convolutions on the feature matrix $H^{(l)}$ refines the features of the segments based on the relationships in their neighborhoods. This iterative refinement enhances the segments' representation for subsequent clustering. After the spectral process, a pool operation aggregates node features to a reduced set X_{pooled} and simplifies the adjacent matrix to A_{pooled}. This can reduce the complexity of computations while retaining structural information relevant to clustering. The final process now executes a spectral clustering procedure on the pooled adjacency matrix A_{pooled}, using the normalized Laplacian matrix L_{norm} to calculate the eigenvectors corresponding to the smallest eigenvalues. These eigenvectors are then fed to a k-means clustering algorithm, which determines to which K cluster each segment should be assigned based on its spectral characteristic. This can capture local and global relationships between segments in an image; thus, this is one robust framework for unsupervised clustering in image segmentation.

Algorithm 3 Spectral Image Clustering using GNN

Input: A Grayscale segmented image I by Felzenszwalb's or Quick shift algorithms.

Output: Cluster assignments for original segments.

1. A Gray-scale Image I consists of N segments. Segments represented as nodes $V = \{v_1, v_2, \ldots, v_N\}$ in a graph $G = (V, E)$.
2. Feature matrix $X \in R^{NxF}$ from Algorithm 3 or 4. $X_i \in R^F$ is an F-dimensional feature vector for the segment v_i.
3. Compute adjacency matrix $A \in R^{NxN}$.
$$A_{ij} = \begin{cases} 1 & Segment\ i\ and\ j\ are\ adjacent \\ 0 & Otherwise \end{cases}$$
4. Compute degree matrix $D \in R^{NxF}$.
$$D_{ii} = \sum_j A_{ij}$$
5. Set $H^{(0)} = X$
6. Compute normalized adjacency matrix $\hat{A}$
$$\hat{A} = D^{-\frac{1}{2}} A D^{\frac{1}{2}} + I$$
7. **for** each layer l **do**
$$H^{(l+1)} = \sigma(\hat{A} H^{(l)} W^{(l)})$$
/* Feature matrix $H^{(l)} \in R^{NxF^{(l)}}$, $W^{(l)} \in R^{F^{(l)} x F^{(l-1)}}$ is a learnable weight matrix at layer l. σ is an activation function. */
8. **end for**
9. Update $X_{pooled} = S^T H^{(L)}$ and $A_{pooled} = S^T A S$
/* Cluster assignment matrix $S \in R^{NxK}$, updated pooled node feature matrix $X_{pooled} \in R^{KxF^{(L)}}$ and pooled adjacency matrix $A_{pooled} \in R^{KxK}$ */
10. Update loss function L value.
$$L = \alpha L_{recon} + \beta L_{cluster}$$
/* α and β are weighting factors */
11. Repeat steps 7-10 to optimize $W^{(l)}$ and S by minimizing loss function L.
12. Compute normalized Laplacian matrix L_{norm}.
$$L_{norm} = I - D_{pooled}{}^{-\frac{1}{2}} A_{pooled} D_{pooled}{}^{\frac{1}{2}}$$
/* D_{pooled} is the degree matrix of A_{pooled}*/
13. Compute eigen vectors $U \in R^{Kxk}$ of the k smallest eigenvalues of L_{norm}.
14. Apply k-means clustering on U_i to obtain cluster assignments for each node v_i.

Algorithm 3 describes the method for spectral image clustering via GNN; graph pooling starts with converting a grey-scale segmented image into a graph where nodes are segments and edges describe their relationship. Node features go initially through a GNN to acquire suitable embeddings. A pooling step reduces the number of nodes to simplify the graph, resulting in a condensed graph that keeps key structural information. The loss function used in the algorithm combines reconstruction loss and cluster assignment loss to define the loss in that the pooled graph reasonably captures the original structure and groups of similar nodes.

$$L_{recon} = \left\| A - SA_{pooled}S^T \right\|_F^2 \quad (1)$$

$$L_{cluster} = \sum_{i,j} A_{ij} \left\| S_i - S_j \right\|^2 \quad (2)$$

Reconstruction loss L_{recon} and cluster assignment loss $L_{cluster}$ in equations 1 and 2 are computed using original adjacency matrix A, pooled adjacency matrix A_{pooled}, and cluster assignment matrix S. Spectral clustering is done on the pooled graph by computing the normalized Laplacian and taking the eigenvectors of the smallest eigenvalues. The eigenvectors give a low-dimensional representation of the nodes, further clustered using k-means. Optimization of the loss function drives the whole process, further improving the clustering accuracy of initial segments.

2. Region Adjacency Graph

RAG is a segmentation stage in images before techniques such as Mincut pooling. It also describes the adjacent relationships between neighboring segments within an image. RAG is created based on some techniques, including Felzenszwalb's technique, which is applied to the pre-processed skin lesion images. It captures adjacent relationships among segments by showing the adjacent pieces to each other in that image. In the case of the Monkey pox Segmentation, the structured representations of segments developed by RAG are an output produced by Felzenszwalb's algorithm [32]. This structured representation is then used as an input for techniques such as MinCutPool, which helps fine-tune the segmentation further to detect lesions caused by Monkey pox with greater precision and accuracy.

3. Mincut Pooling

Mincut pooling is an important technique in graph-based image segmentation refinement for various segmentation tasks [33, 34]. Using the graph representation built from the segments produced through algorithms such as Felzenszwalb's method, Mincut pooling enables the generation of meaningful segments that often correctly outline individual regions in the given images of the skin lesion. A GNN is incorporated into the model to predict the optimal groupings of those segments about the encoded features and spatial relationships in the graph. The min-cut optimization process ensures that the realized segmentation attains minimum-cost partitioning of the graph, resulting in semantically meaningful cohesive segments [28]. Ultimately, Mincut pooling has been very important in refining the accuracy of identifying lesions caused by the Monkey pox virus and thus elevates the overall performance of the segmentation algorithm in realistic clinical environments.

3.4. Performance Metrics

1. Structural Similarity Index

SSIM is a quality metric for predicting the perceived quality of digital images. It does this by calculating how alike two images are, a process that can be applied to various images and contexts. The SSIM index, as a full reference metric, depends on an uncompressed or unadulterated reference image. The SSIM index can be calculated for various windows in an image, and it is possible to carry out a very fine-grained analysis. The measurement between two windows, x and y, of equal size N × N, is illustrated in equation 1.

$$SSIM(x,y) = \left(\frac{2\mu_x\mu_y}{\mu_x^2+\mu_y^2}\right) * \left(\frac{2\sigma_{xy}}{\vartheta_x+\vartheta_y}\right) = L_I * L_V \quad (3)$$

In equation 3, μ_x represents the average value for x, μ_y is the average value for y, ϑ_x and ϑ_y represent the variance of x and y, respectively. σ_{xy} represents the standard deviation of x and y. L_I and L_V refer to local luminance and local variance between x and y.

2. Categorical Cross-entropy Function

The categorical cross-entropy measures the difference between the true class distribution and the class probabilities of the prediction. It calculates cross-entropy loss for each class and sums them up. The formula for categorical cross entropy is as follows:

$$L(y_{true}, y_{pred}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left(y_{true}\, log(y_{pred})\right) \ldots (4)$$

In equation 4, L is the categorical cross-entropy loss function, y_{true} is the true class distribution (one-hot encoded), and y_{pred} is predicted class probabilities. C and N are the number of classes and image samples, respectively.

4. Experimental Setup

Some of the critical steps of the experimental setting to identify Monkey pox lesions with GNN include the captured MSLD images that were originally 224×224 pixels but re-sized to 512×512 pixels for processing. The data preprocessing was conducted using MATLAB R2020a, where data had to be manually freed from any background artifacts, followed by a median filter to remove noise and boost the image through a Laplacian filter. The entire dataset was utilized for both training and evaluation. For Felzenszwalb's algorithm, the hyper- parameter values are min_size set to 50 for the minimum size of segments, scale (k) set to 300 to permit a greater number of merges to form larger segments, σ set to 1.0 for Gaussian smoothing of the input image, and ρ set to 1.0. The Quickshift algorithm has a mode of similarity, which is Gaussian similarity based on the mean color of the segments, a kernel_size of 3, a max_dist of 6 to indicate the maximum distance between features, and a ratio of 0.5 to give equal weight to spatial and color distances. The hyper-parameters for a GNN model are four clusters (K), the Adam optimizer, and a learning rate of 0.001. Grid search techniques were employed to find the best hyper-parameters settings for Felzenswalb's and Quick-shift algorithms and GNN. Performance analysis was performed at 100, 200, 300, and 500 epochs. Over these epochs, the GNN learned the features and patterns of the dataset and accordingly adapted its parameters to fit in the best possible way to identify Monkey pox lesions. The configuration fully tested the model's capability in an unsupervised learning context.

4.1. Dataset Description

The Augmented MSLD dataset presented in this paper has 3192 skin images. The MSLD dataset had originally contained 228 images These images contained 102 images labeled "Monkey pox" and 126 images labeled "Others." In another study, the original dataset authors augmented the dataset using various image processing techniques, including rotation, translation, reflection, shearing, hue, saturation, contrast, brightness, noise addition, and scaling. This augmentation created an Augmented MSLD dataset containing 3192 images, with 1428 'Monkey pox' images and 1764 'Others' images.

Figure 4. (a) Monkey pox disease on hand; (b) Monkey pox disease on palm; (c) Monkey pox disease on whole body; (d) Monkey pox disease on foot; (e) Monkey pox disease on face; (f) non Monkey pox disease on the palm; (g) non Monkey key pox disease on whole body; (h) non Monkey key pox disease on back.

Images in Figure 4 depict various body parts infected by the Monkey pox disease, as indicated in Figures 4.a, 4.b, 4.c, 4.d, and 4.e. On the other hand, other images in this set, as indicated in Figures 4.f, 4.g, and 4.h, are various kinds of skin diseases unrelated to Monkey pox. In other words, Monkey pox is one of the rare viral diseases characterized by symptoms including Smallpox, such as chills, headache, muscle pains, and a characteristic rash. The rash starts on the face and then proceeds to all other body parts. It can also extend to the palms and soles.

4.2. Dataset Preparation

One of the biggest communities for free datasets is Kaggle, which offers a dataset named MSLD that includes skin photos with the disease and normal skin photographs [35]. The dataset initially consists of skin lesion images collected from the Kaggle repository. The images need to be processed before being given to the algorithm. Some image preprocessing techniques for segment extraction include applying a median filter to eliminate salt and Pepper noise and a Laplacian filter to boost the image. Image median filter is a widely applied technique, especially in cases where the images considered are impaired by salt and pepper noise. This filter works as a replacement filter for the central pixel with the median of the values of its surrounding pixels. A formula to apply median-filtered image processing shows the process step of the filter as well as its input filter window size, which is usually formed by a square matrix. The first step is to extract the values of all pixels centered on the current pixel to sort the extracted pixel values in ascending order for each pixel in the image [36]. The median value from the sorted list should be used to replace the current pixel. Algorithm 4 illustrates image smoothing of an N×N pixels image using a Median filter.

Algorithm 4 Image smoothing using Median Filter

Input: A $N \times N$ pixels skin lesion image I

Output: A $N \times N$ pixels skin lesion*n image $I_{smoothed}$ free of salt and pepper noise

1. Extend the boundaries by $\left\lfloor \frac{k}{2} \right\rfloor \left\lfloor \frac{k}{2} \right\rfloor$ of the input image I with zeros or replicate the nearest edge values to handle boundary cases.
2. Define the size of the neighborhood as $k \times k$ pixels.
3. For each pixel in the image, extract the $k \times k$ neighborhood centered on the pixel.
4. Sort the pixel values in the neighborhood in ascending order.
5. Replace the pixel value with the median value of the sorted neighborhood.
6. Repeat steps 2-4 for every pixel in the image.

The median filter has been demonstrated to be highly effective at removing salt and pepper noise by replacing outliers with more representative elements from the neighborhood. The other type of filter would be the Laplacian, a high-pass filter that enhances edges and features in an image [37]. To apply this filter, start by incorporating the filtered output into the original image. If the image is color, apply the filter separately to each channel. Finally, apply a M×M Laplacian filter to the gray-scale image using equation 5 below.

$$L(x,y) = \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} w(i,j) I(x+i, y+j) \quad \dots (5)$$

Here, $L(x, y)$ represents the filtered output at pixel (x, y), $I(x+i, y+j)$ represents the intensity value of the neighboring pixel $(x+i, y+j)$, and $w(i, j)$ represents the Laplacian filter coefficients. The Laplacian filter coefficients can vary depending on the specific Laplacian filter used. Algorithm 5 describes enhancing a $N \times N$ pixels image using a Laplacian filter.

Algorithm 5 Image enhancement using Laplacian filter

Input: A $N \times N$ pixels skin lesion image $I_{smoothed}$

Output: A $N \times N$ pixels high-boosted skin lesion image I'.

1. Define the $3x3$ Laplacian kernel (w)
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
2. Extend the boundaries of the input image $I_{smoothed}$ with zeros or replicate the nearest edge values to handle boundary cases.
3. For each pixel in the image, extract a 3×3 neighborhood centered on the pixel.
4. Compute the Laplacian response $L(x, y)$ using 2D convolution using equation 1.
5. Calculate the high-boosted pixel value $I'(x, y)$.
$$I'(x,y) = I(x,y) + \alpha L(x,y)$$
α is a user-defined parameter controlling the amount of high-frequency enhancement.
6. Repeat steps 2-4 for each pixel in the input image.
7. Normalize $I'(x, y)$ within the valid range (0-255).

5. Results and Discussion

5.1. Data Preprocessing

The MSLD was utilized to segment Monkey pox and no-no key pox blisters on human skin. First, data augmentation techniques were involved in augmenting the size and diversity of the dataset, which was required to improve the generalization capability of the model used for segmentation. The process started with denoising the images with a median filter suitable for edge preservation in filtering noise. The images were then denoised with a high-boost Laplacian filter, which highlighted small details and enhanced lesion characteristics in the images. Once the lesion features were enhanced, background removal was completed to isolate the skin region and ensure that only the relevant region was included for segmentation. The denoised images were segmented, and Felzenszwalb's and Quick Shift algorithms were used first to segment the images. Finally, an unsupervised GNN technique was used on the segmented images to enhance segmentation even further and get better results.

(a) (b)

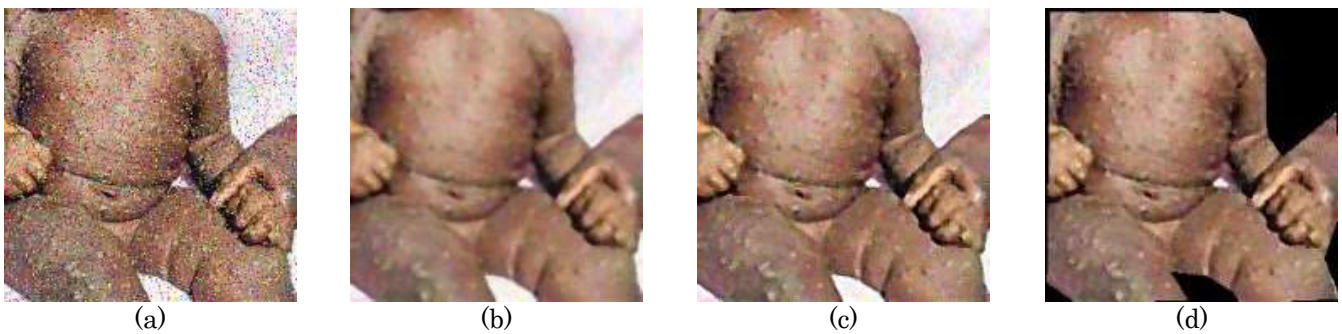
Figure 5. Image smoothing of Monkey pox blisters: (a) Image with salt and pepper noise; (b) Image after smoothing using Median filter.

Figure 5.a represents a Monkey pox skin blister image with salt and pepper noise, and Figure 5.b shows the result after applying a Median filter to remove the noise.

(a) (b)

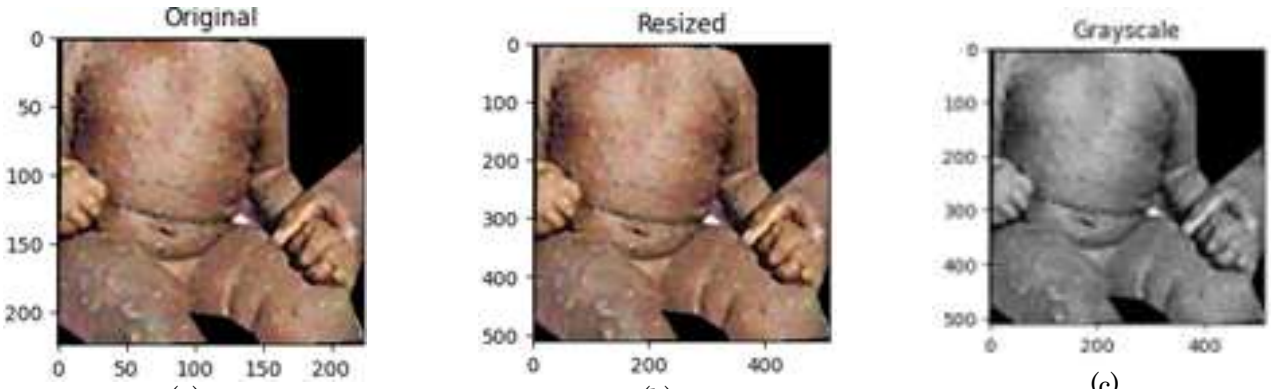
Figure 6. High boosting of image: (a) Image before applying the Laplacian filter for high-boosting; (b) Image after high-boosting using Laplacian filter.

Figure 6.a shows the smoothed image before applying the high-boosting filter, and Figure 6.b shows the image after applying the Laplacian high-pass filter to improve the sharpness in the image. After all the noise removal and boosting, the background is removed from all the images only to consider the skin area in the images. Figure 7 demonstrates the resulting images from various preprocessing stages on Monkey pox images. Figure 7.a shows an example from the MSLD dataset that includes salt and pepper noise. Figure 7.b shows the image after applying a Median low-pass filter to remove the noise. Figure 7.c shows an improvement in the image quality with a high-pass Laplacian filter. Finally, in Figure 7.d, the background noise of the enhanced Monkey pox image is manually and slowly removed as human intervention is noted as vital in image processing before the image can be input to algorithms for segmentation.



(a) (b) (c) (d)

Figure 7: Results of data preprocessing: (a) Original image with salt and pepper noise; (b) Image after smoothing using Median filter; (c) Image after high boosting using Laplacian filter; (d) Image after removing background.



(a) (b) (c)

Figure 8. Result of image re-sizing and color conversion: (a) Original image of size 224x224 pixels after smoothing using Median filter and high boosting using Laplacian filter; (b) Image after re-sizing into 512x512 pixels; (c) Gray-scale image.

Figure 8.a shows the input image of size 224x224 pixels upscaled to an image of 512x512 pixels as in Figure 8.b. The up-sampled image of size 512x512/3 is re-sized to a gray-scale of 512x512 pixels, as shown in Figure 8.c.

5.2. Image Segmentation

1. Felzenszwalb's Segmentation

Felzenszwalb's segmentation algorithm combines small regions from a preprocessed image through a similarity measurement process. This is performed on an image by treating each pixel as a region. Then, a region is merged if the similarity of those regions is less than some initial threshold. This process is repeated until all the regions satisfy the condition of similarity. Next, the segmented image with details is given as input to the GNN with the Adam optimizer and MinCutPool function for down sampling. The over-segmented image (Figure 9) refinement to an optimally segmented output is achieved by minimizing loss across the RAG.

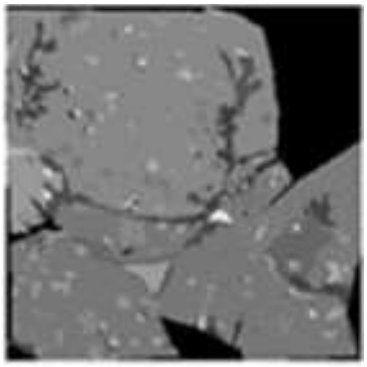
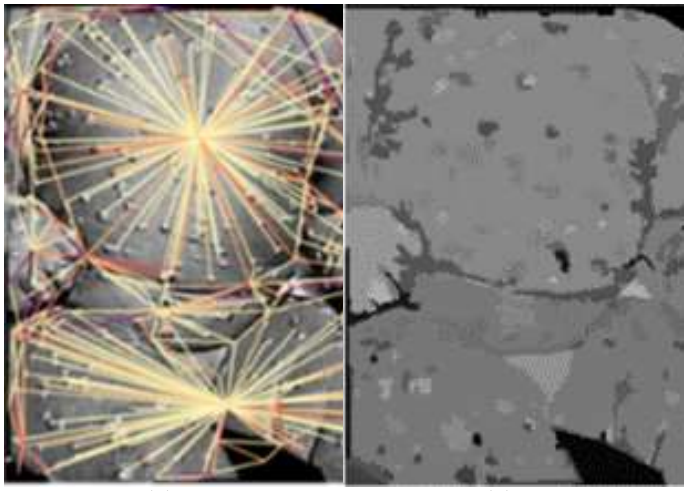


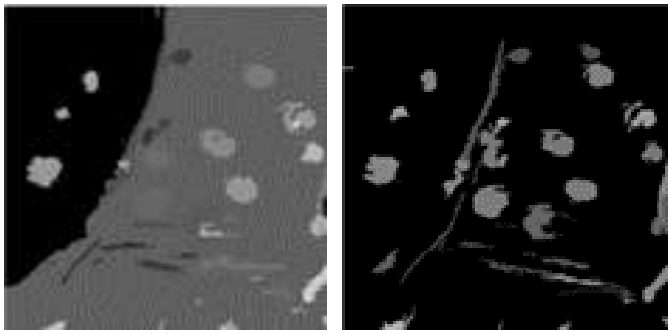
Figure 9. Segmented result of Figure 8.c by Felzenszwalb's algorithm.



(a) (b)

Figure 10. Optimized image segmentation result: (a) RAG; (b) segmented image after Mincut pooling.

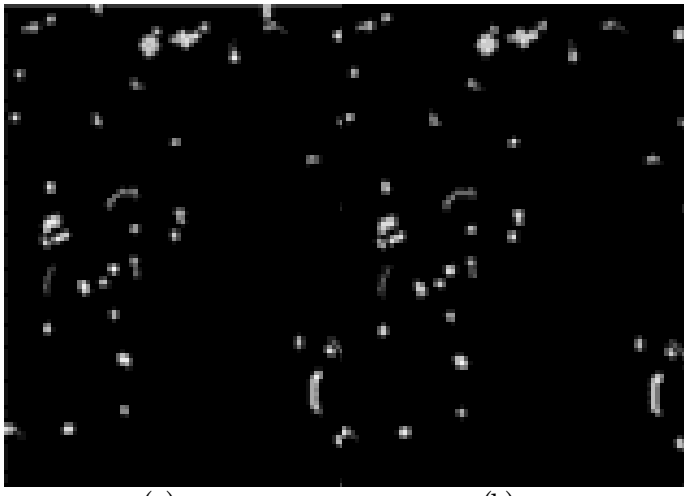
Figure 10 illustrates that the performance of the unsupervised segmentation algorithm improves steadily as the loss value is reduced. As observed, at 500 iterations, it has a considerably lower loss. The total loss of Felzenszwalb's segmentation algorithm and GNN on the test set is 0.0172. Figure 10.a represents the RAG of the segmented output. Figure 10.b represents the segmented image obtained by applying Mincut pooling.



(a) (b)

Figure 11. Segmentation results of Monkeypox Skin lesion image: (a) Segmented image by Felzenszwalb's algorithm; (b) Segmented image after Mincut pooling.

The better image segmentation results show a well-coupled integration of Felzenszwalb's Segmentation Algorithm with GNN processing. Figure 11.a illustrates raw segmentation outcome by the Felzenszwalb algorithm, which captures details in the Monkey pox skin lesion image, but with some over-segmentation. On the other hand, Figure 11.b shows refinement in finer segmentation using the output of Mincut pooling optimized with GNN. The above refinement was achieved during over 500 training iterations with a loss of 0.0172 value, and the segmentation are mainly coherent and semantically meaningful. The above results prove that the algorithm is powerful enough to obtain lesions of Monkey pox, which must be differentiated from the skin tissues.



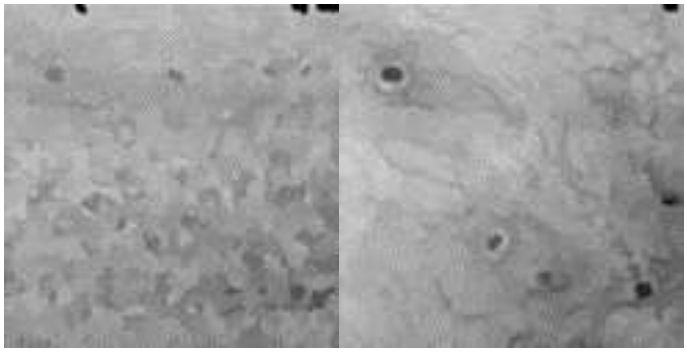
(a) (b)

Figure 12. Segmentation results of no-no key pox skin lesion image: (a) Segmented image by Felzenszwalb's algorithm; (b) Segmented image after Mincut pooling.

Figure 12 displays better segmentation of the no-no key pox skin lesions. The over-segmented input in Figure 12.a captures the most detailed features. However, the improvement was in segmentation, as observed in Figure 12.b, which follows Mincut pooling. This optimization has over 300 iterations and a loss value of 0.0271. That is helpful with precise lesion identification and diagnosis.

2. Quick Shift Segmentation

The Quick Shift algorithm successively aggregates similar pixels into larger, irregularly shaped segments in an iterative manner. For the Quick Shift approach, there is the input image with a segmenting capability more than for Felzenszwalb's algorithm, and these segments are fed into a GNN with Adam optimizer and Mincut pooling functions for down sampling. This improves the segmentation, predicting the best possible segment groupings while reducing the loss. The output of the over-segmented image and an optimized segmented image by GNN is displayed, which gives a summarized illustration of the whole segmentation process.



(a) (b)

Figure 13. Segmentation results of Monkey pox Skin lesion image: (a) Segmented image by Quick Shift algorithm; (b) Segmented image after Mincut pooling.

Figure 13, shows the result after segmentation by combining the Quick Shift algorithm with the GNN processing. Figure 13.a displays the initial over-segmented version where the segments are huge and not so appealing when similar pixels are gathered together. Then, it shows the optimized result obtained after GNN-based optimization with Mincut pooling for coherent and semantically meaningful segments. Figure 13.b shows the post-optimization using GNN-based optimization after employing Mincut pooling to ensure coherent, semantically meaningful segments. This optimization was

conducted over 500 iterations, yielding a minimum loss value of 0.0293.

(a) (b)

Figure 14. Segmentation results of no-no key pox skin lesion image: (a) Segmented image by Quick Shift algorithm; (b) Segmented image after Mincut pooling.

Figure 14 shows the refined segmentation results for No-no key pox skin lesions. In Figure 14.a, the image is over-segmented, revealing details of initial segmentation. Figure 14.b shows results after refinement iterations with the MinCut pooling procedure, reporting better accuracy. This single process took 500 iterations before converging at a loss of 0.0301, demonstrating the accurate identification of lesions.

5.3. Results and Analysis

Table 1 presents summarized performance parameters for the segmentation algorithm of Felzenszwalb combined with a growing neural network for seven trained approaches for monkey pox and no-no key pox skin lesion images for several iterations (100, 200, 300, and 500). The loss reduces with the increasing number of iterations, indicative of the algorithm's improving performance. Figure 10 demonstrates that the unsupervised segmentation algorithm performs better by gradually minimizing the loss value. Notably, at 500 iterations, it achieves a significantly reduced loss. The total loss achieved by Felzenszwalb's segmentation algorithm and GNN on the test data is 0.0172. The presented trend of loss values indicates that the segmentation algorithm achieves higher precision and efficiency in differentiating between lesions from Monkey pox and other skin lesions. Such results are significant for estimating the ability of the algorithm to identify and segment lesions caused by Monkey pox with a suitable level of accuracy in medical images.

Table 1. Losses incurred by Felzenszwalb's segmentation algorithm and GNN in different iterations.

Felzenszwalb + GNN	100	200	300	500
Monkeypox	0.0536	0.0363	0.0264	0.0172
Non-Monkeypox	0.0567	0.0374	0.0271	0.0166

Table 2 shows the loss values captured at different iterations for the combined Quick Shift segmentation algorithm with GNN for both Monkey pox and no-no key pox images. The loss values represent the difference between the predicted outcomes and the true results, with low values showing the improvement of the algorithms. Figure 13.b shows the post-optimization using GNN-based optimization after employing Mincut pooling to ensure coherent, semantically meaningful segments. This optimization was conducted over 500 iterations, yielding a minimum loss value of 0.0293. The iteration count indicates the number of training sessions completed by the combination of algorithms, each focusing on continuous improvements in segmentation accuracy. Thus, the clear decreases in loss values between iterations show how well the algorithm combinations segment Monkey pox lesions from other types of skin lesions.

Table 2. Losses incurred by Quick shift segmentation algorithm and GNN in different iterations.

Quick Shift + GNN	100	200	300	500
Monkey pox	0.0754	0.0586	0.0392	0.0293
No-no key pox	0.0813	0.0612	0.0404	0.0301

Two segmentation algorithms, Felzenszwalb's segmentation algorithm with GNN and Quick Shift segmentation algorithm with GNN, have been compared. Later, SSIM values have been calculated for both algorithms. The optimized Felzenszwalb's segmentation algorithm with GNN has an SSIM value of 0.85 ± 0.091, whereas the Quick Shift segmentation algorithm combined with GNN has an SSIM value of 0.63 ± 0.080. The experiments indicate that Felzenszwalb's algorithm gives better segmentation quality. Felzenszwalb's algorithm and GNN produced a Monkey pox region accuracy of 77.6% and a no-no key pox region accuracy of 92.47%, makes the overall accuracy 85.03%. The scores reflect the strength of the segmentation

model to accurately detail both Monkey pox, and non Monkey pox regions from images of human skin. GNNs are especially valuable for image segmentation tasks because they create relationships between neighboring segmented image pixels; thus, segmentation accuracy improves. The observed loss arising from Felzenszwalb's algorithm combined with a GNN on the test data was 0.0082. Such a low loss proves the efficiency of this combined approach. The results show that the incorporation of GNNs into the process of segmentation of Monkey pox blisters can be used to enhance the accuracy of the segmentation and, ultimately, raise the accuracy rate of diagnosis and treatment for Monkey pox infections.The following table compares these results with those of previous implementations.

Table 3. Comparison between performances of proposed work with existing implementations.

Literature	Model	Tasks	Performance Metrics
Castro et al. [13]	C-Means Algorithm	Segmentation	Acc: 69.3%
McNeil et al. [14]	U-Net	Segmentation	Dice Coefficient:0.74
Muñoz-Saavedra et al. [15]	VGG-19	Classification	Acc: 93.33%
	Ensemble of ResNet50 + EfficientNetB0 + MobileNetV2		Acc: 98.33%
Ahsan et al. [16]	MobileNetV2	Detection and classification	Acc: 99%
Chakroborty [17]	ResNet50	Detection	Acc: 82.96% Precision: 87%
Alhasson et al. [18]	ShuffleNet-V2	Detection and classification	Acc: 79%
Yadav et al. [19]	DenseNet-121	Classification	Acc: 93.7%
Sitaula et al. [20]	Ensemble of Xception + DenseNet-169	Detection and classification	Acc: 87.13%
Yang et al. [21]	FCNResNet10	Segmentation and classification	Precison: 0.9102
Sahin et al. [4]	MobileNetV2	Classification	Acc: 91.11 %
Bamini et al. [22]	CoAtNet model	Segmentation and Classification	Acc: 95.42%
Srinivasan et al. [23]	ESACN	Detection and classification	Acc: 92.68%,
Pratama et al. [24]	Xception + InceptionV3	Detection	Acc: 85.96%
Vandana et al. [25]	MRpoxNet	Detection	Acc: 98.1%
Yue et al. [26]	MpoxMamba	Detection	Acc: 82.47%
Our work	Felzenszwalb + GNN	Segmentation	Acc: 85.03% SSIM: 0.85 ±0.091

Table 3 provides a comparison of research work in this group of Monkey pox diagnoses in terms of models used and their corresponding performance indicators. Each entry in the table is a study, model taken, and the resultant suitable performance obtained. Castro et al. utilized the C-Means Algorithm for the segmentation step with an accuracy of 69.3% [13]. McNeil et al. used U-Net for segmentation and attained a Dice coefficient of 0.74, proving that the model could define the Monkey pox lesions distinctly [14]. Muñoz-Saavedra et al. utilized VGG-19 for classification tasks, achieving an accuracy of 93.33%. In contrast, an ensemble using ResNet50, EfficientNetB0, and MobileNetV2 achieved a better accuracy of 98.33% in their study [15]. Ahsan et al. achieved an accuracy of 99% in a detection and classification task using MobileNetV2 [16]. Chakroborty discussed the ResNet50 model for detection and established an accuracy of 82.96% and a precision of 87% [17]. Alhasson et al. discussed the ShuffleNet-V2-based model, which had an accuracy of 79% for the detection and classification tasks [18]. Yadav et al. discussed the application of DenseNet-121, which had an accuracy of 93.7% [19].

Sitaula et al. used a combination of Xception and DenseNet-169 to get a detection and classification

accuracy of 87.13% [20]. One new method is Yang et al.'s FCNResNet10, which worked well for separating and classifying the Monkey pox lesions with an accuracy of 0.9102 [21]. Sahin et al. used MobileNetV2 to classify things and got an accuracy of 91.11% [4]. Bamini et al. used a hybrid CoAtNet architecture on the MSLD v2 for better lesion segmentation and classification. They got 95.42% accuracy, 95.74% recall, and 94.12% F1-score [22]. Srinivasan et al.'s ESACN detected Monkey pox with 92.68% accuracy, an F1 score of 96.73% and uses a multi-class dataset of skin lesions [23]. Pratama et al. made a model with the properties of Xception and InceptionV3. It could detect Monkey pox skin lesions with an accuracy of 85.96%, precision of 86.47%, recall of 85.25%, and an AUC of 0.8931 [24]. Vandana et al. presented MRpoxNet, which was trained with a larger dataset of the MSID and performed better than other models, and achieved 98.1% accuracy [25]. Yue et al. created MpoxMamba, a lightweight hybrid architecture that used the MSLD dataset and detected Monkey pox with an 82.47% accuracy [26].

In recent studies of Monkey pox, different datasets have been used for segmentation and detection tasks. Castro et al. [13] used C-Means to divide CT images of non-human primates with 69.3% accuracy. McNeil et al. [14] used U-Net on a private clinical image dataset and got a Dice coefficient 0.74. Ahsan et al. [16] used a Kaggle dataset with MobileNetV2 to find and sort objects with 99% accuracy. Chakroborty [17], Alhasson et al. [18], and Sitaula et al. [20] worked with the MSLD dataset to identify lesions and reported accuracy rates of 82.96%, 79%, and 87.13%, respectively.

Yang et al. used FCNResNet10 for segmentation and classification on their dataset, achieving a precision score of 0.9102 [21]. CoAtNet received an accuracy score of 95.42% on the MSLD v2 dataset [22], and ESACN got a 92.68% accuracy score for retrieving Monkeypox on a multi-class dataset [23]. A hybrid model combining Xception and InceptionV3 reached 85.96% accuracy with 86.47% precision on the MSLD dataset [24]. MRpoxNet outperformed the others with a 98.1% accuracy on the larger MSID dataset [25]. Yue et al. achieved 72.33%, EfficientNet-B1 71.02%, MobileOne-S0 73.58%, MobileViT-XS 74.92%, FastViT-t8 75.00%, PoxNet22 74.50%, MpoxNet 74.94%, Swin Transformer-T 77.63%, MonkeyNet 79.44%, and MpoxMamba 82.47% on the MSLD dataset [26].

On the other hand, our work is about segmentation using the MSLD dataset with an unsupervised approach that combines the Felzenszwalb and QuickShift algorithms and optimizes them using GNN. This method gets 85.03% accuracy and an SSIM of 0.85 ± 0.091. This performance is competitive, as it outperforms some CNNs [17, 18] and Transformer models [26], and comes close to ensemble and hybrid methods [20, 24]. It doesn't need any big annotated datasets, though. The high SSIM score indicates that the method can further improve high-quality lesion segmentation. This shows that it could be a strong and efficient alternative to traditional supervised deep learning methods. The MSLD dataset samples were used to test segmentation and identification tasks using low-resource hardware. It included a single-core hyper-threaded Xeon processor running at 2.3 GHz with 13 GB of DDR4 RAM and execution time recorded for a 512x512 pixels RGB image. ResNet50 takes 1.185 seconds, ShuffleNet-V2 takes 0.0594 seconds, FCNResNet10 takes 1.93 seconds, MobileNetV2 takes 0.110 seconds, C-Means takes 0.0867 seconds, U-Net takes 13.87 seconds, and the ensemble of Xception with DenseNet-169 takes 3.08 seconds. CoAtNet takes 6.68 seconds to perform skin lesion segmentation, given a 512x512 pixels RGB image. ESACN takes 3.77 seconds to do the same detection task and the hybrid model of Xception and Inception V3 takes 8.81 seconds, but MRpoxNet is fastest, taking only 0.87 seconds to complete skin lesion detection. The proposed unsupervised approach utilizing Felzenszwalb's algorithm optimized with GNN shows significantly faster computation times: 0.0299 seconds for 100 iterations, 0.0347 seconds for 200 iterations, 0.0397 seconds for 300 iterations, and 0.0496 seconds for 500 iterations. The proposed GNN-based segmentation system has potential use cases in real-world clinical environments. This pipeline can be applied in digital dermatoscopy or teledermatology systems to aid practitioners by automatically segmenting lesions and providing severity indicators based on lesion characteristics. Further, it may be integrated into mobile phone-based automated health-screening programs for early, remote detection of Monkey pox in those areas with limited resources or high outbreak risk. The process offers accurate and consistent analysis of skin lesion images, supporting faster clinical decision-making, improved record-keeping, and reduced diagnostic variability in both hospital and field settings. Future studies can concentrate on creating a light, optimized version of the model for real-time applications to run on embedded systems.

6. Conclusions

This study introduces a novel approach for segmenting Monkey pox blisters with GNNs coupled with state-of-the-art segmentation algorithms, namely Felzenszwalb's and Quick Shift. The method shows better segmentation accuracy with an SSIM of 0.85 ± 0.091 and decreased test loss, which reflects better performance compared to classical approaches. The integration of GNN with segmentation techniques greatly improves the accuracy of lesion detection, and it is strongly applicable in medical image analysis, especially when it comes to detecting skin diseases such as Monkey pox. The results of the segmentation were significant but not better than the best algorithms in their respective areas and suggested that the methodologies in this work can be optimized in several ways, particularly in the understanding of how the models generalize to other datasets. The GNN model's disregard for computational efficiency in real-time clinical applications was one of its drawbacks. Under normal conditions, GNN and Felzenszwalb's algorithms are more reliable and effective; however, Quick Shift segmentation did not appear to function as well in these conditions. This suggests that developing segmentation or hybrid algorithms might offer opportunities for additional research. Furthermore, given the significant differences across the sources of imaging-based data, it seems reasonable to consider the use of multi-modal image data for increased robustness. Overall, the study's findings offer a solid basis for upcoming research prospects in medical imaging, such as automated diagnosis and lesion categorization, as well as GNN-based medical applications.

Acknowledgments: We thank the Chanakya Fellowship Scheme, supported by the iHub Anubhuti-IIITD Foundation and financially aided by the DST, Government of India, for ten months.

Conflicts of Interest: The authors declare no conflict of interest.

Funding Statement: None

References

[1] Lowenstine, L.J.; McManamon, R.; Terio, K.A.; "Apes". In Pathology of Wildlife and Zoo Animals; Terio K.A., McAloose D. and Leger J.S., Eds.; Academic Press: USA, 375–412, 2018.

[2] Ogoina, D.; Damon, I.; Nakoune, E.; "Clinical review of human mpox". Clin. Microbiol. Infect., 29(12): 1493-1501, 2023.

[3] Saxena, S.K.; Ansari, S.; Maurya, V.K.; Kumar, S.; Jain, A.; Paweska, J.T.; Tripathi, A.K.; et al.; "Re-emerging human monkeypox: a major public-health debacle". J. Med. Virol., 95(1): e27902, 2023.

[4] Sahin, V.H.; Oztel, I.; Yolcu Oztel, G.; "Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application". J. Med. Syst., 46(11): 79, 2022.

[5] Ahsan, M.M.; Uddin, M.R.; Ali, M.S.; Islam, M.K.; Farjana, M.; Sakib, A.N.; Momin, K.A.; et al.; "Deep transfer learning approaches for Monkeypox disease diagnosis". Expert Syst. Appl., 216; 119483,1-20; 2023.

[6] Almutairi, S.A.; "DL-MDF-OH2: Optimized deep learning-based monkeypox diagnostic framework using the metaheuristic Harris Hawks Optimizer Algorithm". Electronics, 11(24): 4077,1-19; 2022.

[7] Okoli, G.N.; Van-Caeseele, P.; Askin, N.; Abou-Setta, A.M.; "A global systematic evidence review with meta-analysis of the epidemiological characteristics of the 2022 Mpox outbreaks". Infection, 52(3): 901-921, 2024.

[8] Onifade, A.A.; Akindele, O.A.; Ahmad, I.; Khan, M.A.; Mat Isa, N.; Alzahrani, E.; "Modeling monkeypox transmission with a compartmental framework to evaluate testing, isolation and public awareness strategies". Sci. Rep., 15(1): 27236, 2025.

[9] Öztürk, Ş.; Özkaya, U.; "Skin lesion segmentation with improved convolutional neural network". J. Digit. Imaging, 33(4): 958–970, 2020.

[10] Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W. and Leskovec, J.; "Hierarchical graph representation learning with differentiable pooling". In: Advances in Neural Information Processing Systems, Montréal, Canada, 03-08 December; Bengio S., Wallach H., Larochelle H., Grauman K., Cesa-Bianchi N. and Garnett R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 1-11, 2018.

[11] Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S.; "A comprehensive survey on graph neural networks". IEEE Trans. Neural Netw. Learn. Syst., 32(1): 4–24, 2020.

[12] Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; "Unsupervised graph representation learning

with cluster-aware self-training and refining". ACM Trans. Intell. Syst. Technol., 14(5); 1-21, 2023.

[13] Castro, M.A.; Thomasson, D.; Avila, N.A.; Hufton, J.; Senseney, J.; Johnson, R.F. and Dyall, J.; "Optimization of automated segmentation of monkeypox virus-induced lung lesions from normal lung CT images using hard C-means algorithm". In: Medical Imaging 2013: Biomedical Applications in Molecular, Structural, and Functional Imaging, Florida, United States, 9-14 February; Weaver J.B. and Molthen R.C., Eds.; SPIE: Bellingham, Washington, USA, 482–487, 2013.

[14] McNeil, A.J.; House, D.W.; Mbala-Kingebeni, P.; Mbaya, O.T.; Dodd, L.E.; Cowen, E.W.; Nussenblatt, V.; et al.; "Counting monkeypox lesions in patient photographs: Limits of agreement of manual counts and artificial intelligence". J. Invest. Dermatol., 143(2): 347–350, 2023.

[15] Muñoz-Saavedra, L.; Escobar-Linero, E.; Civit-Masot, J.; Luna-Perejón, F.; Civit, A.; Domínguez-Morales, M.; "A robust ensemble of convolutional neural networks for the detection of monkeypox disease from skin images". Sensors, 23(16): 7134, 2023.

[16] Ahsan, M.M.; Ali, M.S.; Hassan, M.M.; Abdullah, T.A.; Gupta, K.D.; Bagci, U.; Kaushal, C; et al.; "Monkeypox diagnosis with interpretable deep learning". IEEE Access, 11: 81965-81980, 2023.

[17] Chakroborty, S.; "A hybrid deep learning framework for early detection of Mpox using image data". Healthc. Anal., 7: 100396, 2025.

[18] Alhasson, H.F.; Almozainy, E.; Alharbi, M.; Almansour, N.; Alharbi, S.S.; Khan, R.U.; "A deep learning-based mobile application for monkeypox detection". Appl. Sci., 13(23): 12589, 2023.

[19] Yadav, S.; Qidwai, T; "Machine learning-based monkeypox virus image prognosis with feature selection and advanced statistical loss function". Med. Microecol., 19: 100098, 2024.

[20] Sitaula, C.; Shahi, T.B.; "Monkeypox virus detection using pre-trained deep learning-based approaches". J. Med. Syst., 46(11): 78, 2022.

[21] Yang, T.; Yang, T.; Liu, A.; An, N.; Liu, S.; Liu, X.; "AICOM-MP: an AI-based monkeypox detector for resource-constrained environments". Connect. Sci., 36(1): 2306962, 2024.

[22] Bamini, A.; Kumar, J.N.A.; Jayapratha, C.; Govindaprabhu, G.B.; "Revolutionizing Monkeypox Diagnosis: A Cutting-Edge Deep Learning Pipeline for Advanced Lesion Segmentation and Classification Using CoAtNet." J. Appl. Comput. Sci., 6(1): 1420-1439, 2025.

[23] Srinivasan, M.N.; Sikkandar, M.Y.; Alhashim, M.; Chinnadurai, M.; "Capsule network approach for Monkeypox (CAPSMON) detection and subclassification in medical imaging system." Sci. Rep., 15(1): 3296, 2025.

[24] Pratama, N.R.; Setiadi, D.R.I.M.; Harkespan, I.; Ojugo, A.A.; "Feature Fusion with Albumentation for Enhancing Monkeypox Detection Using Deep Learning Models." J. Comput. Theor. Appl., 2(3): 427–440, 2025.

[25] Vandana; Sharma, C.; Shah, M.A.; "MRpoxNet: An enhanced deep learning approach for early detection of monkeypox using modified ResNet50." Digit. Health, 11: 1-18, 2025.

[26] Yue, Y.; Xue, J.; Liang, H.; Li, Z.; & Wang, Y.; "MpoxMamba: A Grouped Mamba-based Lightweight Hybrid Network for Mpox Detection". In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Hyderabad, India, 06-11 April; Rao B.D., Trancoso T., Sharma G. and Mehta N.B., Eds.; IEEE: 1-5, 2025.

[27] Chaudhuri, U.; Banerjee, B.; Bhattacharya, A.; "Siamese graph convolutional network for content-based remote sensing image retrieval". Comput. Vis. Image Underst., 184: 22–30, 2019.

[28] Grattarola, D.; Zambon, D.; Bianchi, F.M.; Alippi, C.; "Understanding pooling in graph neural networks". IEEE Trans. Neural Netw. Learn. Syst., 35(2): 2708-2718. 2022.

[29] Felzenszwalb, P.F.; Huttenlocher, D.P.; "Efficient graph-based image segmentation". Int. J. Comput. Vis., 59(2): 167–181, 2004.

[30] Zhang, S.; Ma, Z.; Zhang, G.; Lei, T.; Zhang, R.; Cui, Y.; "Semantic image segmentation with deep convolutional neural networks and quick shift". Symmetry, 12(3): 427, 2020.

[31] Bianchi, F.M.; Grattarola, D.; Alippi, C.; "Spectral clustering with graph neural networks for graph pooling". In: International Conference on Machine Learning, Vienna, Austria, 12-18 July; Daumé III H. and Singh A., Eds.; PMLR, ML Research Press: Cambridge, MA, USA, 874–883, 2020.

[32] Zhou, S.; Xu, H.; Zheng, Z.; Chen, J.; Li, Z., Bu, J.; Wu, J.; et al.; "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions". ACM Comput. Surv., 57(3): 1-38. 2024.

[33] Mesquita, D.; Souza, A.; Kaski, S.; "Rethinking pooling in graph neural networks". In Advances in Neural Information Processing Systems; Larochelle H., Ranzato M., Hadsell R., Balcan M.F. and Lin H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 1-12, 2020.

[34] Chen, C.; Wu, Y.; Dai, Q.; Zhou, H.Y.; Xu, M.; Yang, S.; Han, X.; et al.; "A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective". IEEE Trans. Pattern Anal. Mach. Intell., 46(12): 10297-10317, 2024.

[35] Thorat, R.; Gupta, A.; "Transfer learning-enabled skin disease classification: the case of monkeypox detection". Multimed. Tools Appl., 83(35): 82925-82943, 2024.

[36] Sudan, S.; "Median filter performance based on different window sizes for salt and pepper noise removal in gray and RGB images". Int. J. Signal Process., Image Process. Pattern Recognit., 8(10): 343-352, 2015.

[37] Siddiqui, M.H.; "Application of 2D convolution in digital image processing". Asian J. Eng. Sci. Technol., 6(1): 20-23, 2016.