

Enhancing Monkeypox diagnosis and explanation through modified transfer learning, vision transformers, and federated learning

Md Manjurul Ahsan^a, Tasfiq E. Alam^a, Mohd Ariful Haque^b, Md Shahin Ali^{c,*},
Rakib Hossain Rifat^d, Abdullah Al Nomaan Nafi^e, Md Maruf Hossain^c, Md Khairul Islam^c

^a Department of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, 73019, USA

^b Department of Cyber-Physical Systems, Clark Atlanta University, Atlanta, 30314, GA, USA

^c Department of Biomedical Engineering, Islamic University, Kushtia, 7003, Bangladesh

^d Department of Computer Science and Engineering, BRAC University, Dhaka, 1212, Bangladesh

^e Department of Information and Communication Technology, Islamic University, Kushtia, 7003, Bangladesh

ARTICLE INFO

Keywords:

Transfer learning
Vision transformer
Disease diagnosis
Image segmentation
LIME

ABSTRACT

The Monkeypox outbreak has emerged as a pressing global health challenge, evidenced by rising cases across nations. Individuals afflicted exhibit diverse dermatological symptoms that risk further transmission via contamination. Our study assessed the efficacy of three modified transfer learning models (M-VGG16, M-ResNet50, M-ResNet101) alongside vision transformers (ViT) across four investigations. We achieved high accuracy in discriminating Monkeypox cases, with M-VGG16 achieving 88%, 76%, and 77% accuracy in Studies One, Two and Four and M-ResNet50 achieving 89% in Study Three. To comprehend triggers for Monkeypox onset, we utilized Local Interpretable Model-Agnostic Explanations (LIME) to explain predictions visually. LIME alignments underscored our models' high accuracy, correlating with segmented identification of infected regions. Further, we implemented Federated Learning on decentralized data to evaluate generalization capabilities. Blending established deep learning with emerging decentralized learning and explanation techniques is vital in improving predictive accuracy and elucidating Monkeypox intricacies amid the persisting global outbreak. Our study emphasizes the continued relevance of pioneering techniques while introducing new approaches to address this major health challenge.

1. Introduction

Monkeypox is a zoonotic viral disease caused by the Monkeypox virus. This virus infects primates, such as monkeys and rodents, but can also be transferred to people by frequent contact with infected animals or human-to-human transmission [1]. The initial identification of the Monkeypox virus occurred in 1958 when it was first discovered in a monkey at a laboratory in Copenhagen, Denmark. This finding was a significant milestone in the understanding and study of this virus, which has since become a subject of ongoing research and monitoring due to its potential impact on human health [2]. The first documented human infection with the Monkeypox virus was reported in 1970 in the Democratic Republic of the Congo during an ongoing campaign to eliminate smallpox [3].

The Monkeypox virus spreads through contact with infected people, animals, or materials. It transmits via physical contact, animal bites, respiratory droplets, or mucosal secretions. Symptoms include fever, headaches, muscle and back pain, swollen lymph nodes, chills, fatigue, and a rash that starts on the face and spreads. Monkeypox's high contagion rate and limited treatments make it a major public health concern [4].

In 1990, there were just 50 documented cases in West and Central Africa [5]. By 2020, this number had increased to 5000. Historically, it was believed that Monkeypox exclusively occurred in Africa, but in 2022, viral cases were recorded in various non-African nations, including the United States and Europe [6]. As of January 31, 2023, the Centers for Disease Control and Prevention (CDC) reported that 85,469 individuals had been diagnosed with Monkeypox in 94 nations [7].

* Corresponding author.

E-mail addresses: ahsan@ou.edu (M.M. Ahsan), tasfiq@ou.edu (T.E. Alam), mohdariful.haque@students.cau.edu (M.A. Haque), shahinbme.iu@gmail.com (M.S. Ali), rakib.hossain.rifat@g.bracu.ac.bd (R.H. Rifat), nomaan.ict.iu@gmail.com (A.A.N. Nafi), marufbme.iu@gmail.com (M.M. Hossain), khairul.ice06@gmail.com (M.K. Islam).

<https://doi.org/10.1016/j.imu.2024.101449>

Received 11 October 2023; Received in revised form 28 December 2023; Accepted 13 January 2024

Available online 18 January 2024

2352-9148/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Currently, there is no proper treatment for Monkeypox; however, symptoms can be managed. To reduce the risk of contracting the disease, one should avoid contact with infected animals and maintain good hygiene practices to prevent transmission. Vaccination is considered the most effective way to combat Monkeypox. While vaccines for the virus have been approved by the Food and Drug Administration (FDA), they have yet to be administered to humans in the United States. In other nations, vaccines developed for smallpox are utilized to treat Monkeypox [8]. Monkeypox diagnosis starts with evaluating skin lesions and exposure history [9]. Confirmation involves testing lesions using Polymerase Chain Reaction (PCR) and Enzyme-linked Immunosorbent Assay (ELISA) to detect the virus. Conducted in Biosafety Level 3 laboratories by trained staff, these tests confirm Monkeypox and maintain safety [10].

The application of Machine Learning (ML) in medical imaging and disease diagnosis is a significant advancement in the field of Artificial Intelligence (AI). ML's rapid and precise decision-making capabilities have made it a valuable tool in diagnostic imaging. A notable example is the computer-aided diagnosis (CAD) system developed by Miranda and Felipe for breast cancer diagnosis, utilizing fuzzy logic to handle uncertainty and imprecision. This system analyzes imaging features like contour, density, and form to determine diagnoses, mimicking a radiologist's reasoning with reduced computational time, enhancing both accuracy and efficiency in medical imaging [11]. Additionally, Ardakani et al.'s research highlights the efficacy of Deep Learning (DL) in identifying COVID-19. By evaluating ten different DL models using data from 108 COVID-19 patients and 86 non-COVID-19 patients, they achieved a remarkable 99% accuracy rate [12]. These examples underscore how ML and DL are revolutionizing medical diagnostics, offering more accurate, efficient, and reliable methods for disease detection and analysis [13].

In recent years, various studies have been conducted using DL models to classify human Monkeypox using image analysis techniques. Most of the research considered different pre-trained models for better and more effective performance. For instance, Sahin et al. evaluated various pre-trained models, including ResNet18, GoogleNet, EfficientNetB0, NasnetMobile, ShuffleNet, and MobileNetv2, using the Monkeypox skin lesion dataset (MSLD). The study found that MobileNetv2 had the highest accuracy of 91.11% in binary classification [14]. Similarly, in a study by Sitaula and Shahi, different pre-trained DL models were compared. The study found that an ensemble approach performed best, achieving a precision of 0.85, recall of 0.85, F1-score of 0.85, and accuracy of 87.13% [15].

Akin et al. suggested a Convolutional Neural Network (CNN)-based auxiliary decision aid system for hospitals. The author considered 572 images to develop the model and achieved an accuracy of 83.89% using Xception-CBAM-Dense for binary classification. The study also employed MobileNetV2, which had an F1-score of 0.98 and an accuracy of 98.25%, as well as sensitivity, and specificity of 1.0. Finally, the model's predictions have been interpreted using the Gradient-Weighted Class Activation Mapping (GradCAM) visualization tools [16].

Irmak, Aydin, and Yağanoğlu aimed to detect Monkeypox skin disease using DL-based models, and the study considered a dataset consisting of 770 images. Two pre-trained models, MobileNetV2 and VGGNet, were used in the study. The results showed that MobileNetV2 had the highest performance in terms of accuracy (91.38%), precision (0.90), recall (0.86), and F1-score (0.88) for the multiclass classification task. The author concluded that MobileNetV2 is a suitable model for detecting Monkeypox disease diagnosis, with high accuracy and performance in both binary and multiclass classification tasks. Ensemble approaches and interpretation methods such as Local Interpretable Model-Agnostic Explanations (LIME) have also been shown to improve performance [17]. However, more research is needed to improve the models' performance further and explore other datasets and methods for the classification of Monkeypox. Alcalá-Rmz et al. present a novel approach to diagnosing exanthematous diseases utilizing infected

images and DL algorithms. The authors employ MiniGoggleNet, a DL-based model, which comprises 2067 images. Their preliminary findings demonstrate promising performance, with an accuracy of 97% and an AUC of 0.76 for binary classification. Nevertheless, the study does have certain limitations, such as the lack of investigation into multiclass classification and the absence of explainable AI techniques to make the DL model's predictions more interpretable. Future studies should address these limitations to enhance the understanding and reliability of this diagnostic approach [18].

In medical imaging, the application of AI and ML has shown immense potential for revolutionizing disease diagnosis and treatment. However, the integration of AI in this domain is accompanied by significant challenges, especially concerning data privacy and security. Since medical images contain sensitive and confidential patient information, it is necessary to safeguard patient privacy while ensuring accurate disease detection.

Over the years, Federated Learning (FL) has emerged as a promising solution to address these privacy concerns in the context of medical imaging [19]. Unlike traditional centralized ML approaches, FL enables model training across multiple distributed data sources without requiring the raw data to be centralized. Instead, individual data sources, such as medical institutions or hospitals, retain complete control over their data, allowing them to keep sensitive images within their secure confines [20]. Only model updates, not raw data, are exchanged during collaborative learning, safeguarding patient privacy [21].

Based on the overall analysis, it can be inferred that very limited research has been conducted using DL-based approaches to diagnose Monkeypox. Most of the literature has only focused on Transfer Learning (TL) based approaches. Furthermore, none of the existing approaches provides information on their models' computational efficiency or interpretability using interpretable ML techniques. As an effect, there is a need to evaluate the performance of TL-based approaches with various hyperparameter settings in order to develop a robust ML-based Monkeypox diagnosis model that is computationally efficient, provides adequate model interpretability, incorporates generalization and regularization techniques to reduce overfitting and is ultimately tested on both binary and multiclass datasets.

2. Motivation

The accurate diagnosis of Monkeypox disease plays a pivotal role in its effective treatment and containment. While traditional diagnostic methods have been developed, they often need to be more accurate and require specialized equipment and expertise. In contrast, ML and DL approaches have demonstrated promising results in image-based medical diagnoses, encompassing lung cancer, brain tumor, skin disease, and COVID-19 [9]. However, research on DL-based approaches for Monkeypox disease diagnosis still needs to be expanded and necessitates model interpretability and computational efficiency considerations.

In this study, we aim to develop and evaluate the performance of cutting-edge TL algorithms, specifically Visual Geometry Group (VGG)-16, Residual Network (ResNet)-50, ResNet101, and Vision Transformer (ViT), for Monkeypox disease diagnosis. The evaluation encompasses not only the accuracy of predictions but also the computational complexity and interpretability of the models. The integration of interpretable ML techniques offers valuable insights into the underlying decision-making process of the models and the factors influencing their predictions.

Moreover, we recognize the importance of advanced disease segmentation techniques in comprehensively understanding Monkeypox diseases. To achieve this, we introduce the utilization of Region of Interest (ROI) and Segment Anything (SAM) models alongside threshold-based segmentation methods. These innovative approaches enhance the visual analysis of infected regions, providing valuable perspectives for disease assessment.

The pressing need to assess these algorithms' performance in

Monkeypox disease diagnosis arises from the increasing urgency to develop efficient and accurate diagnostic tools. As the number of Monkeypox cases continues to rise, developing robust diagnostic models capable of swiftly and precisely identifying infected individuals becomes paramount. Leveraging TL and ViT approaches holds the potential to significantly enhance the performance and efficiency of Monkeypox disease diagnosis, making it more accessible to a larger population.

In conclusion, our motivation for this experiment is to propose and evaluate the performance of TL and ViT algorithms for Monkeypox disease diagnosis, considering accuracy, computational efficiency, and interpretability. The study's outcomes are expected to contribute valuable insights into creating robust and efficient diagnostic tools for Monkeypox disease.

Our contributions to the field of Monkeypox disease diagnosis are outlined as follows:

1. Application of TL techniques, evaluating their effectiveness in developing a preliminary Monkeypox patient detection model using three modified CNN models: M-VGG16, M-ResNet50, and M-ResNet101.
2. Investigation and assessment of ViT-based approaches, accompanied by advanced visual representation techniques for model outcome analysis.
3. Validation of findings through LIME post-image analysis, providing comprehensive explanations of model predictions and enhancing interpretability.
4. Integration of FL into our study, specifically for analyzing Monkeypox diseases, due to its significance in safeguarding sensitive medical images across geographically distributed locations.
5. Implementation Region of Interest and threshold based segmentation (ROITS) and Segment Anything (SAM) models for improved disease segmentation and visual understanding.

The remainder of the paper is organized as follows: The methodology of the experiment is described in detail in Section 3. The results of the experiment are presented in Section 5. A brief discussion of the study's findings is provided in Section 6, and the overall conclusions of the research and potential avenues for future research are summarized in Section 7.

3. Methodology

This section outlines the procedure for acquiring and augmenting the data, developing the proposed DL-based model, experimental setup, and evaluating metrics employed to assess the model's performance.

3.1. Data collection

At the early stage of this work, the lack of a standard Monkeypox dataset posed a challenge for researchers and practitioners in developing and testing DL-based models for Monkeypox disease diagnosis. To address this issue, we conducted a prior study in which we developed a dataset for Monkeypox disease diagnosis. The procedure for data collection and elaboration of details can be found in the following arxiv [22] repository and peer-review [9] papers. Table 1 presents a summary

Table 1
Characteristics of the collected dataset [9].

Dataset	Data class	Sample	Total
One	Monkeypox	43	76
	Normal	33	
Two	Monkeypox (upsample data)	587	1754
	Normal (upsample data)	1167	
Three	Monkeypox (mixed data)	866	2524
	Normal (Mixed data)	1658	

of the developed datasets used in the prior study. As shown in the table, dataset 1 comprises only 60 samples, dataset 2 encompasses 1754 samples, and dataset 3 contains 2524 samples.

3.2. Convolutional neural network

The essential architecture of Convolutional Neural Networks (CNNs) in image classification comprises Convolutional (Conv), Pooling, and Dense layers. CNNs are widely used in Deep Learning (DL) for various applications, including image recognition, video analysis, image segmentation, natural language processing (NLP), and object detection [23]. The basic structure of CNNs for image classification is outlined in referenced Fig. 1 [23].

In the CNN architecture, feature extraction is one of the most important parts. This technique involves reducing the dimensionality of vast amounts of data with the purpose of enhancing the efficiency of DL-based models. The CNN architecture consists of multiple layers specifically designed to recognize and extract the most relevant and important features from the input data. To obtain the output feature map, the CNN model utilizes various weighted kernels, W^l , at each layer, l , in the model. This feature extraction process is crucial in preserving the most meaningful information from the input data, which is then passed on to subsequent layers for further processing [24].

$$o^l = W^{lT} a^{(l-1)} + b^l \quad (1)$$

where

$$\begin{aligned} o^l &= \text{feature map} \\ b^l &= \text{bias of the layer} \end{aligned}$$

The pooling layer is utilized for the extraction of the maximum feature values in the following manner:

$$P_m^l = \max_{(p,q)} o^l \quad (2)$$

where (m, n) represents the dimensions of the pooling window.

$$P_m^l = \text{pooling layer}$$

The final layer, also known as the linked layer, can be defined as follows: If layer l takes the input feature map $P_1^{(l-1)}$ with a size of $P_2^{(l-1)} \times P_3^{(l-1)}$ from the previously linked layer $l-1$ then the linked layer can be expressed as:

$$K_i^l = f(Z_i^{(l)}) \quad (3)$$

$$Z_i^l = \sum_{j=1}^{P_1^{(l-1)}} \sum_{r=1}^{P_2^{(l-1)}} \sum_{s=1}^{P_3^{(l-1)}} w_{ij,r,s}^{(l)} (K_j^{(l-1)})_{r,s} \quad (4)$$

where, the weight $w_{ij,r,s}^{(l)}$ is used to define the connection between the i^{th} unit's location (r, s) and the j^{th} feature map in layer $(l-1)$.

In the training phase, it's necessary to use optimization techniques to find the best model. To achieve this, we evaluated two commonly used optimization algorithms: an adaptive learning rate optimization (Adam) algorithm and stochastic gradient descent (Sgd) [23,24].

3.3. VGG16

The VGG network consists of two deep CNN models named VGG16 and VGG19. These models were developed by the Visual Geometry Group and trained on a dataset of one million samples from the ImageNet database. The VGG16 model requires the images of size 224×224 and processes them through a series of Convolutional layers with filters ranging from 64 to 512, using Max pooling with a 2×2 filter and

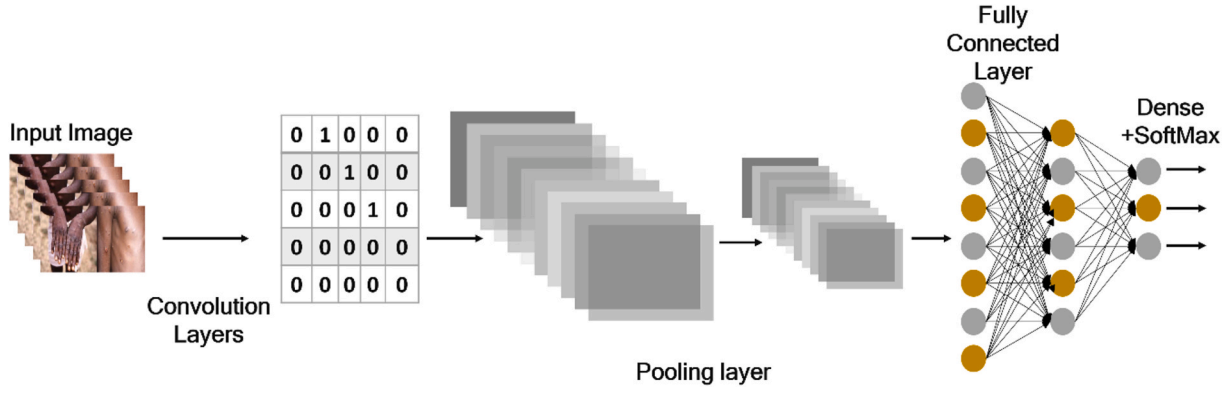


Fig. 1. The basic structure of a CNN designed for image classification.

a stride of 2 and a Rectified Linear Unit (ReLU) activation function. The Pooling layer is then transformed into a 1D vector in the dense layer. The final step involves applying a Softmax activation function to the output layer for categorizing the samples into one of 1000 categories [23].

3.4. ResNet50

ResNet50, a Residual Network model, has 50 layers, including convolutional, max pooling, average pooling, and regular pooling layers. It starts with a 7×7 convolutional layer with 64 kernels and a stride of 2, followed by max pooling with the same stride. It features nine convolutional layers with 64, 64, and 256 kernel filters and another nine layers with 512, 512, and 2048 kernels. The architecture concludes with a 1000-node fully connected layer, an average pooling layer, and a Softmax activation function on the output [25].

3.5. ResNet101

The ResNet101 model is similar in architecture to its counterpart, the ResNet50. The network takes images with a resolution of 224×224 . The main difference between ResNet50 and ResNet101 lies in the fourth block, where ResNet101 has an added three-layer block with 256, 256, and 1024 filters [26].

3.6. ViT

The Vision Transformer (ViT) introduced by Dosovitskiy et al. for image classification treats images as sequences of patches processed by a transformer-based model. This model, pretrained on a large dataset, is fine-tuned for specific tasks. ViT's architecture, as shown in Fig. 2, includes a Transformer Encoder (TE), Multi-Layer Perceptron (MLP) normalizer, Multi-Head Attention Mechanism (MHAM), and Embedded Patches (EP). The TE analyzes sequential data, extracting features, while the MLP normalizes these activations for stable training and enhanced performance. The MHAM focuses on different image parts, enabling finer feature extraction and better understanding of image regions [27].

3.7. Proposed approaches

The proposed modified CNN leverages TL approaches. Fig. 3 depicts the two major sections of the proposed method: the preprocessing of Monkeypox images and the detection of infected individuals based on the modified TL.

3.7.1. Preprocessing of the Monkeypox images

The utilization of ROI patches on image datasets aims to extract ($224 \times 224 \times 3$ uniform dimension) relevant information from the images rather than to process the entire image to enhance both the efficiency and accuracy of the model's predictions. Additionally, the grayscale

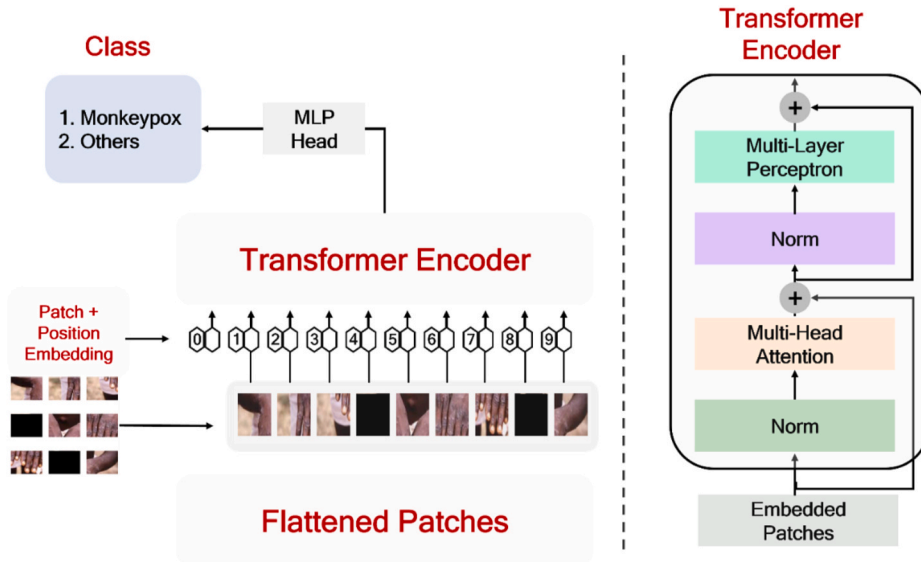


Fig. 2. A diagram of a ViT featuring a Transformer Encoder, Multi-Layer Perceptron normalization, Multi-Head Attention Mechanism, and Embedded Patches for fine-grained feature extraction.

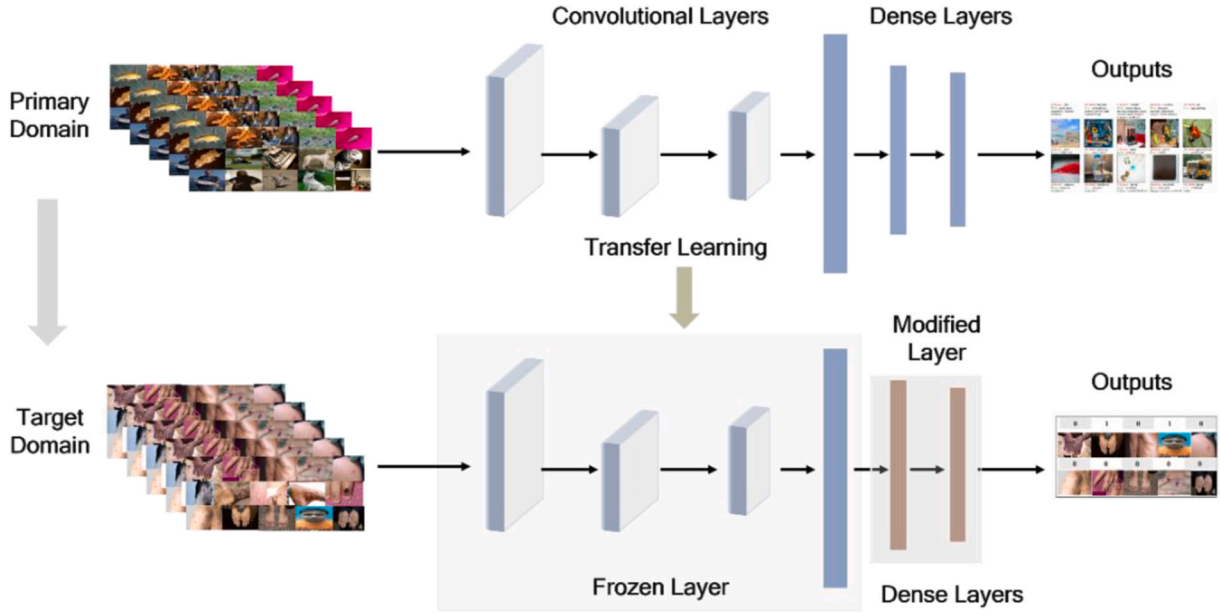


Fig. 3. The framework of the proposed modified TL method.

images were converted into RGB format using functions from the OpenCV library to facilitate compatibility with the input layer of the proposed CNN model. No additional preprocessing was performed, as this was sufficient to reduce computational time complexity.

3.7.2. Proposed CNN architecture

The proposed CNN model comprises three main components: pre-trained architecture, updated layer, and prediction class (partially adopted from Ref. [9]). The pre-trained architecture is leveraged to extract high-dimensional features, which are then incorporated into the modified layer. The model architecture, illustrated in Fig. 3, consists of 16 CNN layers with varying filter sizes and strides [28]. The input layer, which takes 224×224 images, is followed by two 3×3 filter Conv Layers, a Max Pooling Layer, two more Convolutional Layers, and another Max Pooling Layer. The modified layer then flattens the architecture, followed by three Dense and one Dropout Layer.

The proposed model's performance is optimized through parameter tuning, which involves the examination of the batch size, number of epochs, and learning rate inspired by prior research [29,30]. The results of the optimal parameters for all three studies are detailed in Table 2. The data indicates that a learning rate of 0.001 produced an improved performance in all three studies. However, it is noteworthy that the optimal batch size and the number of epochs varied between the studies, which can be attributed to the differing sizes of the datasets used in each study.

Table 2

Optimal parameter identification using the grid search method used in this study.

Study	Parameter	Grid search	Optimal parameters	parameters
One	Number of epochs	30, 35, 40, 50	30	
	Batch size	5, 10, 15, 20	5	
	Learning rate	0.1, 0.01, .001	0.001	
Two	Number of epochs	30, 35, 40, 50	50	
	Batch size	30, 40, 45, 50	30	
	Learning rate	0.1, 0.01, 0.001	0.001	
Three	Number of epochs	30, 35, 40, 50	40	
	Batch size	10, 30, 40, 50	30	
	Learning rate	0.1, 0.01, 0.001	0.001	
Four	Number of epochs	30, 35, 40, 50	50	
	Batch size	10, 30, 40, 50	30	
	Learning rate	0.1, 0.01, 0.001	0.001	

3.8. LIME as explainable AI

The utilization of LIME provides a valuable approach for examining the predictions of a DL model, specifically CNN. LIME helps to shed light on the “black-box” nature of DL models, enabling the evaluation of the underlying decision-making process for a given prediction [31]. According to Ribeiro et al., for a model to be considered locally interpretable, it must approximate well around the predicted sample [32].

LIME uses a formula to generate an optimized model that is locally faithful. Utilizing the following formula, the optimal LIME model for model interpretation is developed [31,32]:

$$(x) =_{g \in G} \mathcal{L}(f, g, \pi_x) + \omega(g) \quad (5)$$

in this equation, g represents the interpretable model that LIME attempts to optimize. The goal is to minimize the loss function $\mathcal{L}(f, g, \pi_x)$, which quantifies the discrepancy between the predictions of the original complex model f and the simpler interpretable model g , within the vicinity of the input sample x . The kernel π_x , typically an exponential kernel weighted by Euclidean distance, prioritizes the fidelity of g to f closer to x . The bandwidth σ of this kernel plays a crucial role in determining the locality of the explanation. A penalty term $\omega(g)$ is included to ensure the simplicity of the interpretable model, thereby preventing overfitting and maintaining the explanation's interpretability.

In our implementation, LIME was applied to analyze the decision-making processes of our CNN models. This involved perturbing the input data around a target prediction and observing the corresponding changes in the output. These minor perturbations were significant enough to provide insights into how the model's predictions change with variations in the input. For each perturbed instance, the output of LIME highlighted the key features and their contributions to the final prediction, thereby elucidating the reasoning behind the model's decision.

Through the application of LIME, we could discern not only the features that led to accurate predictions but also the reasons behind misclassifications. For instance, the analysis revealed instances where the model's attention to specific features was misaligned with the clinical indicators of a condition, leading to an incorrect diagnosis. This has been instrumental in identifying areas for model refinement and enhancing the predictions' overall reliability.

In conclusion, implementing LIME in our study provided a deeper understanding of our CNN models, bridging the gap between high accuracy and interpretability. It underscored the importance of transparency in AI models, especially in critical applications like medical diagnosis, where comprehending the model's reasoning is as crucial as its predictive accuracy.

4. Experiment setup

The experimental setup consisted of a conventional laptop with the Windows 10 operating system specifications, 16 GB of RAM, and an Intel Core i7 processor. Table 3 summarizes the datasets utilized in this Study. As indicated in the table, a total of 76 images were employed for Study One, 1754 images for Study Two, 656 images for Study Three, and 2524 images for Study Four. It is noteworthy to mention that the datasets used for Studies One, Two, and Four consist of binary class data. In contrast, the dataset utilized for Study Three is composed of multiclass data. A typical approach in the ML domain was adopted, where 80% of the sample data was allocated for training and the remaining 20% for testing the model.

Table 4 presents the concise data distribution for Study Four across two servers: "Client One (USA)" and "Client Two (Bangladesh)". Each server has labeled datasets divided into training and test sets. For Client One, there are 235 training instances and 59 test instances, totaling 294. Similarly, there are 235 training instances and 59 test instances for Client Two, totaling 294 instances. These distributions are vital for evaluating and comparing models developed for Study Four, considering the data variability between the two servers and their locations.

For the Client One server0 in the USA, an office-grade laptop running on the Windows 10 operating system, powered by an Intel Core i7-7500U processor and supported by 16 GB RAM, was used. On the other hand, for the Client Two server (located in Bangladesh), a computer system running on Windows 11 Pro, equipped with an Intel® Xeon® E-2186 M Processor, 32 GB DDR4 RAM, and an Nvidia Quadro P1000 GPU, was utilized.

We adopted optimal parameter settings derived from the preceding studies to simplify the experimentation process. Specifically, a batch size of 30 and a learning rate of 0.001 were utilized while keeping the remaining experimental parameters consistent with those used in the previous studies in Table 2.

A representation of sample images from the datasets used in this study is shown in Fig. 4. In this study, the technique of early stopping was employed to prevent overfitting and data leakage and accurately assess the model's performance. Early stopping is a widely used regularization technique in DL, where the evaluation of the validation loss occurs less frequently, and the trained model is saved periodically [33].

Table 3
Allocation of data used for training and testing during this study.

Study	Label	Train set	Test set	Total
Study One	Monkeypox	34	9	43
	Normal	26	7	33
	Total	60	16	76
Study Two	Monkeypox	470	117	587
	Others	933	234	1167
	Total	1403	351	1754
Study Three	Chickenpox	80	20	100
	Measles	64	16	80
	Monkeypox	211	53	264
	Normal	169	43	212
	Total	524	132	656
Study Four	Monkeypox	696	170	866
	Normal	1323	335	1658
	Total	2019	505	2524

Table 4

Data distribution among two servers for Study Four.

Study Four	Label	Train set	Test set	Total
Client One (USA)	Monkeypox	167	42	209
	Normal	68	17	85
Client Two (Bangladesh)	Total Monkeypox	235	59	294
		174	44	218
	Normal	61	15	76
	Total	235	59	294

4.1. Client side experimental setup for FL

In the current study, we incorporated an FL strategy to critically assess the resilience of our proposed model under the conditions of two distinct training environments, employing the mechanism of distributed weight sharing. The primary objective of this investigation is to evaluate the feasibility of utilizing the proposed model as a comprehensive global model that can serve diverse users and organizations without necessitating the disclosure of proprietary data. This aspect is particularly salient in the medical field, where preserving patient confidentiality is of utmost importance.

Fig. 5 presents a schematic representation of the FL framework utilized for training various TL based models, including VGG16, ResNet50, and ResNet101, on two distinct client servers. As depicted, an identical TL-based model is employed to train the Convolutional Neural Network (CNN) models on divergent datasets, with no data sharing occurring between the two clients. Subsequently, these clients share their trained weights, which act as model parameters for the global model. The global model is then assessed on a shared server, with the performance outcomes articulated in Table 1 within the Results section.

4.2. Federated learning approaches

In this study, we implement an FL approach using ResNet101, ResNet50, and VGG16 architectures to train models across multiple clients and a global server.

Our methodology involves a Server-Side Model which operates in three fundamental steps:

1. Initially, the global model is equipped with baseline weights and the training is delegated to the client side.
2. Each client performs local training on their respective datasets. The objective of the clients is to optimize the model by minimizing the categorical cross-entropy loss, a process that involves classifying data into distinct categories. Upon successfully training and identifying the optimal model, the clients relay these model updates or weights to the server.
3. Upon receiving updates from both clients, the server then updates the global model parameters by computing an average weight and generating a new model.

The proposed FL approach involves the utilization of local loss functions (LLF) for each client and a global loss function to achieve collaborative model training across multiple distributed data sources. The LLF for each client process is calculated using the following equations:

$$F_x(w) = \frac{1}{n_x} \sum_{i \in P_k} f_i(w) \quad (6)$$

In Equation (1), (w) represents the local loss function for client x . It is computed as the average of individual loss functions (w) over the data samples i in the local dataset P_k belonging to client x . Here, n_x denotes the total number of data samples in the client x 's dataset. - Global loss function:

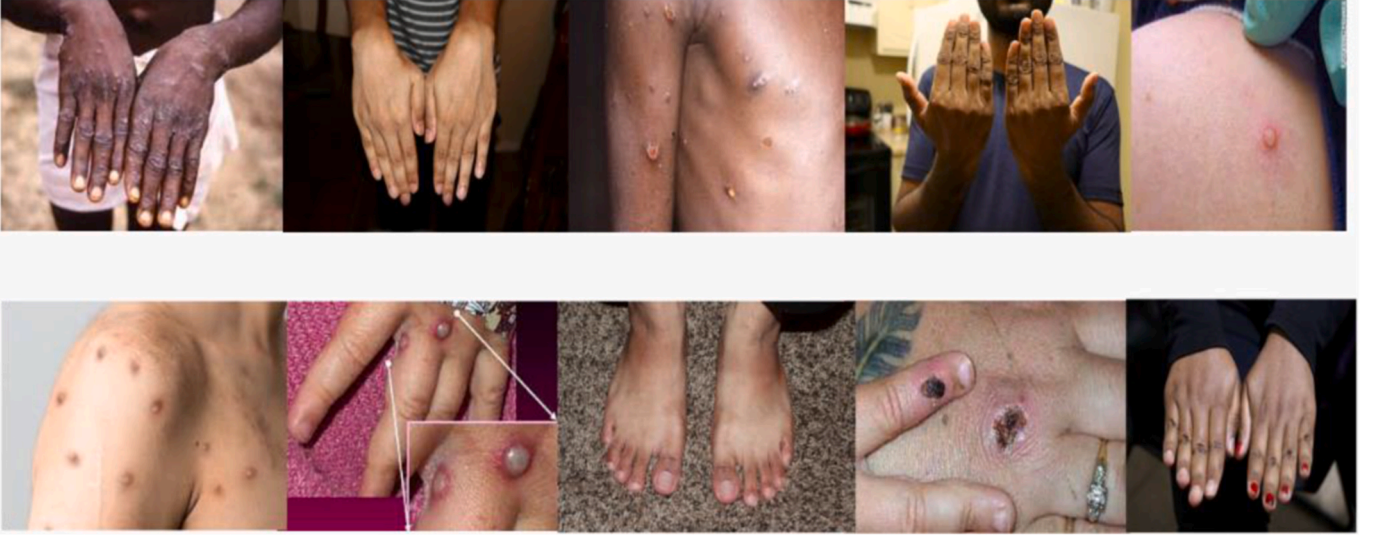


Fig. 4. Representative samples of both Monkeypox and normal images obtained from Ref. [22].

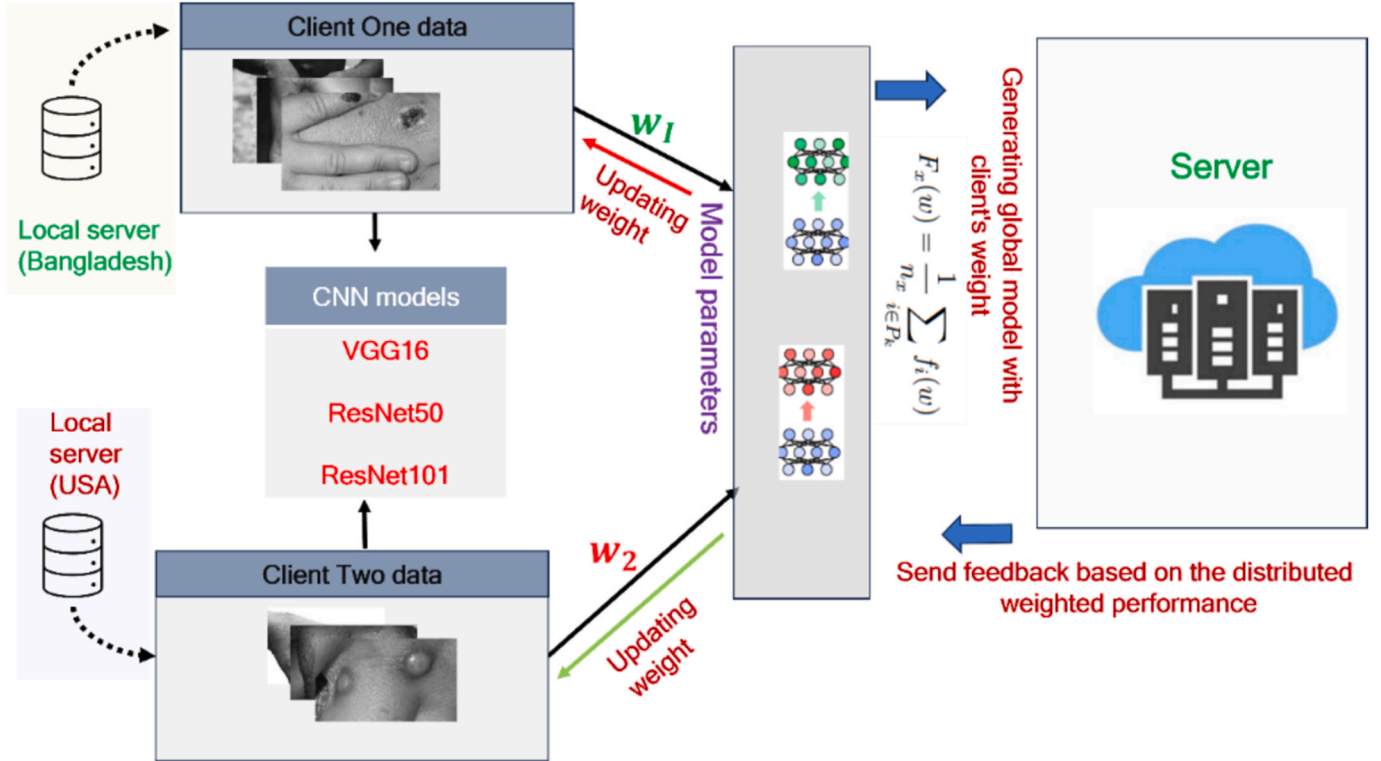


Fig. 5. Privacy-preserving FL approaches for medical data: a distributed framework for model Weight sharing among multiple clients.

$$f(w) = \sum_{x=1}^x \frac{n_x}{n} F_x(w) \quad (7)$$

In Equation (2), (w) represents the global loss function. It is calculated as the weighted sum of the local loss functions (w) across all clients x . The weight of each client's contribution is determined by the scaling factor $\frac{n_x}{n}$, where n_x is the number of data samples in the client x 's dataset, and n is the total number of data samples across all clients.

By the scaling factor derived from each dataset, the weights of the local models are updated, and subsequently, the summation of all client weights is passed on to the global model. This process allows for collaborative model training across multiple distributed data sources

while ensuring that sensitive data remains localized and does not need to be shared centrally [34].

The iterative process of weight updating is repeated for a total of two rounds. It is important to emphasize that the global model does not begin with pre-existing weights but updates its weights after each round by considering the average weights received from the client models. This approach ensures collaborative learning, where the global model effectively incorporates the knowledge and insights derived from diverse client datasets.

4.3. Performance evaluation

The experimental results are evaluated using traditional statistical measures, including accuracy, precision, recall, F1-score, sensitivity, and specificity. The results are presented with a 95% confidence interval, which is in line with previous studies that utilized a small dataset [33,34,35]. In our study, the classification of Monkeypox as true positive (T_p) or true negative (T_n) indicates a correct diagnosis, while false positive (F_p) or false negative (F_n) indicates a misdiagnosis. The explanation of these statistical metrics is provided in detail in Table 5. The notation of x_p and x_n is used to represent positive and negative data samples, respectively, and r_i represents the assigned rating for the i^{th} positive sample in equation (5).

5. Results

Table 6 presents the results of the preliminary computations performed on the train and test sets for the M – VGG16 model, including overall accuracy, precision, recall, F1-score, sensitivity, and specificity scores. The best performance of the model is indicated in bold font. The table demonstrates that the modified M-VGG16 models exhibit consistent performance across Studies One, Two, Three, and Four when trained with either Adam or Sgd optimizer. The highest accuracy on the test set was achieved during Study One, regardless of the optimizer used for training. However, the lowest performance was observed during Study Two when the model was trained with an Sgd optimizer. In Study Four, the model trained with the Adam optimizer maintained high accuracy, precision, recall, F1-score, sensitivity, and specificity scores, whereas the Sgd optimizer showed a considerable variation, with scores ranging from high (1.0 for specificity) to low (0.0 for sensitivity and F1-score).

Table 7 presents the overall performance results of the M-ResNet50 models for Studies One, Two, Three and Four, along with the optimization algorithms used. The table shows that after being trained using either the Adam or Sgd optimizer, the M-ResNet50 models continuously performed poorly during Studies One and Two. In contrast, during Study Three, the proposed M-ResNet50 demonstrated significantly better performance when trained with both the Adam and Sgd optimizers. However, during Study Four, better performance was observed only with the Adam optimizer.

Table 8 presents the results of the experimental evaluations performed on the M-ResNet101 model, utilizing various optimization algorithms in Study One, Two, Three, and Four. The table illustrates that in Study One, the ResNet101 model trained with the Sgd optimization algorithm achieved the highest accuracy, precision, recall, and F1 score. However, in Study Two, the model's performance remained consistent across different optimizers. Contrarily, in Study Three, the best performance was observed for both Adam and Sgd optimizers. Notably, in Study Four, the Adam optimizer again demonstrated superior

performance compared to the Sgd optimizer.

Table 9 presents the results of experimental evaluations of the ViT model utilizing different optimization algorithms in Studies One, Two, Three, and Four. The data indicates that the ViT model trained with the Adam optimization algorithm achieved the highest accuracy, precision, recall, and F1-score levels in Study One, with notable performance in Study Three and Four as well. However, the lowest performance was observed when the ViT model was trained with the Sgd algorithm in Study Two. Notably, in Study Four, the model trained with the Adam optimizer showed improved metrics over the Sgd optimizer, although both displayed enhanced performance compared to earlier studies. These findings indicate that Adam may be a more appropriate optimization algorithm for training ViT models.

Fig. 6 presents the performance of the models on the training and testing sets per epochs during each study. As shown in Fig. 6(a), (b) and (d), the M-VGG16 model with the Adam optimizer demonstrates an improvement in performance until the model starts to overfit after 25 epochs in Study One, Two and Four. In Study Three, the M-ResNet50 model (Fig. 6(c)) shows promising results until it starts to overfit after 20 epochs.

On the other hand, the M-ResNet50 model trained with SGD starts to overfit after only 10 epochs in Study One, as illustrated in Fig. 6(e). In Study Two, the performance of the M-ResNet101 model does not improve and fluctuates; it starts to overfit after 12 epochs, as shown in Fig. 6(f). In Study Three, the ViT model demonstrates the worst performance during the training phase when trained with the Adam optimizer, as depicted in Fig. 6(g). In Study Four, the M-ResNet50 model, trained with SGD, did not overfit but was unable to improve performance, as shown in Fig. 6(h).

To mitigate overfitting, we employed early stopping techniques. Using these techniques, we found that, while the performance of the ViT model did not improve significantly, it did not display any overfitting throughout the training process.

The performance of the best and worst models in terms of the area under the receiver operator characteristic (ROC) curve (AUC) is represented in Fig. 7, with the true positive rate (TPR) and false positive rate (FPR) as the metrics of evaluation. As depicted in Fig. 7(a), (b) and (e), the M – VGG16 model demonstrates the highest level of performance, with AUC scores of 1.0, 0.937, and 0.991 for Studies One, Two, and Four, respectively, on the training set. In Study Three, the M-ResNet50 model trained with the Adam optimizer attains the best performance with an AUC score of 0.90, as shown in Fig. 7(c).

In contrast, the lowest level of performance is observed for the M-ResNet50 with AUC scores of 0.50 both the training and testing sets for Studies One and Four, as shown in Fig. 7(e) and (h) and M-ResNet101 models, with AUC scores of 0.50 on both the training and testing sets for Study Two, respectively, as shown in Fig. 7(f). Additionally, during Study Three, the ViT model demonstrated the least favorable performance with an AUC score of 0.77, as shown in Fig. 7(g).

In Fig. 8, the best and worst models in terms of confusion matrix analysis are presented. In Study One, the M-VGG16 model misclassified four samples. Similarly, M-VGG16 misclassified 80 samples in Study Two, as depicted in Fig. 8(a) and (b). Additionally, in Study Three, the M-ResNet101 model misclassified 20 out of a total of 131 samples, as shown in Fig. 8(c), whereas in Study Four, M-VGG16 misclassified 117 samples, as depicted in Fig. 8(d). On the other hand, the M-ResNet50, M-ResNet101, and ViT models demonstrated lower performance in terms of misclassification rates. Specifically, M-ResNet50 models misclassified 7 out of 7 (e) and 170 out of 170 (h) during Studies One and Four, respectively; M-ResNet101 misclassified 117 out of 117 (f) during Study Two; and ViT models misclassified 40 out of 132 samples (g) during Study Three.

In Table 10, we have presented a detailed summary of the total process time for each module that was deployed throughout the research. To ensure that the training process is robust and efficient, an early stop mechanism was employed to prevent data leakage and

Table 5

Performance metrics used to evaluate the model's performance during the study.

Performance measure	Formula	Equation
Accuracy	$\frac{Tp + Tn}{Tp + Tn + Fp + Fn}$	(6)
Precision	$\frac{Tp}{Tp + Fp}$	(7)
Recall	$\frac{Tn}{Tn + Fp}$	(8)
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	(9)
Sensitivity	$\frac{Tp}{Tp + Fn}$	(10)
Specificity	$\frac{Tn}{Tn + Fp}$	(11)
Area under the curve (AUC)	$\frac{\sum r_i(x_p) - x_p((x_p + 1)/2)}{x_p + x_n}$	(12)

Table 6

The performance of the M-VGG16 model on the dataset used in this study, with a confidence interval of $\alpha = 0.05$. A_{tr} -train accuracy, A_{ts} -test accuracy, Pr_{tr} -train precision, Pr_{ts} -test precision, Rc_{tr} -train recall, Rc_{ts} -test recall, F_{tr} -train F1-score, F_{ts} -test F1-score, Sn_{tr} -train sensitivity, Sn_{ts} -test sensitivity, Sp_{tr} -train specificity, Sp_{ts} -test specificity.

Study	Optimizer	A_{tr}	A_{ts}	Pr_{tr}	Pr_{ts}	Rc_{tr}	Rc_{ts}	F_{tr}	F_{ts}	Sn_{tr}	Sn_{ts}	Sp_{tr}	Sp_{ts}
One	Adam	98 ±	88 ±	0.98 ±	0.90 ±	0.98 ±	0.88 ±	0.98 ±	0.87 ±	1	1	0.96 ±	0.71 ±
		1.6%	7.591%	0.016	0.069	0.016	0.076	0.016	0.079			0.023	0.118
		98 ±	88 ±	0.98 ±	0.90 ±	0.98 ±	0.88 ±	0.98 ±	0.87 ±	1	1	0.961 ±	0.714 ±
	Sgd	1.6%	7.591%	0.016%	0.069	0.016	0.076	0.016	0.079			0.022	0.117
		84 ±	76 ±	0.84 ±	0.76 ±	0.84 ±	0.76 ±	0.84 ±	0.76 ±	0.73 ±	0.55 ±	0.89 ±	0.86 ±
		0.936%	2.292%	0.009	0.023	0.009	0.023	0.009	0.023	0.012	0.031	0.008	0.018
Two	Adam	84 ±	75 ±	0.84 ±	0.73 ±	0.84 ±	0.75 ±	0.84 ±	0.75 ±	0.75 ±	0.55 ±	0.89 ±	0.86 ±
		0.936%	2.339%	0.009	0.024	0.009	0.023	0.009	0.023	0.012	0.031	0.008	0.018
		99 ±	80 ±	1	0.8 ±	1	0.69 ±	1	0.72 ±	0.99 ±	0.92 ±	0.99 ±	0.68 ±
	Sgd	0.04%	2.8%	0.028	0.035	0.035	0.033	0.004	0.018	0.004	0.035		
		99 ±	81 ±	0.99 ±	0.82 ±	0.99 ±	0.81 ±	0.99 ±	0.8 ±	0.99 ±	0.92 ±	0.98 ±	0.69 ±
		0.04%	2.7%	0.004	0.026	0.004	0.027	0.004	0.028	0.004	0.018	0.006	0.035
Three	Adam	99 ±	77 ±	0.99 ±	0.70 ±	0.99 ±	0.74 ±	0.99 ±	0.74 ±	0.99 ±	0.65 ±	0.98 ±	0.82 ±
		0.43%	3.67%	0.004	0.038	0.004	0.038	0.004	0.038	0.004	0.042	0.005	0.033
		66.00 ±	66.00 ±	0.330 ±	0.330 ±	0.500 ±	0.500 ±	0.400 ±	0.400 ±	0.00	0.00	1.00	1.00
	Sgd	2.07%	4.13%	0.021	0.041	0.022	0.044	0.021	0.043				
Four	Adam												
	Sgd												

Table 7

The performance of the M-ResNet50 model on the dataset used in this study, with a confidence interval of $\alpha = 0.05$. A_{tr} -train accuracy, A_{ts} -test accuracy, Pr_{tr} -train precision, Pr_{ts} -test precision, Rc_{tr} -train recall, Rc_{ts} -test recall, F_{tr} -train F1-score, F_{ts} -test F1-score, Sn_{tr} -train sensitivity, Sn_{ts} -test sensitivity, Sp_{tr} -train specificity, Sp_{ts} -test specificity.

Study	Optimizer	A_{tr}	A_{ts}	Pr_{tr}	Pr_{ts}	Rc_{tr}	Rc_{ts}	F_{tr}	F_{ts}	Sn_{tr}	Sn_{ts}	Sp_{tr}	Sp_{ts}
One	Adam	57 ±	56 ±	0.28 ±	0.28 ±	0.50 ±	0.50 ±	0.36 ±	0.36 ±	1	1	0	0
		7.420%	14.53%	0.096	0.186	0.080	0.155	0.091	0.175				
		57 ±	56 ±	0.28 ±	0.28 ±	0.50 ±	0.50 ±	0.36 ±	0.36 ±	1	1	0	0
	Sgd	7.420%	14.53%	0.096	0.186	0.080	0.155	0.091	0.175				
		67 ±	67 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.4 ±	0.4 ±	0	0	1	1
		1.344%	2.68%	0.019	0.038	0.017	0.033	0.018	0.036				
Two	Adam	67 ±	67 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.4 ±	0.4 ±	0	0	1	1
		1.344%	2.68%	0.019	0.038	0.017	0.033	0.018	0.036				
		99 ±	89 ±	1	0.89 ±	1	0.89 ±	1	0.88 ±	0.99 ±	0.96 ±	0.99 ±	0.83 ±
	Sgd	0.4%	2.1%	0.021	0.021	0.021	0.021	0.022	0.022	0.004	0.012	0.004	0.026
		0.99 ±	87 ±	1	0.87 ±	1	0.87 ±	1	0.86 ±	0.99 ±	0.95 ±	0.99 ±	0.79 ±
		0.4%	2.2%	0.022	0.022	0.022	0.022	0.023	0.023	0.004	0.014	0.004	0.029
Three	Adam	79 ±	69 ±	0.78 ±	0.64 ±	0.74 ±	0.62 ±	0.75 ±	0.62 ±	0.58 ±	0.65 ±	0.89 ±	0.82 ±
		1.78%	4.03%	0.018	0.042	0.019	0.042	0.019	0.042	0.021	0.042	0.013	0.033
		66 ±	66 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.40 ±	0.40 ±	0.00	0.00	1.00	1.00
	Sgd	2.07%	4.13%	0.021	0.041	0.022	0.044	0.021	0.043				
Four	Adam												
	Sgd												

Table 8

The performance of the M-ResNet101 model on the dataset used in this study, with a confidence interval of $\alpha = 0.05$. A_{tr} -train accuracy, A_{ts} -test accuracy, Pr_{tr} -train precision, $P r_{ts}$ -test precision, Rc_{tr} -train recall, Rc_{ts} -test recall, F_{tr} -train F1-score, F_{ts} -test F1-score, Sn_{tr} -train sensitivity, Sn_{ts} -test sensitivity, Sp_{tr} -train specificity, Sp_{ts} -test specificity.

Study	Optimizer	A_{tr}	A_{ts}	Pr_{tr}	Pr_{ts}	Rc_{tr}	Rc_{ts}	F_{tr}	F_{ts}	Sn_{tr}	Sn_{ts}	Sp_{tr}	Sp_{ts}
One	Adam	57 ±	56 ±	0.28 ±	0.28 ±	0.5 ±	0.5 ±	0.36 ±	0.36 ±	1	1	0	0
		7.420%	14.53%	0.096	0.186	0.08	0.155	0.091	0.175				
		60 ±	56 ±	0.79 ±	0.28 ±	0.54 ±	0.5 ±	0.44 ±	0.36 ±	1	1	0.076 ±	0
	Sgd	7.157%	14.53%	0.052	0.186	0.077	0.155	0.085	0.175			0.109	
		67 ±	67 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.4 ±	0.4 ±	0	0	1	1
		1.344%	2.68%	0.019	0.038	0.017	0.033	0.018	0.036				
Two	Adam	67 ±	67 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.4 ±	0.4 ±	0	0	1	1
		1.344%	2.68%	0.019	0.038	0.017	0.033	0.018	0.036				
		99 ±	84 ±	1	0.83 ±	1	0.84 ±	1	0.82 ±	0.99 ±	0.93 ±	0.99 ±	0.74 ±
	Sgd	0.4%	2.5%	0.026	0.025	0.025	0.025	0.026	0.026	0.004	0.016	0.004	0.032
		99 ±	84 ±	1	0.83 ±	1	0.84 ±	1	0.83 ±	0.99 ±	0.94 ±	0.99 ±	0.75 ±
		0.4%	2.5%	0.026	0.025	0.025	0.025	0.026	0.026	0.004	0.015	0.004	0.031
Three	Adam	81 ±	70 ±	0.81 ±	0.66 ±	0.76 ±	0.62 ±	0.78 ±	0.62 ±	0.59 ±	0.37 ±	0.93 ±	0.86 ±
		1.71%	4.00%	0.017	0.041	0.019	0.042	0.018	0.042	0.021	0.042	0.011	0.029
		66 ±	66 ±	0.33 ±	0.33 ±	0.50 ±	0.50 ±	0.40 ±	0.40 ±	0.00	0.00	1.00	1.00
	Sgd	2.07%	4.13%	0.021	0.041	0.022	0.044	0.021	0.043				
Four	Adam												
	Sgd												

overfitting issues. This mechanism was set to a patience level of 5, which means that if the performance of the validation loss did not improve after five iterations, the model would automatically stop training to avoid further overfitting and computational issues. To evaluate the

processing time, we computed the time per epoch as each algorithm and optimizer terminated at a different epoch. In Study One, the M-VGG16 model trained using the Adam optimizer required the least time to train, with a processing time of 0.9 seconds. In Study Two, the lowest

Table 9

The performance of the ViT model on the dataset used in this study, with a confidence interval of $\alpha = 0.05$. A_{tr} —train accuracy, A_{is} —test accuracy, Pr_{tr} —train precision, Pr_{is} —test precision, Rc_{tr} —train recall, Rc_{is} —test recall, F_{tr} —train F1-score, F_{is} —test F1-score, Sn_{tr} —train sensitivity, Sn_{is} —test sensitivity, Sp_{tr} —train specificity, Sp_{is} —test specificity.

Study	Optimizer	A_{tr}	A_{is}	Pr_{tr}	Pr_{is}	Rc_{tr}	Rc_{is}	F_{tr}	F_{is}	Sn_{tr}	Sn_{is}	Sp_{tr}	Sp_{is}
One	Adam	90 $\pm$ 3.6%	75 $\pm$ 11.0%	0.9 $\pm$ 0.036	0.77 $\pm$ 0.105	0.9 $\pm$ 0.036	0.75 $\pm$ 0.110	0.9 $\pm$ 0.036	0.75 $\pm$ 0.110	0.91 $\pm$ 0.034	0.67 $\pm$ 0.126	0.88 $\pm$ 0.039	0.85 $\pm$ 0.085
	Sgd	85 $\pm$ 4.4%	75 $\pm$ 11.0%	0.85 $\pm$ 0.044	0.77 $\pm$ 0.105	0.85 $\pm$ 0.044	0.75 $\pm$ 0.110	0.85 $\pm$ 0.044	0.75 $\pm$ 0.110	0.91 $\pm$ 0.034	0.67 $\pm$ 0.126	0.77 $\pm$ 0.054	0.85 $\pm$ 0.085
Two	Adam	70 $\pm$ 1.3%	67 $\pm$ 2.7%	0.68 $\pm$ 0.013	0.64 $\pm$ 0.028	0.7 $\pm$ 0.013	0.67 $\pm$ 0.027	0.67 $\pm$ 0.013	0.63 $\pm$ 0.028	0.33 $\pm$ 0.019	0.22 $\pm$ 0.041	0.89 $\pm$ 0.008	0.89 $\pm$ 0.016
	Sgd	68 $\pm$ 1.3%	67 $\pm$ 2.7%	0.65 $\pm$ 0.014	0.63 $\pm$ 0.028	0.68 $\pm$ 0.013	0.67 $\pm$ 0.027	0.63 $\pm$ 0.014	0.62 $\pm$ 0.029	0.23 $\pm$ 0.021	0.19 $\pm$ 0.042	0.91 $\pm$ 0.007	0.9 $\pm$ 0.015
Three	Adam	98 $\pm$ 0.5%	70 $\pm$ 4.2%	0.98 $\pm$ 0.005	0.69 $\pm$ 0.042	0.98 $\pm$ 0.005	0.7 $\pm$ 0.042	0.98 $\pm$ 0.005	0.69 $\pm$ 0.042	1	0.64 $\pm$ 0.046	1	0.89 $\pm$ 0.025
	Sgd	71 $\pm$ 2.1%	73 $\pm$ 4.0%	0.74 $\pm$ 0.020	0.76 $\pm$ 0.037	0.71 $\pm$ 0.021	0.73 $\pm$ 0.040	0.72 $\pm$ 0.020	0.74 $\pm$ 0.039	0.98 $\pm$ 0.005	0.72 $\pm$ 0.040	0.74 $\pm$ 0.020	0.89 $\pm$ 0.025
Four	Adam	91 $\pm$ 1.25%	76 $\pm$ 3.72%	0.90 $\pm$ 0.013	0.73 $\pm$ 0.039	0.90 $\pm$ 0.013	0.73 $\pm$ 0.039	0.90 $\pm$ 0.013	0.73 $\pm$ 0.039	0.88 $\pm$ 0.014	0.63 $\pm$ 0.042	0.92 $\pm$ 0.011	0.81 $\pm$ 0.034
	Sgd	78 $\pm$ 1.81%	71 $\pm$ 3.96%	0.76 $\pm$ 0.019	0.67 $\pm$ 0.041	0.75 $\pm$ 0.019	0.67 $\pm$ 0.041	0.75 $\pm$ 0.019	0.67 $\pm$ 0.041	0.63 $\pm$ 0.021	0.67 $\pm$ 0.041	0.85 $\pm$ 0.015	0.67 $\pm$ 0.041

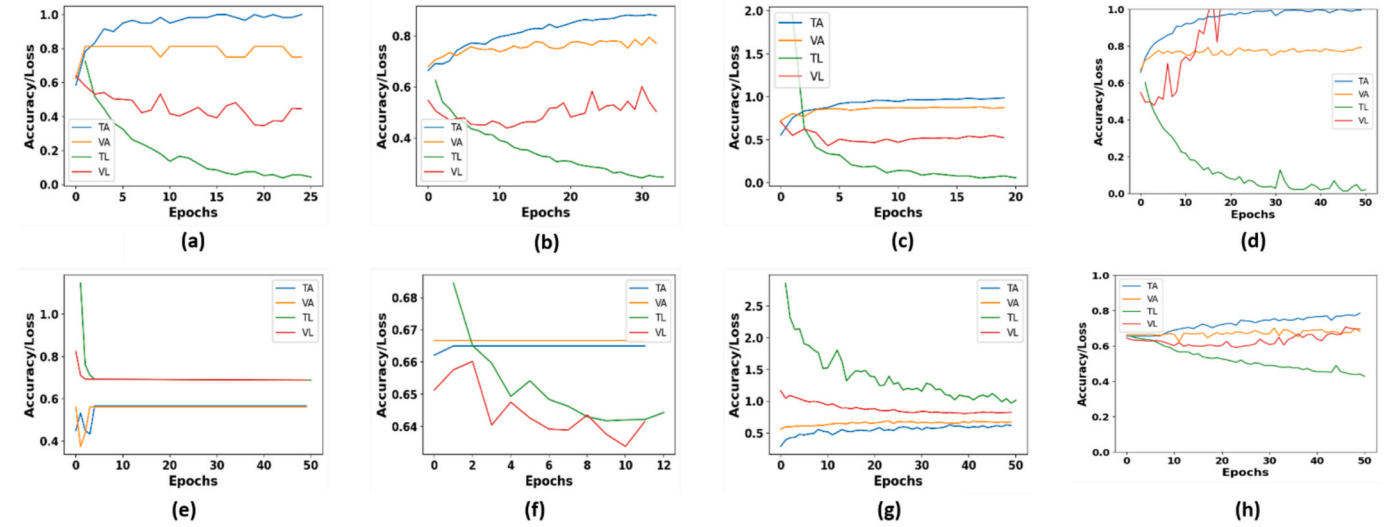


Fig. 6. Models accuracy and loss during each epoch for Study One, Two, Three, and Four. Studies One, Two, and Four observed the best performance for M-VGG16 (a, b and d) with the Adam optimizer. During Study Three, Adam optimizer observed the best performance for (c) M-ResNet50. The worst performance was observed during Study One and Four for (e) and (h) M-ResNet50 with Sgd optimizer, Study Two for (f) ResNet101, and Study Three for (g) ViT with Adam optimizer; TA—train accuracy, VA—validation accuracy, TL—train loss, VS—validation loss.

processing time was observed for the ViT model trained with the Sgd optimizer, with a time of 1.22 seconds. Similarly, for Study Three, the optimal computation time was obtained for the ResNet50 model trained with an Sgd optimizer (0.804 s).

Additionally, in Study Four, we observed that the M-ResNet50 model trained with the Sgd optimizer had a processing time of 20.39 seconds per epoch, while the M-ResNet101 model with the Adam optimizer took 19.60 seconds per epoch. The ViT model in Study Four, trained with both Adam and Sgd optimizers, also showed notable processing times of 5.85 and 4.98 seconds per epoch, respectively.

These findings demonstrate the variability in processing times across different models and optimizers, emphasizing the importance of choosing the right combination for efficient training.

5.1. FL model performance

Table 11 presents the comparative evaluation of test accuracy for the ResNet101, ResNet50, and VGG16 models across two communication rounds. The performance of each model was assessed after the initial training round (communication Round 0) and the subsequent fine-

tuning (communication Round 1) in Bangladesh, the USA, and at the server level. The VGG16 model exhibited superior performance consistently across all three locations. By Communication Round 1, the VGG16 model achieved perfect accuracy in Bangladesh and the USA and an admirable 97.561% on the server. In contrast, the ResNet models demonstrated lower accuracy. Still, the ResNet101 model in Bangladesh displayed noticeable progress between the two communication rounds, with accuracy rising from 72.97% to 86.486%. This data underscores the robust performance of the VGG16 model in the FL approach, demonstrating its potential for reliable application across varied locations and datasets.

Fig. 9 displays the training loss for the best pre-trained model, VGG16, which loss significantly decreases during each epoch. The training loss significantly drops during epoch 2.

6. Discussions

The performance of TL-based models (M-VGG16, M – ResNet50, and M-ResNet101) and a ViT model was observed for image classification in the diagnosis of Monkey-pox disease using various statistical

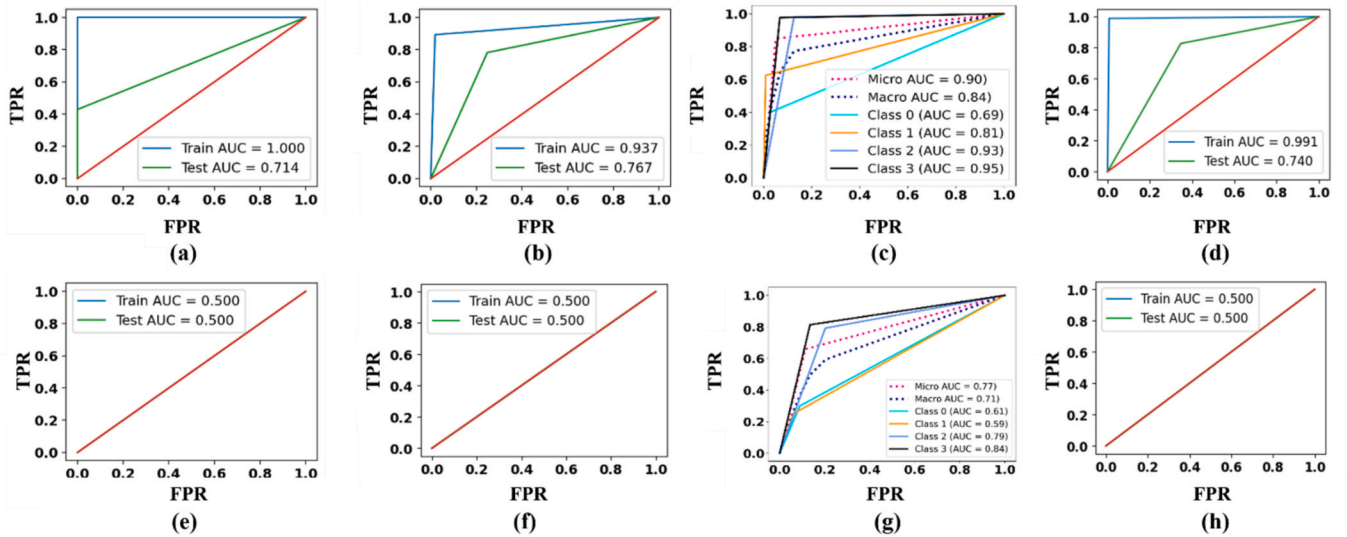


Fig. 7. Best and worst AUC-ROC performances for Studies One, Two, Three, and Four are presented. In Studies One, Two, and Four, the best results were obtained with M-VGG16, as shown in (a), (b), and (d), respectively. In Study Three, the best performance was achieved by M-ResNet50 using Adam optimization, as depicted in (c). Conversely, the worst results were identified for M-ResNet50 with SGD in Studies One and Four, as shown in (e) and (h), for M-ResNet101 in Study Two, as indicated in (f), and for ViT with Adam optimization in Study Three, as illustrated in (g); TPR—true positive rate, FPR—false positive rate.

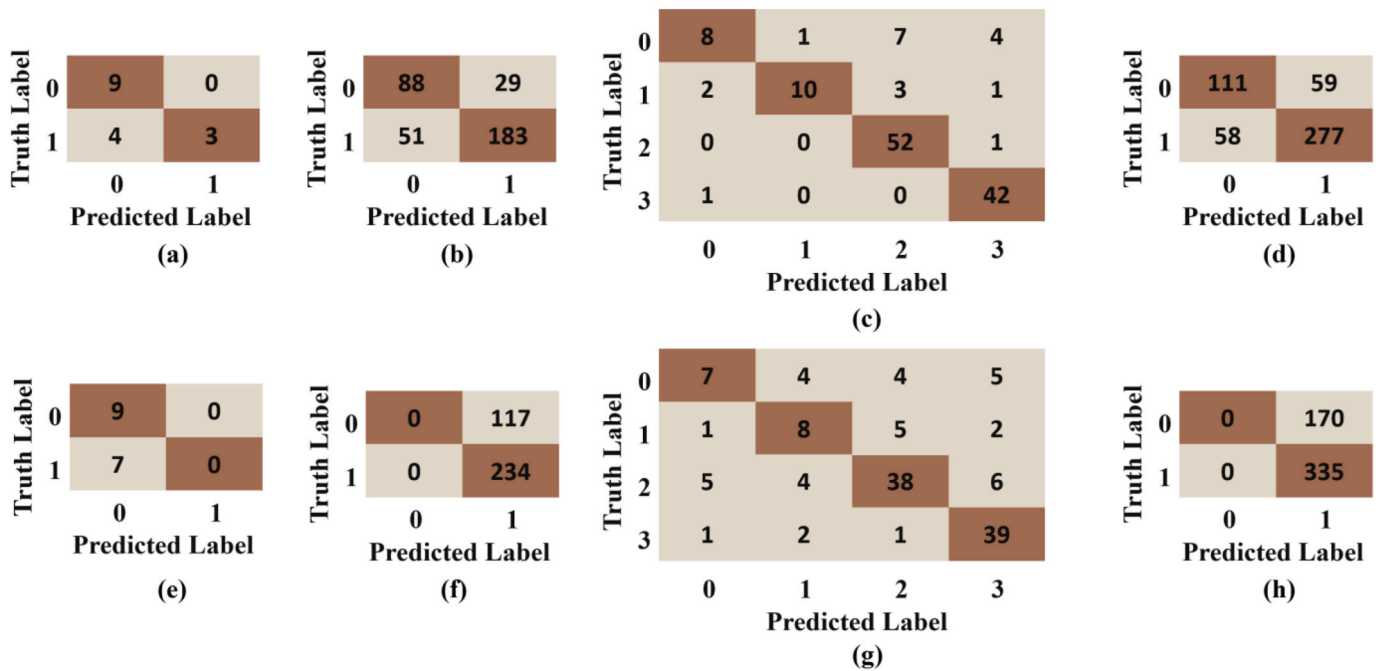


Fig. 8. Confusion matrices for the proposed model applied to Study One, Two, Three and Four. The best performance was observed with M-VGG16 during Studies (a) One, (b) Two and (d) Four and for M-ResNet50 during Study (c) Three. Contrarily, the lowest performance was observed with M-ResNet50 during Studies (e) One and (h) four, M-ResNet101, and ViT during Studies (f) Two, and (g) Three.

measurements. However, there is currently a lack of research utilizing ViT approaches (non-CNN) in this field, which hinders direct comparison with existing reference literature. Despite this, several reference studies have implemented TL approaches for detecting patients with Monkeypox disease, providing the opportunity for a higher level of assessment to evaluate the performance of our proposed models.

Table 12 presents a comprehensive comparison of the performance of several TL algorithms that have been proposed in recent research for the diagnosis of Monkeypox disease. The table compares the results of different algorithms reported by various researchers, such as Islam et al., who used ShuffleNet-V2 to achieve a maximum F1-score of 0.67 and a

precision of 0.79 [36]. Similarly, Ali et al. used ResNet50 and reported a maximum precision of 0.85 and recall of 0.83 [37]. In contrast, our investigation found that the updated VGG16 models using Adam optimization methods obtained a maximum precision of 0.90, recall of 0.88, and F1-Score of 0.87. Thus, based on Study One, we can infer that our proposed model outperforms the existing approaches. Additionally, we employed early stopping approaches to prevent overfitting, which may have resulted in slightly lower performance in Study Two, but it is still comparable to the existing recommended models. Furthermore, our study observed an accuracy of almost 89% using ResNet50 with Sgd optimizer, which outperforms most of the reference literature. Unlike

Table 10

Total computation time for each module used during the training phase of the study.

Model	Study	Optimizer	Early stops (epochs)	Process time (seconds)	Process time/epochs (seconds)
M-VGG16	One	Adam	50	44.51	0.9
		Sgd	7	7.01	1.001
	Two	Adam	54	875.63	16.22
		Sgd	11	194.69	17.7
	Three	Adam	11	17.39	1.58
		Sgd	30	27.21	0.91
	Four	Adam	50	1117.94	22.35
		Sgd	50	998.98	19.97
M-ResNet50	One	Adam	18	21.17	1.2
		Sgd	6	10.09	1.7
	Two	Adam	34	529.22	15.6
		Sgd	10	157.74	15.8
	Three	Adam	19	13.73	0.72
		Sgd	21	16.90	0.804
	Four	Adam	50	1294.88	25.89
		Sgd	50	1019.97	20.39
M-ResNet101	One	Adam	17	25.95	1.53
		Sgd	29	45.52	1.57
	Two	Adam	16	277.22	17.33
		Sgd	12	211.89	17.66
	Three	Adam	18	20.80	1.15
		Sgd	6	11.8	1.97
	Four	Adam	50	980.07	19.60
		Sgd	50	1024.86	20.49
ViT	One	Adam	10	10.51	1.05
		Sgd	10	10.67	1.07
	Two	Adam	40	48.63	1.22
		Sgd	40	79.3	1.98
	Three	Adam	50	90.91	1.82
		Sgd	50	85.55	1.71
	Four	Adam	50	292.62	5.85
		Sgd	50	249.05	4.98

Table 11

Mean Accuracy with 95% confidence Interval for Models on Client One, Client Two, and Servers.

Model	Round	Client 1	Client 2	Server
ResNet101	0	72.97 $\pm$ 6.82	81.081 $\pm$ 9.81	62.195 $\pm$ 13.39
	1	86.486 $\pm$ 4.74	81.081 $\pm$ 9.54	65.854 $\pm$ 14.13
ResNet50	0	83.748 $\pm$ 5.38	78.378 $\pm$ 6.62	62.195 $\pm$ 13.39
	1	81.081 $\pm$ 7.83	76.676 $\pm$ 8.02	62.195 $\pm$ 13.39
VGG16	0	100 $\pm$ 0	91.892 $\pm$ 6.53	93.902 $\pm$ 3.70
	1	100 $\pm$ 0	100 $\pm$ 0	97.561 $\pm$ 1.67

previous studies, we also addressed the issue of overfitting, which is an important consideration in developing robust models. As a result, the performance of our model is more reliable and viable compared to the

existing literature.

Fig. 10 demonstrates the principal features identified by our proposed models by applying LIME. LIME plays a pivotal role in clarifying the decision-making processes of ML models, especially in the context of complex and non-transparent models such as deep neural networks. This technique functions by locally approximating the ML model using an interpretable surrogate, typically a linear model, and subsequently identifying the features that significantly influence the model's predictions.

The implementation of LIME in our study includes generating perturbations in the input data, followed by analyzing the resultant variations in the model's output. This methodology facilitated a deeper comprehension of the model's behavior near each input instance. LIME produced a distinct explanation for every perturbed instance,

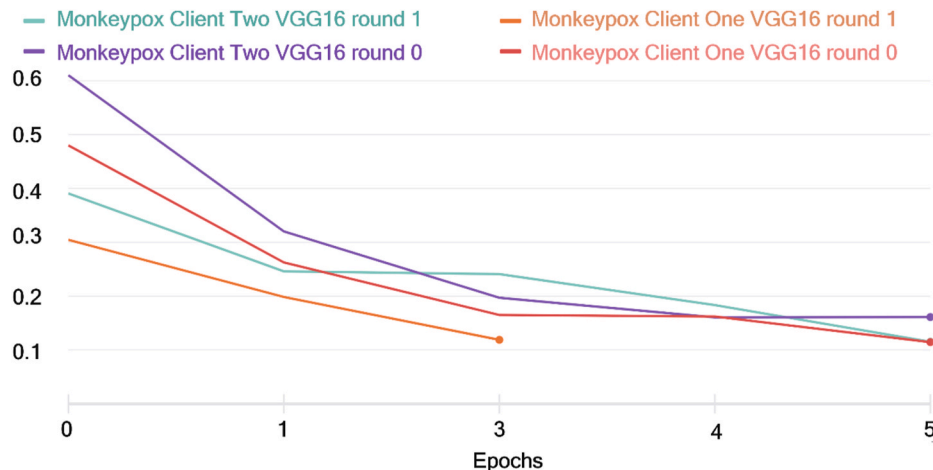
**Fig. 9.** Training loss of pre-trained VGG16 model for client 1, client 2, and the global server.

Table 12
Comparison of proposed and existing DL-based Monkeypox diagnosis model.

Reference	Method	Dataset Size	Accuracy	Precision	Recall	F1-Score
[38]	Xception	572	83.89%	–	–	–
[37]	VGG16	228	81.48 ± 6.87%	0.85 ± 0.08	0.81 ± 0.05	0.83 ± 0.06
	InceptionV3		74.07 ± 3.78%	0.74 ± 0.02	0.81 ± 0.07	0.78 ± 0.04
	Ensemble		79.26 ± 1.05%	0.84 ± 0.05	0.79 ± 0.07	0.81 ± 0.02
	ResNet50		82.96 ± 4.57%	0.87 ± 0.07	0.83 ± 0.02	0.84 ± 0.03
[36]	Inception-V3	804	71%	0.71	0.53	0.61
	ResNet50		72%	0.59	0.51	0.55
	ShuffleNet-V2		79%	0.79	0.58	0.67
	Our best model on the Test set					
Study One	M-VGG16	76	88 ± 7.6%	0.90 ± 0.069	0.88 ± 0.076	0.87 ± 0.079
Study Two		1754	76 ± 2.3%	0.77 ± 0.022	0.77 ± 0.022	0.77 ± 0.022
Study Four		2524	77 ± 3.6%	0.74 ± 0.038	0.74 ± 0.038	0.74 ± 0.038
Study Three	M-ResNet50	656	89 ± 2.1%	0.83 ± 0.026	0.84 ± 0.025	0.82 ± 0.026

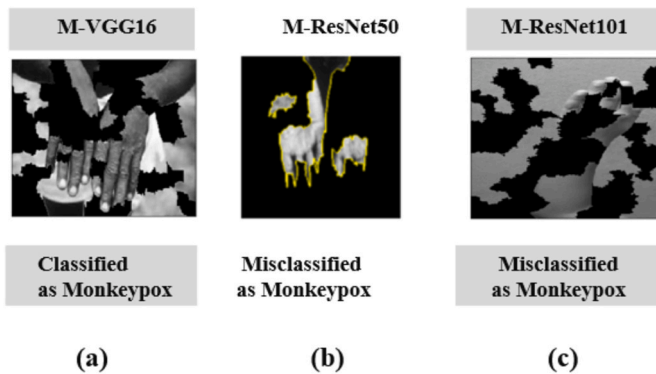


Fig. 10. Top features that enabled the classification and misclassification between Monkeypox and the normal image.

illuminating the rationale behind the model's specific predictions.

As depicted in Fig. 10(a), an image exhibiting symptoms of Monkeypox was accurately classified as Monkeypox by the M-VGG16 model. This accuracy is attributable to the LIME algorithm, which underscored particular features and regions within the image, leading to this precise classification. The figure indicates that these features are the critical elements that the model correlates with Monkeypox symptoms.

Moreover, Fig. 10(b) and (c) offer insights into the reasons behind the misclassification of normal images as those manifesting Monkeypox symptoms by the M-ResNet50 and M-ResNet101 models. The interpretation provided by LIME reveals that these models erroneously emphasized certain non-pathognomonic features, resulting in inaccurate classifications. Such misclassifications are linked to the models' focus on irrelevant or deceptive features within the images, a phenomenon vividly illustrated in these figures.

Therefore, applying LIME has yielded valuable insights into the interpretative dimensions of our models. It has not only confirmed the models' proficiency in accurately identifying symptoms of disease but also pinpointed areas necessitating improvement in instances of misclassification. This degree of interpretability is indispensable in medical applications, where comprehending the justification behind a model's prediction is as imperative as the prediction itself.

Fig. 11 illustrates the segmentation using SAM. The figure reveals that, despite the rapidity of SAM in providing segmentations, it encountered limitations in accurately delineating certain infected areas. Some regions were left unsegmented, while in specific cases, entire areas were erroneously segmented, as denoted by the red arrows. Consequently, despite the SAM-based approach's superiority, accurately segmenting Monkeypox infected areas may prove challenging, as it can be achieved more effectively through a straightforward threshold-based segmentation technique, as demonstrated in Fig. 12.

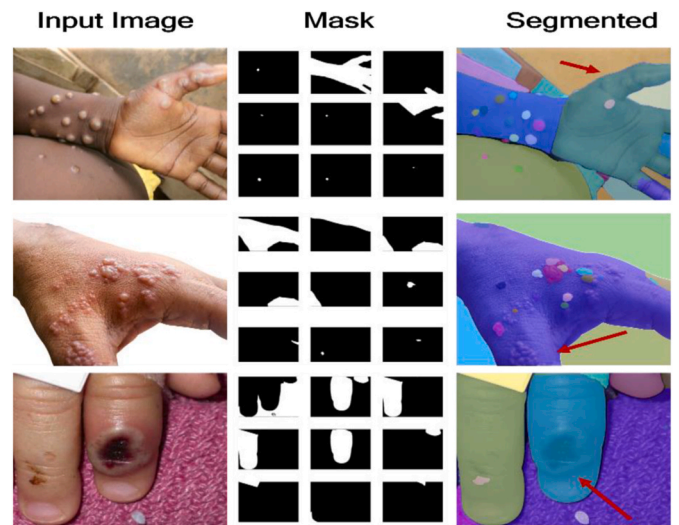


Fig. 11. SAM-based segmentation of three distinct Monkeypox images, accompanied by their respective masks and corresponding segmented areas.

Fig. 13 presents the visualization of the attention weights generated by a Transformer block for attention heads 1, 3, and 9 in three representative Monkeypox images. The attention weights, as depicted in the heatmap format, indicate the relative importance of various regions within the image with respect to the query. As the key and query are derived from the same image, the attention weights effectively highlight the most relevant regions for the task at hand. The attention heads 1, 3, and 9 are specifically highlighted in the figure to demonstrate the attention patterns learned by the model during the image analysis process.

The self-attention mechanism in ViT allows it to gather information from all parts of the image, even in the early layers of the network. In Fig. 14, we investigate how much the model utilizes this capability. Specifically, we calculate the average distance in the image from which the model is integrating information, as indicated by the attention weights (as shown on the right side of Fig. 14). The left side of the diagram illustrates (Fig. 14) the filters used in the initial linear embedding of the RGB values in the ViT model. This process is important as it converts the raw pixel values into a more meaningful and compact representation that the model can use to understand the image. The right side of the diagram shows the size of the area attended to by each head and the network depth. The attention mechanism used in the ViT model allows the model to focus on different regions of the image, which is important for understanding the image and making accurate predictions. Each dot on the right side of the diagram represents the average attention distance across all images for one of the 16 heads at a specific

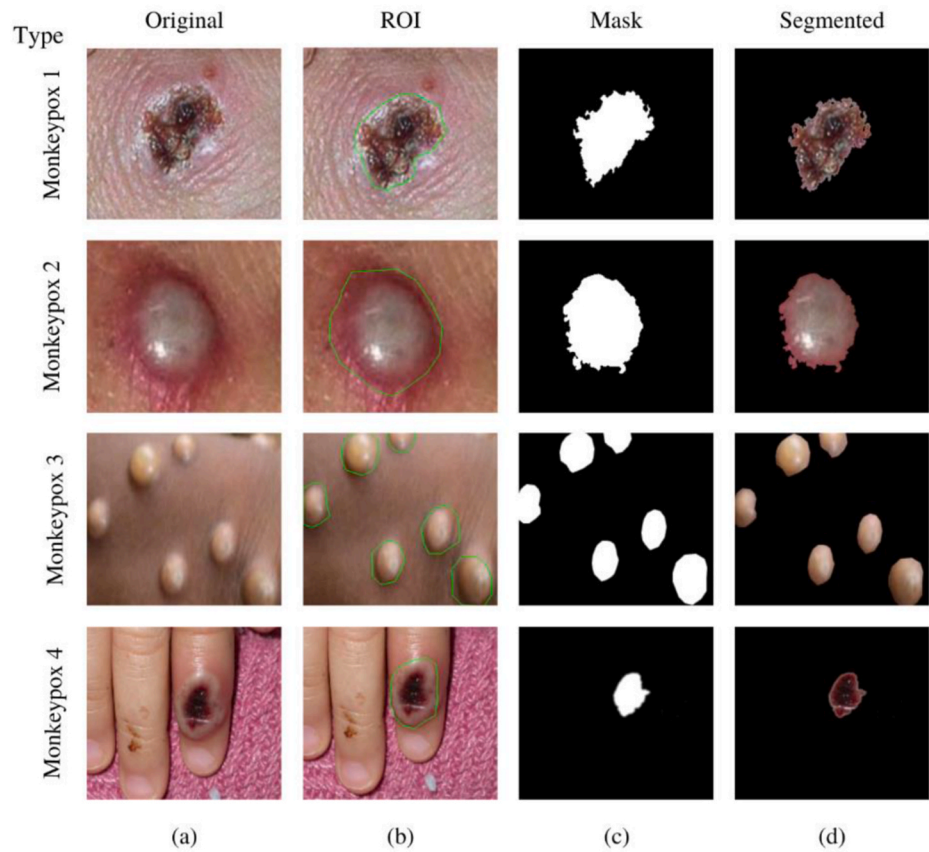


Fig. 12. Threshold-based segmentation (a) Original, (b) ROI, (c) Mask, (d) Segmented.

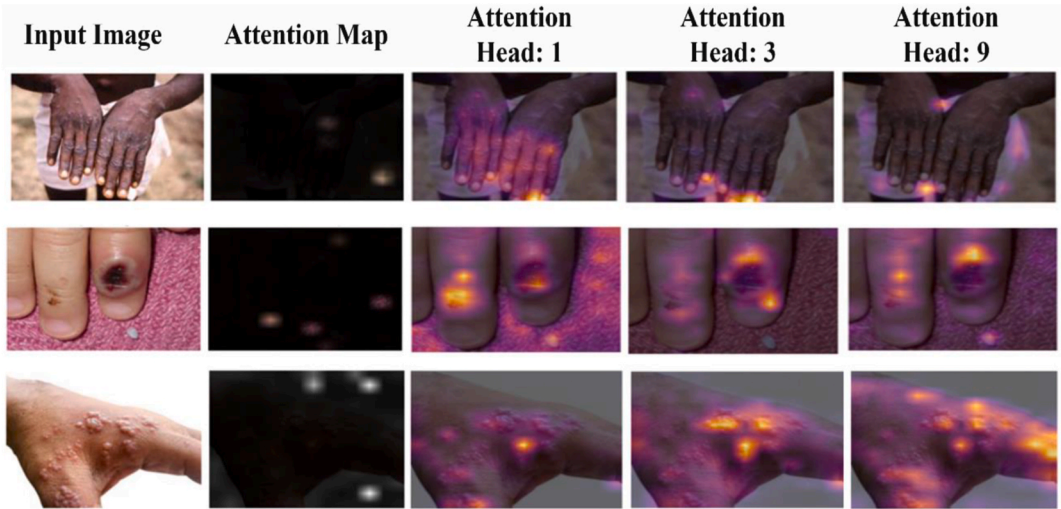


Fig. 13. Visualization of Attention Weights in Monkeypox Images using Transformer block and attention heads 1,3 and 9.

layer. This visualization allows us to understand how the model focuses its attention on different layers and with different heads.

During our experiment, we found that the pre-trained models and ViT models trained with Adam performed better compared to those trained with SGD. In our study, which included four distinct experiments, this trend remained constant. Based on our understanding, there are several reasons that might play important roles in this performance discrepancy. First, as described by Kingma and Ba (2014), Adam optimizes training by adapting the learning rates for individual parameters, a feature particularly beneficial for dealing with the complex

architectures of pre-trained and ViT models (Kingma & Ba, 2014). Second, the inclusion of momentum in Adam, as outlined by Ruder (2016), helps in accelerating convergence and effectively navigating the complex optimization landscapes of typical deep neural networks (Ruder, 2016). Lastly, Adam’s ability to handle sparse gradients more efficiently than SGD, a point also noted by Reddi, Kale, & Kumar (2018) and Zhang et al. (2019), could also contribute to its superior performance in these specific scenarios (Reddi, Kale, & Kumar, 2018; Zhang et al., 2019) and making it a more suitable choice than SGD.

From our overall observations, it can be inferred that TL models,

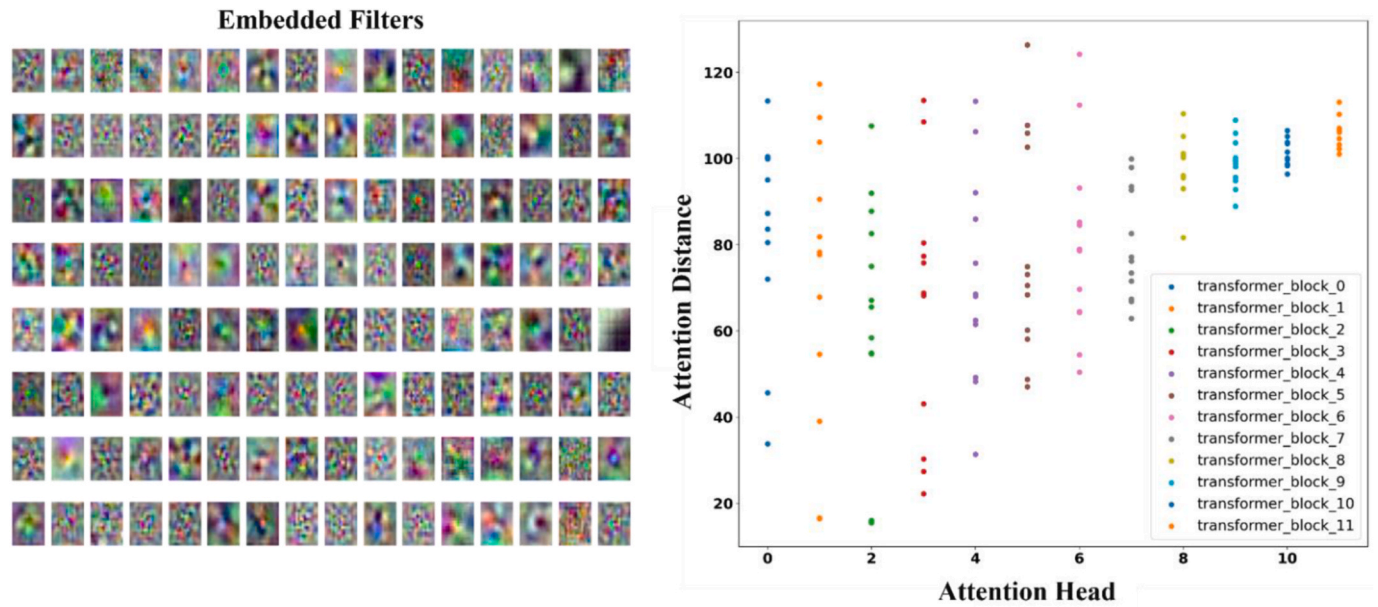


Fig. 14. On the left, the filters for the initial linear embedding of the RGB values in the Vision Transformer (ViT) can be seen. On the right, the size of the area being attended to by the head and network depth is illustrated. Each dot represents the average attention distance across all images for one of the 16 heads at a specific layer.

such as M-VGG16 and M-ResNet50, perform better than the ViT model. Across multiple studies, it was also observed that TL-based approaches are more feasible and computationally efficient. One possible reason for this could be the limited size of the dataset. During the early stages of the experiment, the proposed approaches demonstrated high levels of accuracy in diagnosing Monkeypox disease. However, as the performance of the models improved, over-fitting began to occur. To combat this issue, we implemented early stopping techniques in order to stabilize the training and validation phases.

In the field of AI-based clinical diagnosis, it is crucial to evaluate the true potential of a model's predictions in terms of Explainability. In our work, we used LIME to interpret the predictions of TL models (M-VGG16, M-ResNet50, and M-ResNet101) and visual representations to interpret the predictions of ViT models. Our overall observations found that the proposed models successfully localized the infected regions during the prediction phase. To further verify our findings, we cross-checked them with segmented images of the skin of patients with Monkeypox disease, which is absent in the existing literature. Our findings suggest that the proposed model's performance is satisfactory for binary classification using M-VGG16 models trained (during Study One and Two) with Adam optimizer and for a multi-class dataset using M-ResNet50 (during Study Three) once trained with Adam optimizer.

In order to confirm the accuracy of our results, we compared them to segmented images of the skin of patients diagnosed with Monkeypox disease. This type of validation is not commonly found in the existing literature. Our findings indicate that the proposed models perform satisfactorily for binary classification using M-VGG16 models trained with Adam optimizer during Study One and Two and for multi-class datasets using ResNet50 during Study Three.

While our study has yielded promising results, it is important to note that certain limitations have been identified, providing opportunities for further research. Specifically, these limitations include:

1. The experiment was conducted solely utilizing image data. However, incorporating additional patient information, such as age and temperature, may enhance the robustness of the ML-based diagnosis model. Unfortunately, due to the unavailability of a dataset that includes both patient history and image data, creating such a model is beyond the scope of this study.

3. The evaluation of the proposed model was conducted on the

collected dataset. However, to further assess the model's performance and generalizability, it is necessary to evaluate it in a clinical setting. This step is crucial for real-time deployment, but it requires comprehensive clinical trials involving real-time patients, specific clinical settings, and regulatory approvals from entities like the FDA and CDC. Given the limited access to clinical facilities and the constrained time-frame of our study, the feasibility of conducting extensive clinical trials involving real-time patient data was beyond our current scope. Therefore, the primary objective of our research was to establish a foundational understanding and create a baseline for the application of deep learning (DL) methodologies in clinical domains, particularly for life-threatening contagious diseases like Monkeypox. We aim to highlight the potential of DL approaches as future solutions in this domain, acknowledging that further research, including clinical trials, is essential for practical implementation. This study, therefore, serves as a preliminary step towards more exhaustive future research that could bridge the gap between theoretical models and clinical application in the diagnosis of such diseases.

7. Conclusion

In this study, we proposed and evaluated M-VGG16, M – ResNet50, M-ResNet101, and ViT approaches for the classification of Monkeypox using image analysis techniques. Our proposed M-VGG16 model achieved an accuracy of $88 \pm 7.6\%$ when trained and tested on a small dataset using the Adam optimizer. This aligns with the findings of many previous studies, which demonstrate the superiority of TL on small datasets [34,37,38]. Additionally, we tested our proposed model on an imbalanced dataset (during Study Two and Four) and obtained an accuracy of $76 \pm 2.23\%$ and $77 \pm 3.67\%$ on the test set.

Furthermore, in Study Three, the M-ResNet50 model demonstrated the best performance by achieving an accuracy of $89 \pm 2.1\%$ on a multi-class dataset. In our experiments, we consistently observed that our proposed pre-trained and ViT models trained with Adam outperformed those trained with SGD, likely due to Adam's adaptive learning rates, momentum inclusion, and superior handling of sparse gradients.

Additionally, the proposed model's performance is interpreted using various interpretation approaches, such as LIME and visual representation, often missing in most of the referenced literature. Given that our

proposed model aligns with previous literature that has employed the TL approach in developing AI-based diagnosis models, it may inspire future research and practitioners to leverage this approach in clinical diagnosis. Moreover, the addition of FL serves as a testament to our commitment to embracing cutting-edge technologies in the pursuit of improved healthcare solutions. Some limitations of our work may be addressed through the ongoing collection of new images of Monkeypox-infected patients, verification of the model's performance by medical experts, testing the proposed model on highly imbalanced data, and implementation of the model in the development of mobile-based diagnostic tools.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.

CRediT authorship contribution statement

Md Manjurul Ahsan: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Tasfiq E. Alam:** Methodology, Writing – review & editing. **Mohd Ariful Haque:** Methodology, Writing – review & editing. **Md Shahin Ali:** Conceptualization, Data curation, Software, Writing – review & editing. **Rakib Hossain Rifat:** Data curation, Software, Validation, Writing – review & editing. **Abdullah Al Nomaan Nafi:** Data curation, Writing – review & editing. **Md Maruf Hossain:** Data curation, Writing – review & editing. **Md Khairul Islam:** Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the full team members of this manuscript headed by Dr. Md Manjurul Ahsan who helped us and provided valuable insight into the research.

References

- [1] Alakunle E, Moens U, Nchinda G, Okeke MI. Monkeypox virus in Nigeria: infection biology, epidemiology, and evolution. *Viruses* Nov. 2020;12(11):1257. <https://doi.org/10.3390/V12111257>.
- [2] Moore MJ, Rathish B, Zahra F. Mpox (monkeypox). *Nurs Pract* May 2023;48(4): 13–20. <https://doi.org/10.1097/01.npr.00000000000000025>.
- [3] Nolen LD, et al. Extended human-to-human transmission during a monkeypox outbreak in the democratic Republic of the Congo. *Emerg Infect Dis* Jun. 2016;22(6):1014. <https://doi.org/10.3201/EID2206.150579>.
- [4] Nguyen PY, Ajiseigiri WS, Costantino V, Chughtai AA, MacIntyre CR. Reemergence of human monkeypox and declining population immunity in the context of urbanization, Nigeria, 2017–2020. *Emerg Infect Dis* Apr. 2021;27(4):1007. <https://doi.org/10.3201/EID2704.203569>.
- [5] Doucleff M. Scientists warned us about monkeypox in 1988. Here's why they were right. 2022. . [Accessed 27 May 2022].
- [6] Ola P. The origin of the mysterious multi-country monkeypox outbreak in non-endemic countries. 2022.
- [7] Organization WH. Multi-country monkeypox outbreak in non-endemic countries, vol. 7. World Health Organization; 2022. www.who.int/emergencies/disease-outbreak-news/item/2022-DON390.
- [8] Park A. There's already a monkeypox vaccine. But not everyone may need it. 2022. . [Accessed 27 May 2022].
- [9] Ahsan MM, et al. Deep transfer learning approaches for Monkeypox disease diagnosis. *Expert Syst Appl* Apr. 2023;216:119483. <https://doi.org/10.1016/J.ESWA.2022.119483>.
- [10] de Oliveira Thomasi RM, et al. Molecular methods for diagnosis of monkeypox: a mini-review. *medRxiv* 2022:2012–22.
- [11] Ahsan MM, Siddique Z. Machine learning-based heart disease diagnosis: a systematic literature review. *Artif Intell Med* 2022;128:102289.
- [12] Miranda GHB, Felipe JC. Computer-aided diagnosis system based on fuzzy logic for breast cancer categorization. *Comput Biol Med* 2015;64:334–46.
- [13] Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks. *Comput Biol Med* 2020;121: 103795.
- [14] Sahin VH, Oztel I, Yölcü Oztel G. Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application. *J Med Syst* 2022; 46(11):79.
- [15] Sitaula C, Shahi TB. Monkeypox virus detection using pre-trained deep learning-based approaches. *J Med Syst* 2022;46(11):78.
- [16] Akin KD, Gurkan C, Budak A, Karataş H. Classification of monkeypox skin lesion using the explainable artificial intelligence assisted convolutional neural networks. *Avrupa Bilim ve Teknol Derg* 2022;40:106–10.
- [17] Irmak MC, Aydin T, Yağanoglu M. Monkeypox skin lesion detection with MobileNetV2 and VGGNet models. In: 2022 medical technologies congress. TIPTEKNO; 2022. p. 1–4.
- [18] Alcalá-Rmz V, Villagrana-Bañuelos KE, Celaya-Padilla JM, Galván-Tejada JJ, Gamboa-Rosales H, Galván-Tejada CE. Convolutional neural network for monkeypox detection. In: International conference on ubiquitous computing and ambient intelligence; 2022. p. 89–100.
- [19] Rieke N, et al. The future of digital health with federated learning. *NPJ Digit Med* 2020;3(1):119.
- [20] Li P, et al. ChainSDI: a software-defined infrastructure for regulation-compliant home-based healthcare services secured by blockchains. *IEEE Syst J* 2019;14(2): 2042–53.
- [21] Hao M, Li H, Xu G, Liu Z, Chen Z. Privacy-aware and resource-saving collaborative learning for healthcare in cloud computing. In: ICC 2020-2020 IEEE international conference on communications. ICC; 2020. p. 1–6.
- [22] Ahsan MM, Uddin MR, Luna SA. Monkeypox image data collection. *arXiv Prepr. arXiv2206.01774* 2022.
- [23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv Prepr. arXiv1409.1556* 2014.
- [24] Popescu MC, Sasu LM. Feature extraction, feature selection and machine learning for image classification: a case study. In: 2014 international conference on optimization of electrical and electronic equipment. OPTIM; 2014. p. 968–73.
- [25] Akiba T, Suzuki S, Fukuda K. Extremely large minibatch sgd: training resnet-50 on imagenet in 15 minutes. *arXiv Prepr. arXiv1711.04325* 2017.
- [26] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
- [27] Dosovitskiy A, et al. An image is worth 16x16 words: transformers for image recognition at scale. *arXiv Prepr. arXiv2010.11929* 2020.
- [28] Ahmed WS. The impact of filter size and number of filters on classification accuracy in CNN. In: 2020 International conference on computer science and software engineering. CSASE; 2020. p. 88–93.
- [29] Ahsan MM, et al. Detecting SARS-CoV-2 from chest X-Ray using artificial intelligence. *IEEE Access* 2021;9:35501–13.
- [30] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res* 2012;13(2).
- [31] Ribeiro MT, Singh S, Guestrin C. 'Why should i trust you?' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1135–44.
- [32] Ribeiro MT, Singh S, Guestrin C. Model-agnostic interpretability of machine learning. *arXiv Prepr. arXiv1606.05386* 2016.
- [33] Corneanu C, Madadi M, Escalera S, Martinez A. Explainable early stopping for action unit recognition. In: 2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020); 2020. p. 693–9.
- [34] Ali MS, Islam MK, Das AA, Duranta DUS, Haque MF, Rahman MH. A novel approach for best parameters selection and feature engineering to analyze and detect diabetes: machine learning insights. *BioMed Res Int* May 2023;2023:1–15. <https://doi.org/10.1155/2023/8583210>.
- [35] Narin A, Kaya C, Pamuk Z. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *Pattern Anal Appl* 2021;24:1207–20.
- [36] Hussain MA, Islam T, Chowdhury FUH, Islam BMR. Can artificial intelligence detect Monkeypox from digital skin images? *BioRxiv* 2022:2008–22.
- [37] Ali SN, et al. Monkeypox skin lesion detection using deep learning models: a feasibility study. *arXiv Prepr. arXiv2207.03342* 2022.
- [38] Haque ME, Ahmed MR, Nila RS, Islam S. Classification of human monkeypox disease using deep learning models and attention mechanisms. *arXiv Prepr. arXiv2211.15459* 2022.