H-MpoxNet: A Hybrid Deep Learning Framework for Mpox Detection from Image Data

Sajal Chakroborty[a]

[a] *Worcester Polytechnic Institute, Worcester, Massachusetts, USA, email:schakroborty@wpi.edu , sajal.math@yahoo.com*

ARTICLE INFO

Keywords:
Deep learning
Machine learning classifiers
Feature extraction
Mpox detection
Infectious disease
Image data analysis

ABSTRACT

Infectious diseases pose significant global threats to public health and economic stability by causing pandemics. Early detection of infectious diseases is crucial to prevent global outbreaks. Mpox, a contagious viral disease first detected in humans in 1970, has experienced multiple epidemics in recent decades, emphasizing the development of tools for its early detection. In this paper, we propose a hybrid deep learning framework for Mpox detection. This framework allows us to construct hybrid deep learning models combining deep learning architectures as a feature extraction tool with machine learning classifiers and perform a comprehensive analysis of Mpox detection from image data. Our best-performing model consists of MobileNetV2 with LightGBM classifier, which achieves an accuracy of 91.49%, precision of 86.96%, weighted precision of 91.87%, recall of 95.24%, weighted recall of 91.49%, F1 score of 90.91%, weighted F1-score of 91.51% and Matthews Correlation Coefficient score of 0.83.

1. Introduction

Mpox, also known as monkeypox (MPX), is caused by the Mpox virus (MPXV). It is a zoonotic disease, more specifically an infectious skin disease that can spread between animals and humans. It is a 200-250 nm brick-like or ovoid shaped double helix DNA virus that belongs to the Orthopoxvirus genus, Proxviriade family, and Chordopoxvirinae subfamily [68]. MPXV can be transmitted into the human body in several ways, from animal to human and human to human [68]. Moreover, it can be transmitted by contact with contaminated objects, a patient's droplets, body fluids, and lesions [68]. After exposure to MPXV, it takes 3-17 days to develop symptoms, which can last nearly 2-5 weeks [15]. According to the Centers for Disease Control and Prevention, a few symptoms of Mpox are fever, swollen lymph nodes, headache, muscle aches and backache.

The first human case caused by MPXV was reported in 1970 [27, 43]. In 2022-23, there was a global outbreak of Mpox [31]. Before that, between 1970 and 2017, several outbreaks occurred in central and west Africa [53, 13, 24]. Clade I, previously known as the Congo Basin Clade or Central African Clade, and Clade II, previously known as West African Clade, were responsible for the outbreaks in Central and West Africa [14]. The first outbreak outside Africa occurred in the USA in 2003 [46, 14]. In 2023, 809 positive cases were reported in different states of the USA, and 1,945 have been reported up to September 09, 2024 [16]. In Figure 1, we present a scenario based on data collected from the websites of the "Centers for Disease Control and Prevention" and "Our World in Data" for different states in the USA and worldwide. From these reported cases, it is visible that developing tools for the early detection of Mpox has evolved into a significant research problem.

A widely used diagnostic tool for infectious diseases is Polymerase Chain Reaction (PCR) test [25]. It provides precise genetic analysis. However, PCR has risks of contamination, and limitations in detecting large or unknown sequences [69]. Moreover, it requires specialized equipment and expertise, making it less accessible in some resource-limited settings [69].

In recent years, deep learning (DL) techniques have become an alternative to detect infectious diseases. For a detailed review, one can see [11], [32], and [60]. While PCR is more likely inclined toward the lab resources and has a risk of contamination, DL techniques are mainly focused on biomedical images. DL techniques are data driven and a subclass of artificial intelligence [30]. It can predict patterns from large and complex datasets [30]. It involves training neural networks that can mimic the neural orientation of the human brain [30]. Most importantly, once the model is trained, it can be deployed on a simpler system to predict, classify, and generate output [30].

In addition to DL, there are machine learning (ML) tools for classification. For example, Support vector machine (SVM), Logistic regression, Random Forest (RF), Naïve bayes (NB). Due to computational efficiency these classifiers have found wide ranging applications in various fields [49, 56].

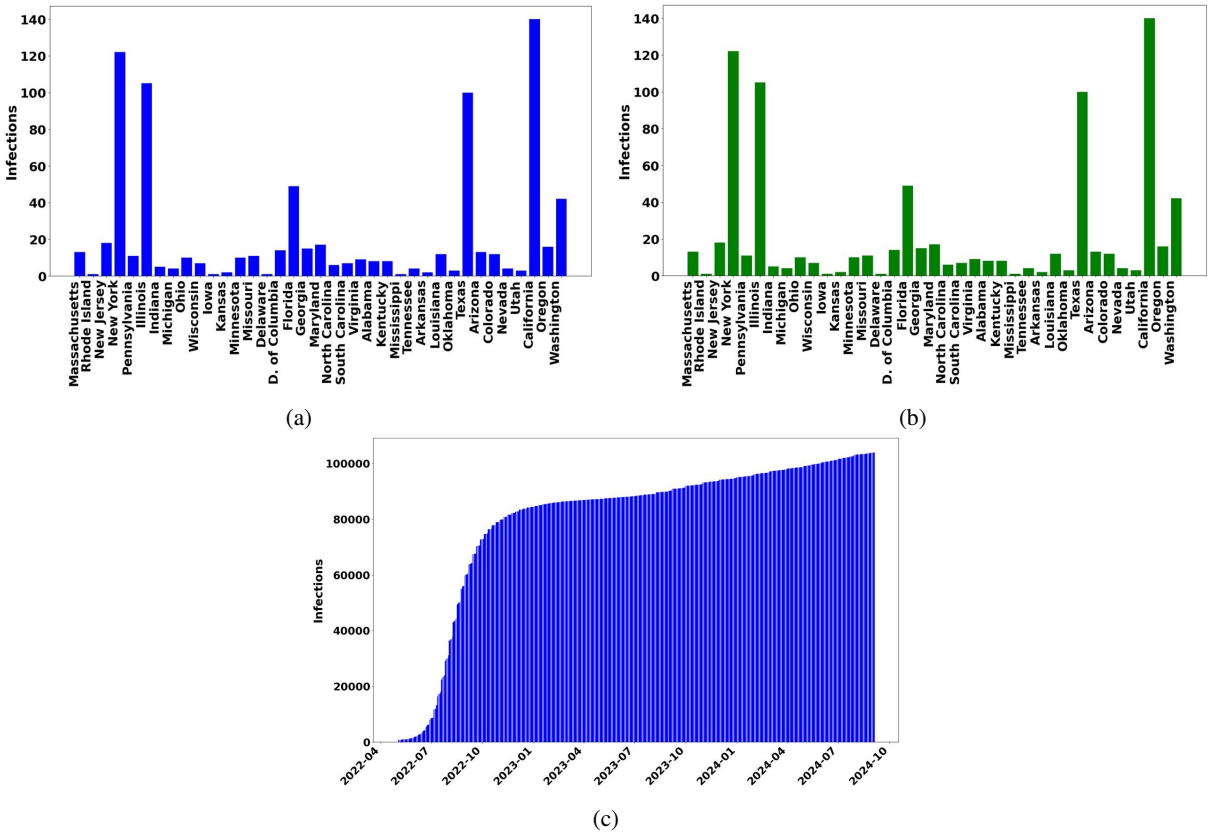

Figure 1: Mpox reported cases in the USA and Worldwide. Figures 1a and 1b illustrate the number of infected cases in 2023 and 2024, respectively, in different states of the USA, and 1c shows cumulative infected cases reported globally up to September 2024.

A common thread among these ML and DL techniques for disease detection is their use of biomedical images, such as histopathological ([45, 2, 4]) and MRI images ([7, 8, 40]). For instance, chest X-ray images have been used for early COVID-19 detection (see [59], [35], and [34]). Notably, these techniques do not pose any contamination risk, unlike PCR.

Although DL and ML techniques have demonstrated satisfactory performance in disease prediction, they require a large volume of images for model training [39]. Moreover, the training process demands high-performance computational resources [39]. Feature extraction from images is an initial and essential step for training and testing the models, which also requires significant computational time [39]. Therefore, developing a computational strategy to reduce this time has been an active area of research for a long period.

Furthermore, datasets could be imbalanced, and due to this, model prediction accuracy could be erroneous [18]. Consequently, developing a robust model could be very challenging. Most studies in literature have addressed this issue through data augmentation [61]. However, data augmentation for images is much more complex compared to tabular data. Some widely used techniques for image data augmentation include generative adversarial networks (GANs) and diffusion generative models [61]. However, the training process for these models is highly computationally expensive [61].

Various DL, ML, and hybrid methods have been proposed in prior studies to identify Mpox from skin lesion images. A few research gaps we have identified in these studies include: to develop robust models, most of these papers implement various data augmentation techniques; however, they did not address whether this augmentation introduces any bias. Since imbalanced data can produce erroneous accuracy, some other metrics that are widely used for imbalanced data were not explored. Finally, very few studies have shown a comprehensive analysis of various DL or ML models for Mpox detection. We elaborate on these research gaps in the literature review section.

The primary contributions of this paper to Mpox detection include:

i. We develop a hybrid DL framework, which combines a pretrained DL architecture with an ML classifier. We use pretrained DL architectures to extract features from images. After that, we train the ML model for Mpox prediction.

ii. We perform image augmentation using standard techniques that have been used in prior studies to develop robust hybrid models. To validate that these augmented images do not introduce any biases, we perform the Kullback-Leibler divergence test. To our knowledge, no prior studies have done this.

iii. We assess the robustness of our hybrid models through a comprehensive analysis, computing various metrics, including accuracy, precision, weighted precision, recall, weighted recall, F1 score, weighted F1 score, receiver operating characteristic (ROC) curve, and area under the curve (AUC). Since imbalanced data can impact accuracy, we also compute the Matthews correlation coefficient (MCC), a widely used metric for imbalanced datasets. To our knowledge, very few studies have computed MCC scores to validate models for Mpox detection. Furthermore, no prior studies have conducted the comprehensive analysis that we present in this work.

iv. Finally, we make a comparison of our work with results published in prior studies.

We organize this paper as follows: In Section 2, we review existing studies on Mpox detection and highlight their research gaps. In Section 3, we discuss our dataset, train-test split, and data augmentation procedure and perform Kullback-Leibler divergence on both the original and augmented images. We also develop a hybrid deep learning framework in this section. Moreover, we provide an overview of the pre-trained DL and ML methods and metrics used in this study. In Section 4, we present the results and compare them with prior studies. Finally, we conclude the paper in Section 5.

2. Literature review

This section elaborates the introduction by reviewing existing techniques implemented in Mpox detection. Mpox is an infectious disease and one can study the dynamics of an infectious disease by developing models using a system of ordinary differential equations. A simple framework for these models is the SIR (susceptible, infectious, recovered) model, given by [28]:

$$\begin{aligned} \frac{dS}{dt} &= \Lambda - a\frac{IS}{N} \\ \frac{dI}{dt} &= a\frac{IS}{N} - bI \\ \frac{dR}{dt} &= bI, \end{aligned} \tag{1}$$

with initial conditions $S(0) = S_0, I(0) = I_0, R(0) = R_0$. Here S, I, and R are the number of individuals in the susceptible, infectious, and recovered compartments, Λ represents the recruitment number in the susceptible compartment, a is the infection or disease transmission rate from susceptible to infection, and b denotes the recovery rate. Note that $t > 0$ and the total population is given by:

$$N(t) = S(t) + I(t) + R(t). \tag{2}$$

Using this model, one can compute the basic reproduction number $\mathcal{R}_0$, which is a threshold that determines whether the disease will spread or die out.

Due to the mathematical elegance, this model is widely popular and found wide ranging applications in the published literature. Recently, this infectious disease model has been studied to analyze the dynamics of Mpox and predict the virus's spread [33, 3, 54].

[44] proposed a SEIR-type model for continuous monitoring and early intervention of the spread of Mpox virus. They proposed a robust model and performed sensitivity analyses for different parameters such as infection rate and recovery rate. Most importantly, they estimated the basic reproduction number for 2022 outbreak to investigate how rapidly the virus could spread.

[9] investigated the dynamics of Mpox outbreak using fractional-order SEIR model. They analyzed the effects of vaccination in the transmission and mitigation of Mpox virus. One can find a similar type of model outlined in [1].

[10] developed a model focusing on pair formation that reflects how close contact with individuals contributes to the spread of Mpox. Additionally, they investigated recovery dynamics to observe how quickly infected individuals recover and how this recovery rate affects the overall spread of the infection. Using infectious disease modeling and statistical techniques, [71] forecasted the global spread and trajectory of the Mpox virus. They emphasized identifying key factors responsible for the spread of MPXV. By analyzing the current data, they predicted the future scenario of the outbreak.

[36] performed a comprehensive analysis of the 2022 Mpox outbreak in New York City. They analyzed and compared different mathematical models,

including SEIR, and evaluated their effectiveness in predicting and forecasting outbreak patterns. [48] has provided a comprehensive review on infectious disease modeling to investigate the Mpox transmission.

Although these models have received significant attention for investigating the dynamics of Mpox transmission, they are not well-suited for early disease prediction. Machine learning and deep learning techniques fill this gap.

[5] created an image dataset to detect and analyze Mpox with DL algorithms. The dataset consists of images with a resolution of 224×224×3 and allows the implementation of DL models. [5] applied three pre-trained models, VGG-16, InceptionV3, and ResNet-50, to classify Mpox images. In addition to that, they combined these DL algorithms by employing majority voting and proposed an ensemble model. Among these models, ResNet-50 achieved 82.96% accuracy, 87% precision, 83% recall, and 84% F1 score, while other models, including the ensemble, produced lower scores in every metric. However, this study did not mention whether their data augmentation introduced any biases in their training dataset, which is clearly a significant research gap.

Using the same dataset, [58] developed a DL framework tailored for mobile-based applications to diagnose Mpox lesions. They obtained 91.11% accuracy, 90% precision, 90% recall, and 90% F1 score. This study did not validate the augmented images for their medical relevance.

[62] proposed an ensemble-based DL framework focused not only on Mpox detection but also on classifying other skin diseases. Their dataset was curated with samples of different skin diseases. Initially, they implemented 13 different DL techniques, then ensembled two best-performing algorithms—Xception and DenseNet-169 and achieved 87.13% accuracy, 85.44% precision, 85.47% recall, and 85.40% F1 score. However, this study analyzed their models only based on accuracy, precision, recall, and F1-scores. Note that imbalanced data could produce biased accuracy. Additionally, this study did not compute metrics such as the MCC score, which is widely used to evaluate model performance on imbalanced data.

[23] explored ResNet-50, EfficientNetb3, and EfficientNetb7 on a dataset that contains 160 Mpox lesions, 178 chickenpox, 54 cowpox, 358 small pox, and 50 healthy skins. They obtained 87% accuracy, 92% precision, 87% recall, and 90% F1 score by EfficientNetb3. Although this study trained several deep learning algorithms to detect Mpox, similar to prior studies ([5, 58]), they did not investigate whether their data augmentation introduced any biases. Additionally, they did not compute MCC scores or any other metrics tailored for imbalanced data.

[67] proposed a hybrid DL framework for Mpox detection. Unlike developing an ensemble model employing majority voting, they combined the two models with the highest accuracy with the Long Short-term Memory (LSTM) model and obtained 87% accuracy, 84% precision, 87% recall, and 85% F1 score. This study did not address the interpretability of the model's decisions, which is considered a crucial component in medical diagnostics for clinical acceptance and trust.

[50] tested various deep learning models, including GoogleNet, ResNet-18, ResNet-50, ResNet-101, and SqueezeNet, for multi-class classification and Mpox detection, achieving an average accuracy of 91.19%. Explainable artificial intelligence (XAI) plays a crucial role in medical research, and [50] introduced XAI to make the deep learning model more interpretable. However, there is no discussion on the potential biases that could be introduced by the data augmentation process.

[47] used VGG19, Xception, InceptionV3 transfer learning models on augmented dataset created by [4], which includes, 1428 Mpox and 1764 others Skin lesions images. They achieved 98% accuracy by using InceptionV3 model. It is important to note that this study used augmented data created by [5]. As previously mentioned, there is no information on the qualitative or quantitative validation conducted to ensure that these augmented images do not introduce bias.

[70] explored vision transformers (VITs) and convolutional neural networks (CNNs) for Mpox lesion analysis. They proposed an ensemble learning strategy using the bagging ensemble technique and obtained 81.91% accuracy. This study did not explore different data augmentation strategies, nor did it address the interpretability of the Vision Transformer and CNN.

3. Methodology

In this section, we discuss our data, along with the train-test split and data augmentation procedure. Moreover, we introduce Kullback-Leibler divergence as a validation tool to demonstrate that augmentation does not significantly alter the data distribution. We also explain the hybrid DL framework for Mpox detection. Additionally, we briefly describe the DL architecture and ML classifiers, along with various evaluation metrics.

3.1. Data description

We use a publicly available dataset for Mpox detection, created by [5]. The dataset contains images

with a resolution of 224 × 224 × 3. It is divided into two categories–Mpox and non-Mpox, where the latter includes images of other diseases–chickenpox and measles. In Table 1, we present a summary of the dataset, including the number of images in each category.

Table 1
Summary of the dataset

Category	Number of Images	Percentage
Mpox	102	45%
Others	126	55%
Total	**228**	**100%**

3.1.1. Data preprocessing

In this step, we clean the data to ensure that there are no mislabeled or noisy images. Generally, image intensity values range from 0 to 255. We normalize the images so that the pixel values lie between 0 and 1.

3.1.2. Train-test split

We split the dataset into two subsets—the train set, which contains 80%, and the test set, which includes 20% of the samples from the original dataset. We present a detailed summary in Table 2.

Table 2
Summary of the train and test data

Category	Training set	Test set
Mpox	81	21
Others	100	26
Total	**181**	**47**

3.2. Data augmentation

From Tables 1 and 2, we can see that the dataset is imbalanced, with significantly fewer Mpox images compared to other skin lesion images. To create robust models, we perform data augmentation on the training data. We employ various strategies, including rotation, horizontal flipping, width and height shifts, zooming, fill mode adjustments, and shear range modifications. Table 3 presents the total number of augmented images, while Table 4 provides detailed information on the augmentation strategies and their optimized values.

3.2.1. Kullback-Leibler divergence

Kullback-Leibler (KL) divergence, also known as relative entropy, is an essential metric for determining statistical differences between two distributions. We compute KL divergence to assess whether there is a significant difference between the original and augmented images. For empirical distributions P and Q defined on the same sample space $\mathcal{V}$, KL divergence is defined as [19]:

Table 3
Summary of the augmented images

Category	Augmented images
Mpox	200
Others	200
Total	**400**

$$\mathrm{KL}(P||Q) = \sum_{x\in\mathcal{V}} P(x)\log\left(\frac{P(x)}{Q(x)}\right). \tag{3}$$

Equation (3) shows that the two distributions P and Q are nearly identical if $\mathrm{KL}(P||Q)$ is zero or approximately close to zero.

3.3. Hybrid deep learning framework for Mpox detection

Consider the dataset $S = \left\{ \left(\mathcal{X}_i, \mathcal{Y}_i\right) \right\}_{i=1}^{n}$, where $\mathcal{X}_i$ denotes the ith image and $\mathcal{Y}_i$ represents its corresponding image label. We develop the framework by combining a pre-trained DL architecture and an ML classifier in the following steps:

Step 1 (Preprocessing): In this initial phase, we clean the dataset through various processes, including noise reduction, handling mislabeled images, and removing duplicate images to ensure the dataset is prepared for optimal model performance.

Step 2 (Train-test split): We partition the given dataset S into two distinct subsets, S_{Train} and S_{Test}, such that

$$S = S_{\text{Train}} \cup S_{\text{Test}}, \text{ and } S_{\text{Train}} \cap S_{\text{Test}} = \phi,$$

where S_{Train} denotes the training set containing 80% of the sample and S_{Test} denotes the test set containing 20% of the samples from the dataset S. We perform data augmentation on S_{Train}.

Step 3 (Feature extraction): In this step, using a pre-trained DL architecture $\mathcal{F}$, we extract features from each image as follows:

$$x_j^{\text{Train}} = \mathcal{F}\left(\mathcal{X}_j^{\text{Train}}\right),\ j = 1, 2, \ldots, N \tag{4}$$

$$x_k^{\text{Test}} = \mathcal{F}\left(\mathcal{X}_k^{\text{Test}}\right),\ k = 1, 2, \ldots, M, \tag{5}$$

where x_j^{Train} and x_k^{Test} represent the features extracted from $\mathcal{X}_j^{\text{Train}} \in S_{\text{Train}}$ and $\mathcal{X}_k^{\text{Test}} \in S_{\text{Test}}$ respectively.

Table 4
Summary of the augmentation parameters

Augmentation Parameter	Description	Value
Rotation range	Degree range for random rotation	20 degree
Width shift range	Range of random horizontal shift	20% of the image width
Height shift range	Range of random vertical shift	20% of the image height
Shear range	Transformation that slants the image shape	0.2 radians
Zoom range	Range of random zooming	20% of the image
Horizontal flip	Randomly flip images horizontally	True
Fill mode	Fill strategy for newly created pixels	Nearest

Note that each extracted feature is a tensor of dimension $l \times p \times q$, where l and p are the height and width of the feature map and q denotes the number of feature channels. In order to train a ML classifier, we have reshaped each of these features into a d-dimensional vector where $d = l \times p \times q$. The reshaping procedure is performed as follows:

$$z_j^{\text{Train}} = \text{reshape}\left(x_j^{\text{Train}}, d\right), \tag{6}$$

$$z_k^{\text{Test}} = \text{reshape}\left(x_k^{\text{Test}}, d\right), \tag{7}$$

where j ranges from 1 to N, and k ranges from 1 to M.

Step 4 (Train classification model): Using these features z_j^{Train}, we train a classification model $\mathcal{G}$. After that, we test the model using the features z_k^{Test} and predict the labels $\hat{\mathcal{Y}}_k^{\text{Test}}$ as follows:

$$\hat{\mathcal{Y}}_k^{\text{Test}} = \mathcal{G}(z_k^{\text{Test}}),\ k = 1, 2, \ldots, M. \tag{8}$$

Step 5 (Model Evaluation): Finally, we evaluate the hybrid model using standard metrics such as accuracy, precision, weighted precision, recall, weighted recall, F1 score, weighted F1 score, and Matthews Correlation Coefficient (MCC) score. We discuss these metrics in subsection 3.6.

In Figure 2, we illustrate this framework in detail. Additionally, we include all essential variables and notations, along with their descriptions, in Table 5.

Table 5
Description of the notations and variables

Notations	Description
S	Image dataset
S_{Train}	Training dataset
S_{Test}	Test dataset
n	Total number of images
N	Total number of images in S_{Train}
M	Total number of images in S_{Test}
$\mathcal{F}$	Pre-trained deep learning architecture
$\mathcal{G}$	Classification model
$\mathcal{X}_i$	i^{th} image in S
$\mathcal{Y}_i$	Label of the ith image $\mathcal{X}_i$
$\mathcal{X}_j^{\text{Train}}$	j^{th} image in S_{Train}
$\mathcal{X}_k^{\text{Test}}$	k^{th} image in S_{Test}
x_j^{Train}	Extracted feature from $\mathcal{X}_j^{\text{Train}}$
x_k^{Test}	Extracted feature from $\mathcal{X}_k^{\text{Test}}$
l	Height of the feature map
p	Width of the feature map
q	Number of feature channels
d	Dimension of the reshaped features
z_j^{Train}	Reshaped training feature
z_k^{Test}	Reshaped test feature
$\hat{\mathcal{Y}}$	Predicted labels
H_0	Null hypothesis
H_A	Alternative hypothesis
TP	True positive
TN	True negative
FP	False positive
FN	False negative

3.4. Pre-trained deep learning architectures

We implement several pre-trained DL architectures for feature extraction, including DenseNet, MobileNet, Inception, Inception-Residual Network, and Xception. We discuss them briefly in the following subsections.

3.4.1. DenseNet

All the layers of this architecture are connected in a feedforward manner, where all the features from the preceding layers are concatenated and transferred to the successive layers [22]. This process allows to reuse the features, leverage the gradient flow, and prevent the vanishing gradient problem. In addition to that, it has transition layers that consist of batch normalization, ReLu activation function, 1×1 convolution to reduce dimensions, and 2×2 average pooling that controls the growth of the number of features and consequently reduces the computational complexity [22]. We implement three versions of DenseNet for feature extraction: DenseNet-121, DenseNet-169, and

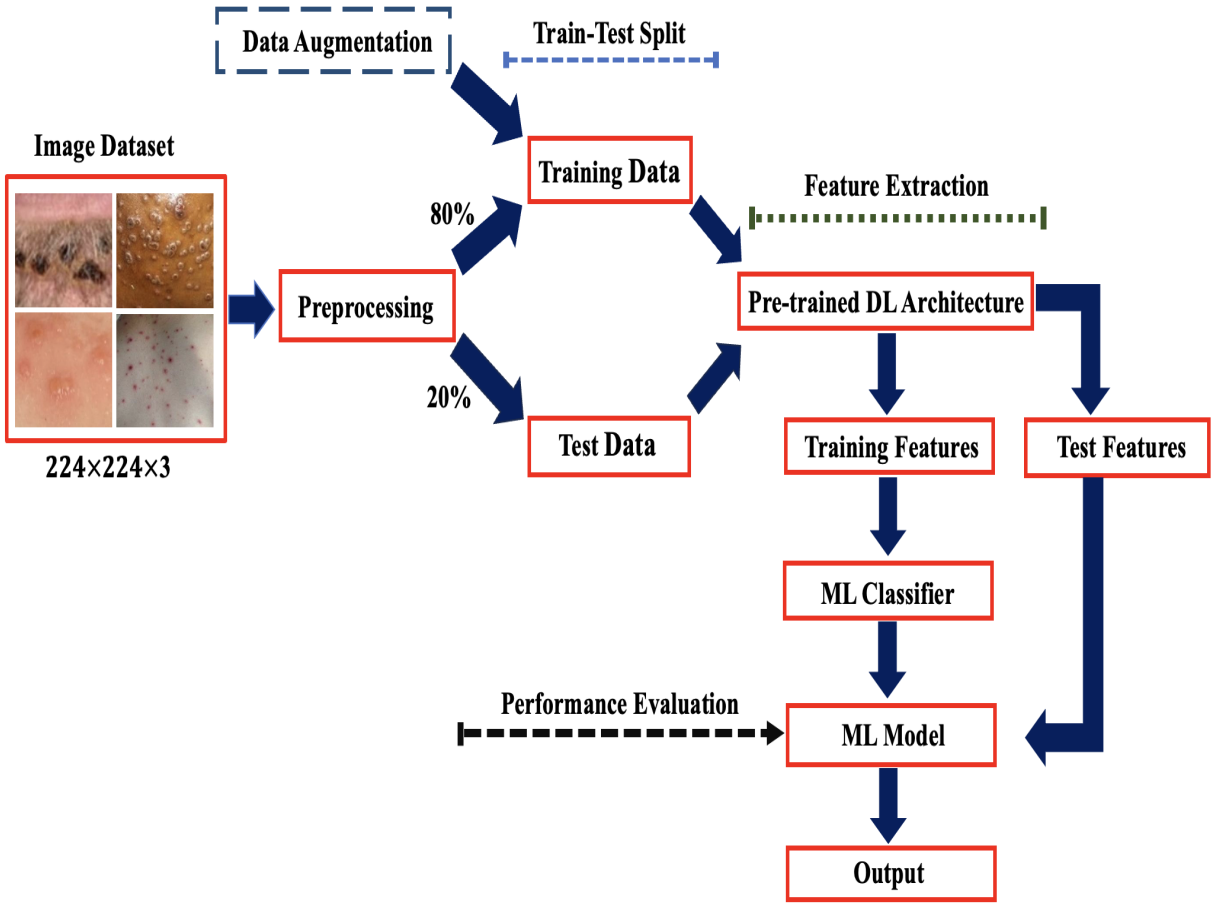

Figure 2: A hybrid deep learning framework for Mpox detection, combining a pre-trained deep learning architecture and machine learning classifier.

DenseNet-201, which contain 121, 169, and 201 layers, respectively.

3.4.2. Inception

This DL architecture uses an inception module, which allows Inception to learn global and local patterns and leverage its ability to learn features comprehensively. More specifically, the inception module applies convolutional filters of different sizes (1×1, 3×3, and 5×5) in parallel, which allows this architecture to learn features simultaneously at multiple scales [6]. In addition, 1×1 convolution reduces the dimensionality before applying the larger filters, which leverages the architecture's computational efficiency [6]. In this paper, we use InceptionV3 for feature extraction.

3.4.3. Inception-residual network

It ensembles the architecture of Inception and residual networks, incorporating residual connections within the inception block structure. Residual connections resolve the vanishing gradient issue and enhance the efficiency of this architecture in classification tasks, particularly for image data [6]. We use the Inception-ResNetV2 architecture for feature extraction in this paper.

3.4.4. Xception

This architecture is an improvement over Inception. One can consider this a bundle of depthwise separable convolution (DSC) with residual connection (RC), where DSC reduces the computational cost as well as improves memory usage by separating the learning of channel-wise and space-wise features, and RC solves the vanishing gradient issue [6].

3.4.5. MobileNet

It is an alternative to CNNs, computationally more efficient than CNNs and tailored for deployment on devices with limited computational capacities, such as smartphones [29]. Unlike CNNs, which use regular

Table 6
Summary of the deep learning architectures

Architectures	Activation function	Weights	Input image shape	Key Differences
DenseNet-201	ReLU	ImageNet	$224 \times 224 \times 3$	Consists of 201 layers
DenseNet-169	ReLU	ImageNet	$224 \times 224 \times 3$	4 dense blocks with 169 layers
DenseNet-121	ReLU	ImageNet	$224 \times 224 \times 3$	4 dense blocks with 121 layers
InceptionV3	ReLU	ImageNet	$224 \times 224 \times 3$	Uses kernels of different sizes
InceptionResNetV2	ReLU	ImageNet	$224 \times 224 \times 3$	Includes residual connections
Xception	ReLU	ImageNet	$224 \times 224 \times 3$	Applies depthwise separable convolutions
MobileNetV1	ReLU	ImageNet	$224 \times 224 \times 3$	Uses depthwise and pointwise convolutions
MobileNetV2	ReLU	ImageNet	$224 \times 224 \times 3$	Linear bottlenecks exist between layers

convolution operations where a filter processes all the input channels (e.g., red, green, blue for a color image) together, MobileNet uses a depthwise separable convolution operation [51]. This convolution operation comprises into two steps— Depthwise convolution (DC) and Pointwise convolution (PC) [51]. Initially DC, filters are applied to each input channel individually and reduce the number of parameters, which makes the computational process faster compared to the regular convolution operation. Then, PC combines the outputs from DC and transforms them into a single output [51]. In this paper, we implement MobileNetV1 and MobileNetV2 for feature extraction from images.

We utilize Python packages from the scikit-learn library to implement these deep learning architectures [55]. These models utilize the ReLU (Rectified Linear Unit) activation function, defined as:

$$ReLU(x) = \max(0, x), \tag{9}$$

and are trained on the ImageNet database. We provide a summary of these deep learning architectures in Table 6.

3.5. Machine learning classifiers

We have used the following feature set

$$\mathcal{A} = \{z_j^{\text{Train}} : j = 1, 2, \ldots, N\},\ z_j^{\text{Train}} \in \mathbb{R}^d$$

to train several ML classifiers. In the following subsections, we provide an overview of how these classifiers generally function.

3.5.1. Support vector machine

A crucial step in training a support vector machine (SVM) is to find a hyperplane, given by:

$$w^T z + b = 0, \tag{10}$$

so that the margin between the nearest data points of each class and the hyperplane is maximized. Here, $w \in \mathbb{R}^d$ denotes the weight, and $b \in \mathbb{R}$ denotes the bias in (10). It is important to note that to train an SVM; the labels must be mapped either -1 and 1 or 0 and 1 [63, 65]. For linearly separable data, to obtain the final classifier $x \mapsto \text{sign}\left(w^T z + b\right)$, one has to solve the following optimization problem [38]:

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 \tag{11}$$

$$\text{subject to, } \mathcal{Y}_j \left(w^T z_j + b\right) \geq 1, \tag{12}$$

where $j \in \{1, 2, \ldots, N\}$. Note that $\text{sign}\left(w^T z + b\right) = 1$ if $w^T z + b > 0$, and 0 otherwise.

In this paper, to train SVM, we have mapped the labels associated with each feature into 0 and 1.

3.5.2. Random forest

It performs classification task by ensembling predictions made by multiple decision trees based on majority voting strategy [12]. Suppose there are ℓ decision trees $T_1, T_2, \ldots, T_\ell$ in the forest. These trees are also known as "base learner" [20]. Initially, it trains each decision tree on a subset drawn from the training feature space using bootstrap sampling with replacement. Each of these trees make a prediction for a given input z, then final prediction is made as [20]:

$$\hat{\mathcal{Y}} = \underset{\mathcal{Y} \in C}{\text{argmax}} \sum_{i=1}^{\ell} \mathbb{I}\left(\mathcal{Y} = \hat{T}_\ell(z)\right), \tag{13}$$

where $\mathbb{I}$ denotes the indicator function, $\hat{T}_\ell$ denotes the ℓ-th fitted tree at z, and C is the set of all possible values of $\mathcal{Y}$.

3.5.3. Logistic regression

For any probability of success p expressed in terms of odds as [41, 52]:

$$\frac{p}{1-p} = \exp\left(\beta_0 + \beta^t z\right), \tag{14}$$

the Logisitic regression (LR) is defined as follows:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta^t z. \tag{15}$$

Here $\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$ is the coefficient vector of z and β_0 is a constant.

3.5.4. Decision tree

In general, a Decision tree (DT) is trained by deploying a continuously growing binary decision tree in training features z_j^{Train}, $j = 1, 2, \ldots, N$. It starts from a root node and ends with the decision outputs known as the leaf node [21, 64].

3.5.5. K-nearest neighborhood

The K-Nearest Neighbor (K-NN) classifier predicts the label of a target point z^* based on the patterns that are nearest to z^*. To identify these patterns, the classifier computes the similarity between the target point z^* and the input features using distance metrics such as the Euclidean distance or the Minkowski distance [57, 42]. For binary classification, K-NN is defined as [42]:

$$\hat{\mathcal{Y}} = \begin{cases} 1, & \text{if } \sum_{i \in N_K(z^*)} \mathcal{Y}_i \geq 0, \\ -1, & \text{if } \sum_{i \in N_K(z^*)} \mathcal{Y}_i < 0, \end{cases} \tag{16}$$

where K denotes the neighborhood size, and $N_K(z^*)$ denotes the set of indices of the K nearest neighbor. In this paper, we consider $K = 5$.

3.5.6. Naïve bayes classifier

It assumes that the features are conditionally independent given any target label [66]. In other words, it considers the contribution of each feature independently. For binary classification, it predicts a label as follows [66]:

$$\hat{\mathcal{Y}} = \underset{k \in \{0,1\}}{\arg\max}\, \mathbb{P}(\mathcal{Y} = k) \prod_{j=1}^{d} \mathbb{P}\left(z_j | \mathcal{Y} = k\right). \tag{17}$$

In this paper, for each z_j, we have considered that the likelihood function $\mathbb{P}\left(z_j | \mathcal{Y} = k\right)$ is normally distributed with mean μ_{jk} and variance σ_i.

3.5.7. Adaptive boosting

Adaptive boosting (AdaBoost) combines weak classifiers $T_1, T_2, \ldots, T_m$ and creates a strong classifier as follows:

$$\mathcal{Y} = \sum_{i=1}^{m} w_i T_i(z), \tag{18}$$

where m denotes the total number of classifiers, and w_i is the weight assigned to the weak classifier T_i (see [26]).

3.5.8. Extreme gradient boosting (XGBoost)

It is an improvement over Gradient boosting (GB), which forms a model for prediction by ensembling multiple decision trees where each successive tree depends upon the previous tree and tries to fix errors made by the preceding ones [17]. One of the important features of XGBoost is it prevents overfitting by regularizing the loss function as follows:

$$\sum_{i=1}^{n} \mathcal{L}\left(\mathcal{Y}_i, \hat{\mathcal{Y}}_i\right) + \mathcal{R}, \tag{19}$$

where $\hat{\mathcal{Y}}_i$ denotes the predicted observations, $\mathcal{L}$ denotes the loss function, and $\mathcal{R}$ is the regularization term, given by:

$$\mathcal{R} = \gamma t + \frac{1}{2}\lambda \sum_{j=1}^{t} w_j^2. \tag{20}$$

Here λ denotes the regularized parameter to scale the penalty, γ the minimum loss needed to further partition the leaf node, w denotes the weight of the leafs, t is the number of leaves in the tree.

3.5.9. Light gradient boosting (LightGBM)

It was built at the top of the Gradient boosting framework for classification and other machine-learning tasks. It was developed by Microsoft [37]. Its architecture includes histogram based learning, which reduces memory usage and accelerates computational speed, leaf-wise tree growth, Gradient based one-sided sampling (GOSS), exclusive feature bundling, which groups categorical features with common values, parallel and GPU learning, and regularization to prevent overfitting such as L_1 or L_2 regularization [37].

Besides AdaBoost, XGBoost, and LightGBM, there are other gradient boosting methods, such as CatBoost, which we found to be more time-consuming to train compared to the others. As a result, we explored computationally efficient gradient boosting classifiers.

We implemented these classifiers using Python packages and modules provided by the Scikit-learn library [55]. We summarize the hyperparameters of these classifiers, along with their optimal values, in Table 7. We optimized these hyperparameters through the grid search algorithm.

3.6. Evaluation metrics

In this section, we present evaluation metrics to evaluate hybrid DL models. Our goal is to classify Mpox and other conditions from image data. To formalize the evaluation process, we construct the following

Table 7
Summary of hyperparameters and optimal values for machine learning classification models

Models	Tuning Parameters	Best Parameters	Description
SVM	$C = [10^{-1}, 10^{3}]$, $\gamma = [10^{-4}, 1]$, Kernel = 'Linear'	$C = 0.1$, $\gamma = 1.0$	• C, regularization parameter. • γ controls a training example's influence.
RF	$\mu = \{3, 5, 10, \infty\}$, $n = \{10, 100, 200\}$, $\alpha = \{1, 3, 5, 7\}$, $\ell = \{1, 2, 3\}$, $m = \{2, 3\}$	$\mu = 3$, $n = 100$, $\alpha = 5$, $\ell = 1$, $m = 3$	• μ, maximum depth of the decision trees. • n, number of trees in the forest. • α limits the features used for node splits. • ℓ sets the minimum samples per leaf. • m sets the minimum samples to split a node.
LR	$C = [10^{-3}, 10^{3}]$, Penalty $\in \{\ell_1, \ell_2\}$	$C = 0.1$, Penalty $= \ell_2$	• C, regularization parameter. • Penalty, specifies the type of regularization applied.
DT	$f = \{$'auto', 'sqrt', 'log2'$\}$, $\overline{\alpha} = [10^{-2}, 10^{-1}]$, $\mu = \{5, 6, 7, 8, 9\}$, $\mathcal{C} = \{$'gini', 'entropy'$\}$	$f =$ 'log2', $\overline{\alpha} = 0.01$, $\mu = 7$, $\mathcal{C} =$ 'entropy'	• f, number of features to consider when splitting a node. • $\overline{\alpha}$, controls tree pruning. • μ, maximum depth of the tree. • $\mathcal{C}$, function to measure the quality of a split.
K-NN	$N = \{5, 7, 9, 11, 13, 15\}$, $W = \{$'uniform', 'distance'$\}$	$N = 5$, $W =$ 'uniform'	• N, number of nearest neighbors. • W denotes weight.
NB	$\delta =$ LogSpace$[10^{-9}, 1]$	$\delta = 0.01874$	• δ, smoothing parameter.
XGBoost	$\mu = \{3, 5, 7\}$, $r = \{0.1, 0.01, 0.001\}$, $J = \{0.5, 0.7, 1\}$	$\mu = 3$, $r = 0.1$, $J = 0.5$	• μ, maximum depth. • r, learning rate. • J, fraction of the training data to use for each boosting round.
AdaBoost	$n = \{50, 100, 150\}$, $r = \{0.01, 0.1, 1.0\}$	$n = 3$, $r = 0.1$	• n, number of trees in the forest. • r, learning rate.
LightGBM	$\lambda = \{31, 63, 127\}$, $\mu = \{-1, 3, 5\}$, $J = \{0.8, 1.0\}$, $\eta = \{0.8, 1.0\}$	$\lambda = 31$, $\mu = 3$, $J = 0.8$, $\eta = 0.8$	• λ, number of leaves. • μ, maximum depth. • J, fraction of the training data to use for each boosting round. • η, fraction of features to be randomly selected for each tree

hypotheses:

H_0 : The observation belongs to the "Others" class
H_A : The observation belongs to the "Mpox" class

where H_0 denotes the null hypothesis and H_A is the alternative hypothesis. Based on the predictions made by a hybrid DL model, the possible outcomes of this hypothesis test are:

i) True positive (TP): observation is correctly predicted as Mpox. In other words, H_0 is rejected.

ii) True negative (TN): H_0 is accepted, that is observation is classified as others.

iii) False positive (FP): observation is incorrectly classified as Mpox when it is actually others, which means that the H_0 is rejected when it is true.

iv) False negative (FN): observation is incorrectly classified as others when it is actually Mpox. In other words, H_0 is accepted when it is false.

These outcomes can be represented in a matrix form, commonly referred to as the confusion matrix (CM), as follows:

$$CM = \begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix}$$

We have computed accuracy, precision, recall, F1 score, weighted precision, weighted recall, weighted F1 score, and Matthews Correlation Coefficient (MCC) using these outcomes. In Table 8, we present the evaluation metrics in detail.

Table 8
Evaluation metrics

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall (Sensitivity)	$\frac{TP}{TP+FN}$
F1 Score	$2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$
Weighted Precision	$\frac{\sum_{i=1}^{k} w_i \times \text{Precision}_i}{\sum_{i=1}^{k} w_i}$
Weighted Recall	$\frac{\sum_{i=1}^{k} w_i \times \text{Recall}_i}{\sum_{i=1}^{k} w_i}$
Weighted F1 Score	$\frac{\sum_{i=1}^{k} w_i \times \text{F1}_i}{\sum_{i=1}^{k} w_i}$
MCC score	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

In addition to these evaluation metrics, we have also computed the area under the receiver operating characteristic curve (ROC). ROC plots the sensitivity (true positive rate) against the false positive rate (FPR). FPR is given by:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{21}$$

For n points on the ROC curve, area under the curve (AUC) can be computed using Trapezoidal rule approximation as follows:

$$\text{AUC} \approx \sum_{i=1}^{n-1} \left(\text{FPR}_{i+1} - \text{FPR}_i\right) \frac{\text{TPR}_{i+1} + \text{TPR}_i}{2}. \tag{22}$$

4. Results and discussion

Lack of skin lesion images and computational complexity are two major challenges in developing deep learning frameworks for Mpox detection. The objective of our hybrid deep learning framework is to detect Mpox using a small dataset while optimizing memory usage and computational time.

As an initial step to ensure model robustness, we performed data augmentation in the training dataset S_{Train}. To address that data augmentation is not going to introduce any bias, we performed KL divergence test between original and augmented Mpox images as well as original and augmented other images. We present these results in Figure 3. Furthermore, we have comprehensively analyzed this framework using several hybrid DL models (see Table 9, Table 10, and Table 11).

The hybrid models listed in Table 9 combine different DenseNet architectures (DenseNet-201, DenseNet-169, and DensNet-121) with ML classifiers. Notably, the hybrid model consists of DenseNet-201, and LR (D201LR) obtained an accuracy of 85.11%, outperforming other models that incorporated DenseNet-201 as a feature extraction tool. This model identified 18 true positive Mpox images out of 21 Mpox cases reported in the test dataset (see Figure 4a). On the other hand, DenseNet-201 combined with K-NN (D201KNN) achieved the highest precision (90%) among all DenseNet-201 incorporated hybrid models, highlighting its ability to minimize false positive cases.

The highest overall accuracy, 87.23%, was achieved by the model consisting of DenseNet-169 with LR (D169LR). While it outperforms other hybrid models, its precision is significantly small (82.61%) compared to D201KNN. It shows that D169LR had a slightly higher rate of predicting false positive cases.

DenseNet-169 combined with ML classifier Adaboost (D169Adaboost) is another model which attained the same accuracy as D169LR (87.23%). Other evaluation metrics of this model, such as weighted precision (87.39%), weighted recall (87.23%), F1-score (85%), and weighted F1-score (87.15%) are comparable to D169LR, although its precision 89.47% is higher than D169LR (82.61 %), which indicates its strength while minimizing false positive cases is essential.

DenseNet-169 integrating with SVM (D169SVM) is another noteworthy hybrid model for its highest recall rate, 90.48%, the same as D169LR. Hybird Models,

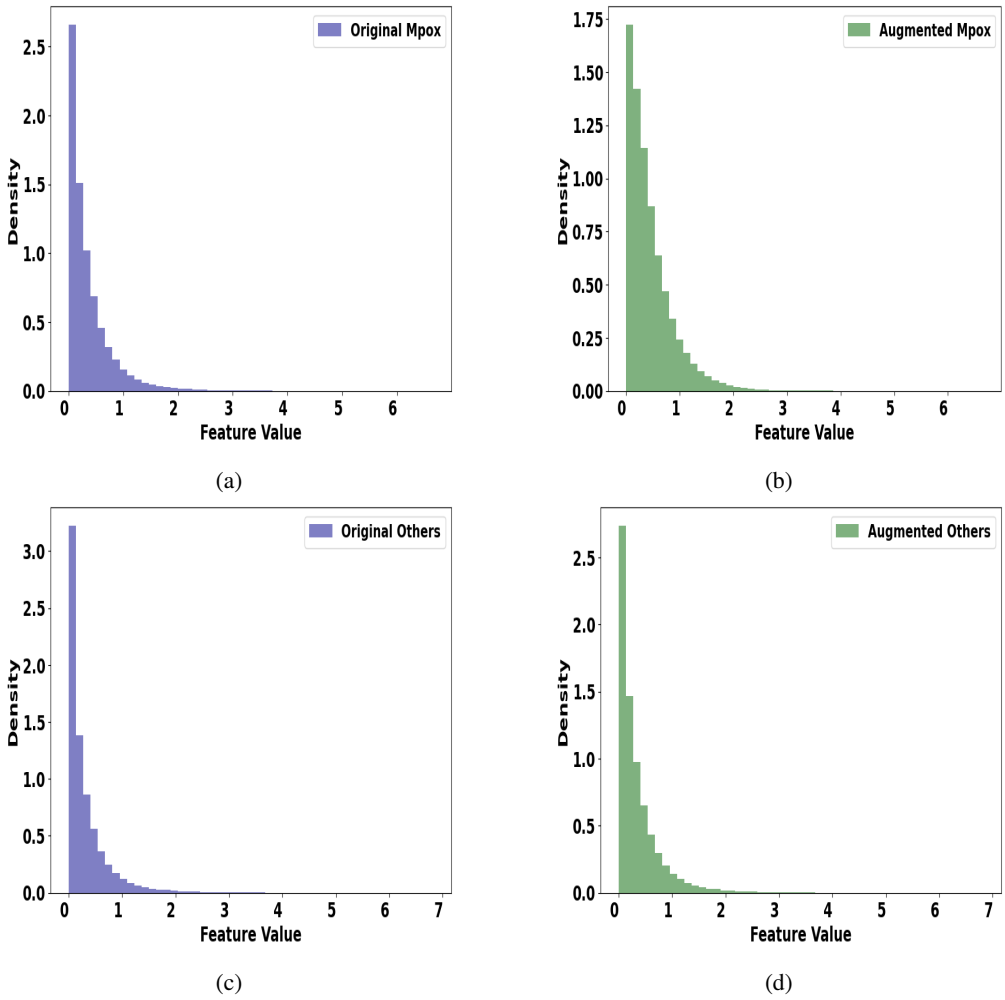

Figure 3: The Kullback-Leibler (KL) divergence analysis compares the distributions of original and augmented images. Figures 3a and 3b depict the distribution of original and augmented Mpox images. Both of these two figures exhibit a right-skewed distribution, indicating that augmentation has preserved the original data distribution. The KL divergence score for Mpox images is 0.0505, indicating no significant difference between the distributions. Similarly, Figures 3c and 3d illustrate the distribution of other images, with a KL divergence score of 0.0101, confirming that the augmentation process did not introduce any statistically significant differences.

DenseNet-121 with LR (D121LR) and DenseNet-121 with LightGBM (D121LightGBM) both performed equally in terms of every metric for Mpox prediction (see Table 9).

However, D169LR outperformed other models in MCC score (Table 9). Note that, MCC score provides valuable information about the model performance, most importantly when the dataset is imbalanced. D169LR provided a score of 0.75, which highlights its strength in Mpox prediction. While D169Adaboost achieved a very close sore (0.74) to D169LR, performance of other competing models are significantly low. From the models enlisted in Table 9, we have selected three best performing models based on their MCC scores.

In Figure 4, we incorporated confusion matrices and ROC curves of these models. Since D121SVM and D121LightGBM performed equally in each evaluation metric, we only considered one of these two models in Figure 4. Confusion matrices allowed us to distinguish between positive and negative cases with accuracy. For example, from figure 4b, one can see that D169LR has predicted 19 true Mpox images which are clearly better than D201LR and D121lightGBM. Along with these metrics, we investigated ROC curves for these models to assess model performance.

AUC of D161LR reflects its efficiency in Mpox identification. AUC of D161LR is 0.88, which indicates that the model has an 88% probability of ranking a randomly chosen positive instance higher than a randomly chosen negative instance. In comparison, D201LR has an AUC of 0.85, and D121LightGBM has an AUC of 0.83, both of which are significantly lower than D161LR.

Although DenseNet architectures are very efficient feature extraction tools, these are computationally demanding in terms of computational resources. Therefore, we further investigated other DL architectures such as InceptionV3, InceptionResNetV2, Xception, MobileNetV1, and MobileNetV2.

In Table 10, we have constructed a few more hybrid models by integrating InceptionV3, InceptionResNetV2, and Xception architecture as a feature extraction tool. Out of all the models developed based on InceptionV3, InceptionV3 with AdaBoost (IV3AdaBoost) obtained the highest accuracy (87.23%). Although IV3XGBoost (InceptionV3 with XGBoost) outperformed IV3AdaBoost in sensitivity (95.24%), its MCC score (0.72) and other metric scores are significantly lower than IV3AdaBoost. IV3AdaBoost and D169LR performed equally to classify Mpox and other images since both models scored equal values in every evaluation metric.

InceptionResNetV2 with RF (IRV2LR) performed better in every metric compared to other models, which are based on InceptionResNetV2 architecture. Even though IRV2LR and D201LR have equal accuracy (85.11%) and MCC score (0.70), their precision, weighted precision, recall, weighted recall, F1 score, and weighted F1 score are significantly different.

However, AUC of D201LR (0.85) is slightly higher than IRV2LR (0.84). Among hybrid models integrating Xception as a feature extraction tool, XcepSVM (Xception with SVM) and XecpLR (Xeception with LR) performed better than others. However, its accuracy (78.72%) is significantly low compared to D169LR and IRV2LR. XcepSVM and XecpLR identified 13 images as Mpox (see Figure 5c).

Moreover, AUC of IV3AdaBoost is 0.88, which indicates that this model has 88% probability of identifying randomly chosen positive cases, which is 4% higher than IRV2LR and 11% higher than XcepSVM (see Figures 5d, 5e, and 5f). Since XcepSVM and XcepLR performed equally, we incorporated only the confusion matrix and ROC curve for XcepSVM (Figure 5c and 5f).

In Table 11, we have more hybrid models that were developed by combining MobileNet architectures and ML classifiers. MobileNetV1 and MobileNetV2 were used to create these models. MV1LR (MobileNetV1 with LR) obtained 80.85% accuracy, which indicates its efficiency in Mpox prediction. In addition to that, its other evaluation metrics, such as precision (83.33%), are also significantly higher than other models that were developed based on MobileNetV1. Although its sensitivity or recall (71.43%) is lower than the model MV1LightGBM (MobileNetV1 with LightGBM), which is 76.19%, its MCC score of 0.61 outperformed other models. It identified 15 Mpox images and 23 other images (see Figure 6a).

MV2LightGBM achieved an accuracy of 91.49%, the highest among the models based on the MobileNetV2 architecture. Although its precision score (86.96%) is slightly lower than that of MV2AdaBoost (MobileNetV2 with AdaBoost) (88.89%), it outperformed other models in terms of the MCC score (0.83). Furthermore, MV2LightGBM attained the highest accuracy among all the models listed in Table 9, Table 10, and Table 11.

86.96% Precision score of MV2LightGBM indicates its ability to identify true positive cases of Mpox, which is higher than MV1LR. Furthermore, its weighted precision of 91.87% shows an overall scenario of how MV2LightGBM is suitable for binary classification, especially for an imbalanced dataset. In addition to that, its 91.87% recall and 95.24% weighted recall highlighted the robustness of this model in detecting true positive cases out of all true positive cases. More specifically, one can see that this model identified 20 Mpox images out of 21 (Figure 6b), which is the highest number of predictions over all other models.

Next, the 90.91% F1 score and the 91.51% weighted F1 score of MV2LightGBM highlighted the strength of this model in minimizing false positive and false negative cases. In other words, it can correctly identify 90.91% of the true positive instances while keeping the false positive and false negative cases low.

Furthermore, MV2LightGBM outperformed all the models in terms of AUC score, demonstrating its superior ability to distinguish between classes. Specifically, MV2LightGBM achieved an AUC of 0.92, which is significantly higher compared to several other models. Its AUC score is 15% higher than XcepSVM, 9% higher than D121LightGBM, 8% higher than MV1LR and IRV2LR. Moreover, its AUC is 4% higher than D169LR and IV3AdaBoost. These results along with its MCC score (0.83) established this model as an efficient robust model for Mpox detection.

Table 9
Hybrid DL models combining DenseNet architectures and ML Classifiers

	Classifiers	Accuracy	Precision	Weighted Precision	Recall (Sensitivity)	Weighted Recall	F1-score	Weighted F1 score	MCC score
	SVM	82.98	78.26	83.37	85.71	82.98	81.82	83.03	0.66
	RF	80.85	83.33	81.11	71.43	80.85	76.92	80.64	0.61
	LR	**85.11**	**81.82**	**85.24**	**85.71**	**85.11**	**83.72**	**85.13**	**0.70**
	DT	51.06	46.15	52.23	57.14	51.06	51.06	51.06	0.03
DenseNet-201	K-NN	72.34	90.00	77.59	42.86	72.34	58.06	69.85	0.47
	NB	72.34	66.67	73.08	76.19	72.34	71.11	72.42	0.45
	XGBoost	74.47	76.47	74.74	61.90	74.47	68.42	74.04	0.48
	AdaBoost	78.72	73.91	79.12	80.95	78.72	77.27	78.78	0.58
	LightGBM	76.60	77.78	76.72	66.67	76.60	71.79	76.33	0.52
	SVM	85.11	79.17	85.88	90.48	85.11	84.44	85.15	0.71
	RF	76.60	75.00	76.54	71.43	76.60	73.17	76.53	0.52
	LR	**87.23**	**82.61**	**87.62**	**90.48**	**87.23**	**86.36**	**87.27**	**0.75**
	DT	53.19	48.00	54.14	57.14	53.19	52.17	53.28	0.07
DenseNet-169	K-NN	65.96	85.71	72.87	28.57	65.96	42.86	61.06	0.35
	NB	68.09	63.64	68.26	66.67	68.09	65.12	68.14	0.36
	XGBoost	76.60	72.73	76.75	76.19	76.60	74.42	76.64	0.53
	AdaBoost	87.23	89.47	87.39	80.95	87.23	85.00	87.15	0.74
	LightGBM	78.72	78.95	78.74	71.43	78.72	75.00	78.59	0.57
	SVM	80.85	77.27	80.99	80.95	80.85	79.07	80.89	0.61
	RF	78.72	76.19	78.72	76.19	78.72	76.19	78.72	0.57
	LR	**82.98**	**80.95**	**82.98**	**80.95**	**82.98**	**80.95**	**82.98**	**0.66**
	DT	63.83	58.33	64.55	66.67	63.83	62.22	63.93	0.28
DenseNet-121	K-NN	63.83	83.33	70.97	23.81	63.83	37.04	57.83	0.30
	NB	74.47	69.57	74.88	76.19	74.47	72.73	74.54	0.49
	XGBoost	74.47	68.00	75.64	80.95	74.47	73.91	74.51	0.50
	AdaBoost	74.47	69.57	74.88	76.19	74.47	72.73	74.54	0.49
	LightGBM	**82.98**	**80.95**	**82.98**	**80.95**	**82.98**	**80.95**	**82.98**	**0.66**

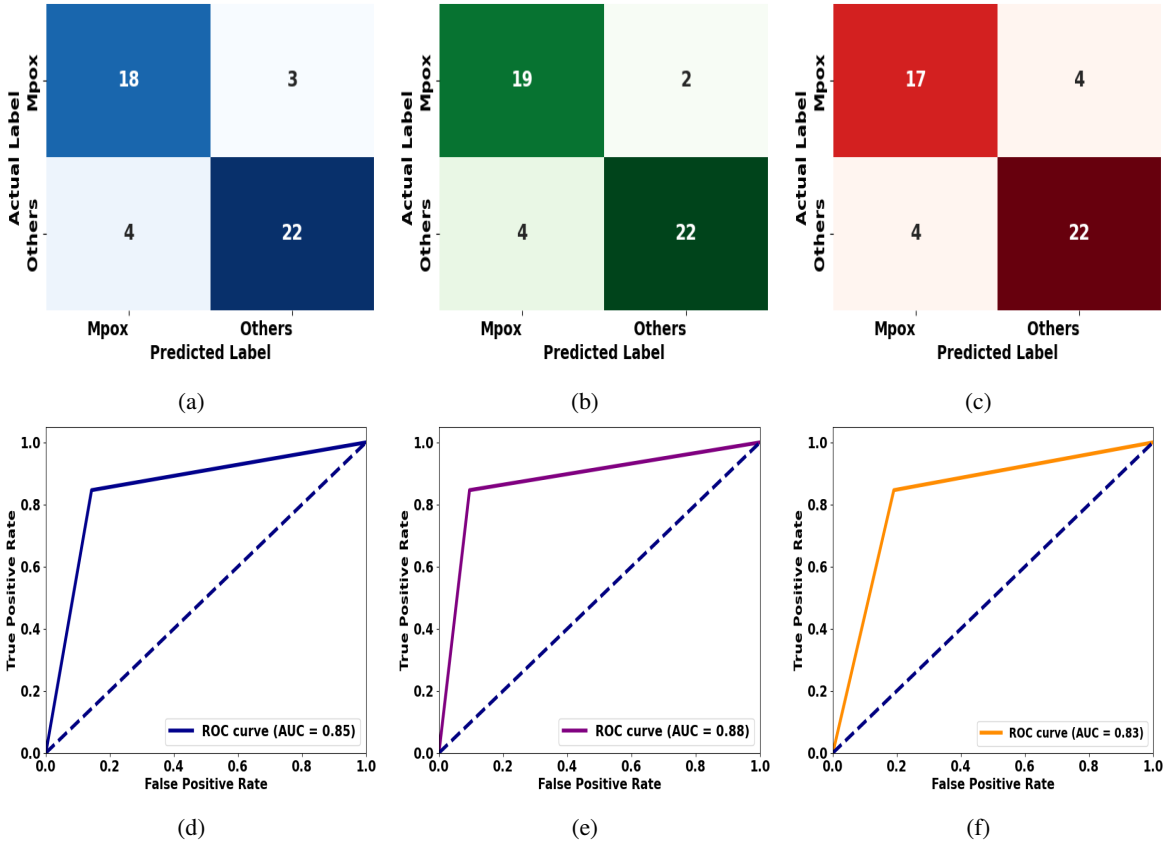

Figure 4: Confusion matrices and ROC curves for the selected models, incorporating DenseNet-201, DenseNet-169, and DenseNet-121 deep learning architectures as feature extraction tools from the image data. These models are selected based on their MCC score from each category listed in Table 9. Figure 4a, 4b and 4c are the confusion matrices for D201LR, D169LR, and D121LightGBM respectively, and 4d, 4e, and 4f illustrate their corresponding ROC curves. Confusion matrices display the counts of true positives, false negatives, false positives, and true negatives identified by these models. AUC scores measure the overall performance of these models in distinguishing between positive and negative classes by calculating the area under the ROC curve. A higher AUC indicates better discrimination between the two classes.

Table 10
Hybrid DL models combining InceptionV3, InceptionResNetV2, and Xception architectures with ML classifiers

	Classifiers	Accuracy	Precision	Weighted Precision	Recall (Sensitivity)	Weighted Recall	F1-score	Weighted F1 score	MCC score
	SVM	80.85	77.27	80.99	80.95	80.85	79.07	80.89	0.61
	RF	82.98	78.26	83.37	85.71	82.98	81.82	83.03	0.66
	LR	82.98	80.95	82.98	80.95	82.98	80.95	82.98	0.66
	DT	78.72	73.91	79.12	80.95	78.72	77.27	78.78	0.58
InceptionV3	K-NN	63.83	70.00	65.66	33.33	63.83	45.16	60.57	0.26
	NB	74.47	71.43	74.47	71.43	74.47	71.43	74.47	0.48
	XGBoost	85.11	76.92	87.05	95.24	85.11	85.11	85.11	0.72
	AdaBoost	**87.23**	**82.61**	**87.62**	**90.48**	**87.23**	**86.36**	**87.27**	**0.75**
	LightGBM	82.98	80.95	82.98	80.95	82.98	80.95	82.98	0.66
	SVM	78.72	82.35	79.21	66.67	78.72	73.68	78.36	0.57
	RF	**85.11**	**88.89**	**85.50**	**76.19**	**85.11**	**82.05**	**84.94**	**0.70**
	LR	78.72	82.35	79.21	66.67	78.72	73.68	78.36	0.57
	DT	68.08	66.67	67.94	57.14	68.09	61.54	67.73	0.35
InceptionResNetV2	K-NN	55.32	50.00	53.37	14.29	55.32	22.22	47.91	0.04
	NB	68.09	66.67	67.94	57.14	68.09	61.54	67.73	0.35
	XGBoost	78.72	73.91	79.12	80.95	78.72	77.27	78.78	0.58
	AdaBoost	76.60	77.78	76.72	66.67	76.60	71.79	76.33	0.52
	LightGBM	76.60	77.78	76.72	66.67	76.60	71.79	76.33	0.52
	SVM	**78.72**	**86.67**	**80.21**	**61.90**	**78.72**	**72.22**	**78.05**	**0.58**
	RF	70.21	73.33	70.80	52.38	70.21	61.11	69.27	0.39
	LR	**78.72**	**86.67**	**80.21**	**61.90**	**78.72**	**72.22**	**78.05**	**0.58**
	DT	65.96	63.16	65.76	57.14	65.96	60.00	65.74	0.31
Xception	K-NN	59.57	60.00	59.70	28.57	59.57	38.71	55.93	0.16
	NB	65.96	57.14	76.24	95.24	65.96	71.43	63.94	0.43
	XGBoost	70.21	73.33	70.80	52.38	70.21	61.11	69.27	0.39
	AdaBoost	76.60	81.25	77.35	61.90	76.60	70.27	76.04	0.53
	LightGBM	76.60	81.25	77.35	61.90	76.60	70.27	76.04	0.53

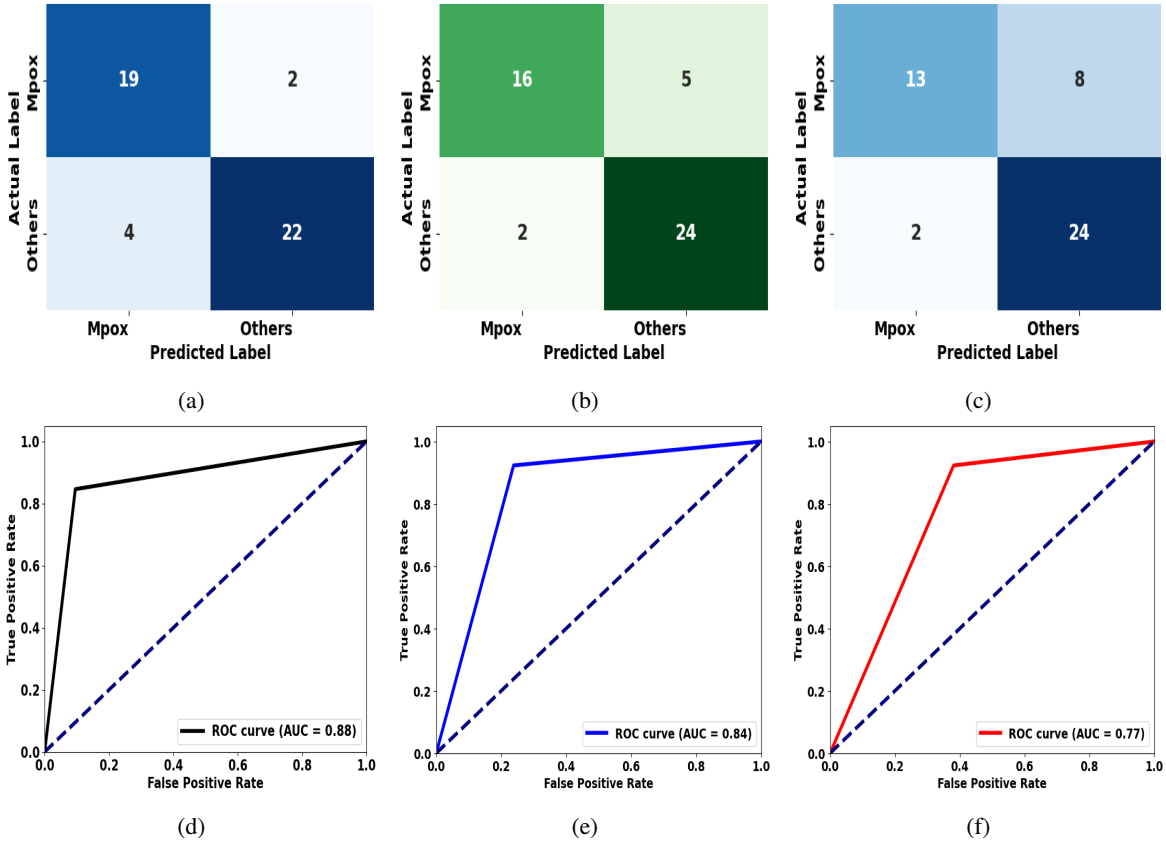

Figure 5: Performance evaluation of the models IV3AdaBoost (5a, 5d), IRV2LR (5b, 5e), and XcepSVM (5c, 5f) based on confusion matrices and ROC curves. These models are selected based on their MCC score from the list of models generated incorporating InceptionV3, InceptionResNetV2, and Xception as feature extraction tools (see Table 10). The AUC score and the number of true positive images identified by IV3AdaBoost highlight its efficiency in Mpox detection compared to the other two models.

Table 11
Hybrid DL models combining MobileNetV1 and MobileNetV2 architectures with ML classifiers

	Classifiers	Accuracy	Precision	Weighted Precision	Recall (Sensitivity)	Weighted Recall	F1-score	Weighted F1 score	MCC score
MobileNetV1	SVM	78.72	78.95	78.74	71.43	78.72	75.00	78.59	0.57
	RF	78.72	73.91	79.12	80.95	78.72	77.27	78.78	0.58
	LR	**80.85**	**83.33**	**81.11**	**71.43**	**80.85**	**76.92**	**80.64**	**0.61**
	DT	63.83	59.09	64.02	61.90	63.83	60.47	63.90	0.27
	K-NN	57.45	100	75.95	4.76	57.45	9.09	44.01	0.16
	NB	51.06	47.73	76.64	100	51.06	64.62	40.32	0.23
	XGBoost	78.72	76.19	78.72	76.19	78.72	76.19	78.72	0.57
	AdaBoost	78.72	78.95	78.74	71.43	78.72	75.00	78.59	0.57
	LightGBM	78.72	76.19	78.72	76.19	78.72	76.19	78.72	0.57
MobileNetV2	SVM	82.98	84.21	83.07	76.19	82.98	80.00	82.87	0.65
	RF	80.85	80.00	80.82	76.19	80.85	78.05	80.80	0.61
	LR	82.98	88.24	83.68	71.43	82.98	78.95	82.69	0.66
	DT	74.47	68.00	75.64	80.95	74.47	73.91	74.51	0.50
	K-NN	59.57	75.00	65.67	14.29	59.57	24.00	50.81	0.19
	NB	53.19	48.78	67.89	95.24	53.19	64.52	46.11	0.22
	XGBoost	85.11	85.00	85.10	80.95	85.11	82.93	85.07	0.70
	AdaBoost	85.11	88.89	85.50	76.19	85.11	82.05	84.94	0.70
	LightGBM	**91.49**	**86.96**	**91.87**	**95.24**	**91.49**	**90.91**	**91.51**	**0.83**

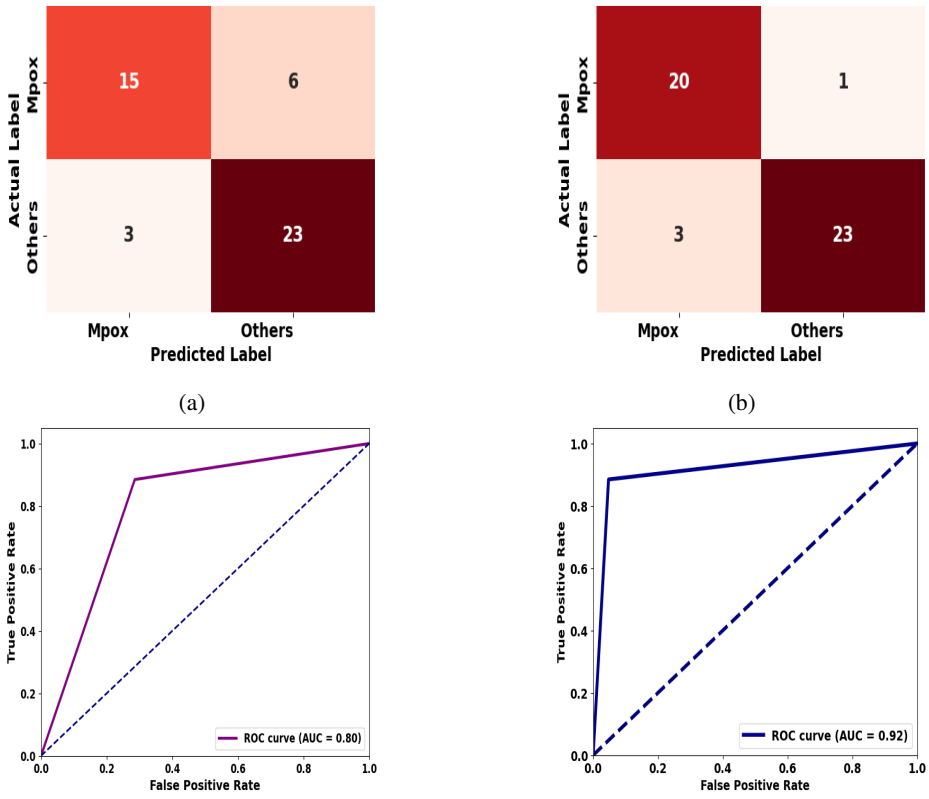

Figure 6: Comparison of the models constructed using MobileNetV1 and MobileNetV2 architectures. Figures 6a and 6c are the confusion matrix and ROC curve for MV1LR, respectively, while Figures 6b and 6d show the confusion matrix and ROC curve for MV2LightGBM. Figure 6b demonstrates that MV2LightGBM identifies the highest number of true positive cases compared to the other models. Additionally, its AUC score of 0.92 highlights its efficiency in distinguishing between positive and negative cases.

4.1. Comparison with previous studies

Through our analysis, MV2LightGBM emerged as the best-performing hybrid model, achieving the highest accuracy compared to all other evaluated hybrid models. This section compares this model with the different models previously published in the literature for Mpox detection.

In this paper, we have used the dataset created by [5]. However, MV2LightGBM, which obtained 91.49% accuracy, outperformed the model proposed by [5]. Note that the total augmented images reported in [5] is 2,562, which is much larger than our total augmented images.

Additionally, they did not conduct any qualitative or quantitative validation tests to ensure their models did not produce any biases, moreover they only computed accuracy, precision, recall and F1 scores. In contrast, we have performed a quantitative validation test on our augmented images and performed a comprehensive analysis by computing various evaluation metrics.

MV2LightGBM also outperformed the model proposed by [58] by a slight margin. Note that [58] utilized the same dataset, including augmented images from [5] and did not provide any evidence of quantitative or qualitative validation.

Furthermore, MV2LightGBM outperformed [62] not only in terms of accuracy but also across other evaluation metrics (see Table 12). [62] trained their models on a different dataset than ours, using a total of 1,754 images, of which 587 are Mpox. However, they did not compute any additional metrics beyond the four we reported in Table 12, despite their dataset being imbalanced.

Although MV2LightGBM outperformed the best-performing model, EfficientNetB3, proposed by [23] in terms of accuracy, [23] achieved higher precision

Table 12
Comparison of our best-performing model with previously published results

	Accuracy	Precision	Recall	F1 score
MV2LightGBM	91.49	86.96	95.24	90.91
[5]	82.96	87	83	84
[58]	91.11	90	90	90
[62]	87.13	85.44	85.47	85.40
[23]	87	92	87	90

than MV2LightGBM. It is important to note that their augmented dataset contains 847 images, of which 160 are Mpox. Moreover, they did not report any validation results to confirm that the augmented images are free from bias. However, MV2LightGBM achieved higher recall and F1 scores compared to [23].

It is important to note that, compared to us, all these prior studies had a larger amount of training data. We explored some DL architectures that contain over 200 layers, as well as architectures suitable for mobile devices, such as MobileNetV1 and MobileNetV2. Moreover, to train models, we did not have to utilize any high-performance computational resources, which highlights the efficiency of our proposed framework in a resource-limited setting.

5. Conclusion

A deep learning framework for Mpox detection, integrating pre-trained deep learning architectures and machine learning classifiers, was developed in this paper. This framework was designed as a tool for generating robust models when the amount of data is limited. The input data to the models are skin lesion images, which include Mpox and other skin disease images. To enhance model robustness, data augmentation was applied to the training data. Results from the Kullback-Leibler divergence analysis showed that data augmentation did not significantly alter the data distribution. Using various deep learning architectures, such as DenseNet, Inception, Xception, MobileNet along with nine machine learning classifiers a total of 72 models were trained, followed by a comprehensive analysis. To ensure that the models efficiently analyzed the imbalanced data, various evaluation metrics were computed. The results showed that MV2LightGBM achieved the highest accuracy of 91.49%. Since imbalanced data can lead to erroneous accuracy, MCC scores were given more priority over other metrics in this paper. The MCC results showed that MV2LightGBM achieved the highest score 0.83 over other models, which highlights its reliability in handling imbalanced data. Moreover, the AUC of MV2LightGBM (0.92) further confirms its strong classification capability. These findings demonstrate the potential of the framework for Mpox detection. Further research could explore extending this deep learning framework across diverse datasets in resource-limited settings.

Funding

The authors did not receive any funding for this research.

Competing Interests

The authors declare no competing interests.

Data availability

The data we have used to plot Figure 1c is available in Centers for Disease Control and Prevention and Our world data website.

References

[1] Adom-Konadu, A., Bonyah, E., Sackitey, A.L., Anokye, M., Asamoah, J.K.K., 2023. A fractional order Monkeypox model with protected travelers using the fixed point theorem and newton polynomial interpolation. Healthcare Analytics 3, 100191.

[2] Akilandasowmya, G., Nirmaladevi, G., Suganthi, S., Aishwariya, A., 2024. Skin cancer diagnosis: Leveraging deep hidden features and ensemble classifiers for early detection and classification. Biomedical Signal Processing and Control 88, 105306.

[3] Akinyemi, S.T., Idisi, I.O., Rabiu, M., Okeowo, V.I., Iheonu, N., Dansu, E.J., Abah, R.T., Mogbojuri, O.A., Audu, A.M., Yahaya, M.M., et al., 2023. A tale of two countries: optimal control and cost-effectiveness analysis of Monkeypox disease in germany and nigeria. Healthcare Analytics 4, 100258.

[4] Ali, M.S., Miah, M.S., Haque, J., Rahman, M.M., Islam, M.K., 2021. An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. Machine Learning with Applications 5, 100036.

[5] Ali, S.N., Ahmed, M.T., Paul, J., Jahan, T., Sani, S., Noor, N., Hasan, T., 2022. Monkeypox skin lesion detection using deep learning models: A feasibility study. arXiv preprint arXiv:2207.03342 .

[6] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts,

cnn architectures, challenges, applications, future directions. Journal of Big Data 8, 1–74.

[7] Amin, J., Sharif, M., Yasmin, M., Fernandes, S.L., 2020. A distinctive approach in brain tumor detection and classification using mri. Pattern Recognition Letters 139, 118–127.

[8] Arumugam, M., Thiyagarajan, A., Adhi, L., Alagar, S., 2024. Crossover smell agent optimized multilayer perceptron for precise brain tumor classification on MRI images. Expert Systems with Applications 238, 121453.

[9] Batiha, I.M., Abubaker, A.A., Jebril, I.H., Al-Shaikh, S.B., Matarneh, K., Almuzini, M., 2023. A mathematical study on a fractional-order SEIR Mpox model: analysis and vaccination influence. Algorithms 16, 418.

[10] Betti, M.I., Farrell, L., Heffernan, J., 2023. A pair formation model with recovery: Application to Mpox. Epidemics 44, 100693.

[11] Bhatele, K.R., Jha, A., Tiwari, D., Bhatele, M., Sharma, S., Mithora, M.R., Singhal, S., 2024. COVID-19 detection: A systematic review of machine and deep learning-based approaches utilizing chest x-rays and ct scans. Cognitive Computation 16, 1889–1926.

[12] Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

[13] Breman, J.G., Steniowski, M., Zanotto, E., Gromyko, A., Arita, I., et al., 1980. Human Monkeypox, 1970-79. Bulletin of the World Health Organization 58, 165.

[14] Brown, K., Leggat, P.A., 2016. Human Monkeypox: current state of knowledge and implications for the future. Tropical Medicine and Infectious Disease 1, 8.

[15] Centers for Disease Control and Prevention, 2024a. About Mpox (Monkeypox). URL: https://www.cdc.gov/poxvirus/mpox/about/index.html. accessed: 2024-09-14.

[16] Centers for Disease Control and Prevention, 2024b. Nndss - table 968: Weekly cases of notifiable diseases - 2024. https://wonder.cdc.gov/nndss/static/2024/36/2024-36-table968.html. Accessed: 2024-09-04.

[17] Chen, T., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2 1.

[18] Chicco, D., Jurman, G., 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics 21, 1–13.

[19] Cover, T.M., 1999. Elements of information theory. John Wiley & Sons.

[20] Cutler, A., Cutler, D.R., Stevens, J.R., 2012. Random forests. Ensemble machine learning: Methods and applications , 157–175.

[21] De Ville, B., 2013. Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics 5, 448–455.

[22] Dhillon, A., Verma, G.K., 2020. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence 9, 85–112.

[23] Dwivedi, M., Tiwari, R.G., Ujjwal, N., 2022. Deep learning methods for early detection of Monkeypox skin lesion, in: 2022 8th International Conference on Signal Processing and Communication (ICSC), IEEE. pp. 343–348.

[24] Foster, S.O., Brink, E.W., Hutchins, D.L., Pifer, J.M., Lourie, B., Moser, C.R., Cummings, E.C., Kuteyi, O., Eke, R.E., Titus, J., et al., 1972. Human Monkeypox. Bulletin of the World Health Organization 46, 569.

[25] Fredricks, D.N., Relman, D.A., 1999. Application of polymerase chain reaction to the diagnosis of infectious diseases. Clinical Infectious Diseases , 475–486.

[26] Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139.

[27] Gessain, A., Nakoune, E., Yazdanpanah, Y., 2022. Monkeypox. New England Journal of Medicine 387, 1783–1793.

[28] Hattaf, K., Dutta, H., 2020. Mathematical modelling and analysis of infectious diseases. volume 302. Springer.

[29] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861. URL: http://arxiv.org/abs/1704.04861, arXiv:1704.04861.

[30] Huang, K., Hussain, A., Wang, Q.F., Zhang, R., 2019. Deep learning: fundamentals, theory and applications. volume 2. Springer.

[31] Ianache, I., Skrzat-Klapaczynska, A., Jilich, D., Fleischhans, L., Gmizic, I., Ranin, J., Papadopoulos, A., Protopapas, K., Mulabdic, V., Lakatos, B., et al., 2024. Mpox across countries from central and eastern europe-2022 outbreak. Travel Medicine and Infectious Disease 59, 102719.

[32] Ibrahim, A.U., Ozsoz, M., Serte, S., Al-Turjman, F., Yakoi, P.S., 2024. Pneumonia classification using deep learning from chest X-ray images during COVID-19. Cognitive Computation 16, 1589–1601.

[33] Idisi, O.I., Yusuf, T.T., Adeniyi, E., Onifade, A.A., Oyebo, Y.T., Samuel, A.T., Kareem, L.A., 2023. A new compartmentalized epidemic model to analytically study the impact of awareness on the control and mitigation of the Monkeypox disease. Healthcare Analytics 4, 100267.

[34] Ismael, A.M., Şengür, A., 2021. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Systems with Applications 164, 114054.

[35] Jiang, Y., Chen, H., Loew, M., Ko, H., 2020. COVID-19 CT image synthesis with a conditional generative adversarial network. IEEE Journal of Biomedical and Health Informatics 25, 441–452.

[36] Kaftan, D., Kim, H.Y., Ko, C., Howard, J.S., Dalal, P., Yamamoto, N., Braithwaite, R.S., Bershteyn, A., 2024. Performance analysis of mathematical methods used to forecast the 2022 new york city Mpox outbreak. Journal of Medical Virology 96, e29791.

[37] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems 30.

[38] Kecman, V., 2005. Support vector machines–an introduction, in: Support vector machines: theory and applications. Springer, pp. 1–47.

[39] Keshavamurthy, R., Dixon, S., Pazdernik, K.T., Charles, L.E., 2022. Predicting infectious disease for biopreparedness and response: A systematic review of machine learning and deep learning approaches. One Health 15, 100439.

[40] Khan, S.U.R., Zhao, M., Asif, S., Chen, X., 2024. Hybridnet: A fusion of densenet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. International Journal of Imaging Systems and Technology 34, e22975.

[41] Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M., 2002. Logistic regression. Springer.

[42] Kramer, O., 2013. K-nearest neighbors. Dimensionality reduction with unsupervised nearest neighbors , 13–23.

[43] Ladnyj, I., Ziegler, P., Kima, E., 1972. A human infection caused by Monkeypox virus in basankusu territory, democratic republic of the congo. Bulletin of the World Health Organization 46, 593.

[44] Liu, T., Yang, S., Luo, B., Fan, X., Zhuang, Y., Gao, G.F., Bi, Y., Teng, Y., 2023. Anticipating the transmissibility of the 2022

Mpox outbreak. Journal of Medical Virology 95, e28683.
[45] Lu, C., Mandal, M., 2015. Automated analysis and diagnosis of skin melanoma on whole slide histopathological images. Pattern Recognition 48, 2738–2750.
[46] McCollum, A.M., Damon, I.K., 2014. Human Monkeypox. Clinical Infectious Diseases 58, 260–267.
[47] Meena, G., Mohbey, K.K., Kumar, S., 2024. Monkeypox recognition and prediction from visuals using deep transfer learning-based neural networks. Multimedia Tools and Applications , 1–25.
[48] Molla, J., Sekkak, I., Ortiz, A.M., Moyles, I., Nasri, B., 2023. Mathematical modeling of Mpox: A scoping review. One Health 16, 100540.
[49] Narudin, F.A., Feizollah, A., Anuar, N.B., Gani, A., 2016. Evaluation of machine learning classifiers for mobile malware detection. Soft Computing 20, 343–357.
[50] Nayak, T., Chadaga, K., Sampathila, N., Mayrose, H., Muralidhar Bairy, G., Prabhu, S., Katta, S.S., Umakanth, S., 2023. Detection of Monkeypox from skin lesion images using deep learning networks and explainable artificial intelligence. Applied Mathematics in Science and Engineering 31, 2225698.
[51] Nelson, M., Gailly, J.L., 1995. The data compression book 2nd edition. M & T Books, New York, NY .
[52] Nick, T.G., Campbell, K.M., 2007. Logistic regression. Topics in Biostatistics , 273–301.
[53] Organization, W.H., et al., 2022. Monkeypox https://www.who. int/news-room/fact-sheets/detail/monkeypox. Accessed 10th Jun .
[54] Peace, O.O., Godwin, O.A., Bolaji, B., 2024. A compartmental deterministic epidemiological model with non-linear differential equations for analyzing the co-infection dynamics between COVID-19, HIV, and Monkeypox diseases. Healthcare Analytics 5, 100311.
[55] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.
[56] Pereira, F., Mitchell, T., Botvinick, M., 2009. Machine learning classifiers and fMRI: a tutorial overview. Neuroimage 45, S199–S209.
[57] Peterson, L.E., 2009. K-nearest neighbor. Scholarpedia 4, 1883.
[58] Sahin, V.H., Oztel, I., Yolcu Oztel, G., 2022. Human Monkeypox classification from skin lesion images with deep pre-trained network using mobile application. Journal of Medical Systems 46, 79.
[59] Sailunaz, K., Özyer, T., Rokne, J., Alhajj, R., 2023. A survey of machine learning-based methods for COVID-19 medical image analysis. Medical & Biological Engineering & Computing 61, 1257–1297.
[60] Sharma, S., Guleria, K., 2024. A systematic literature review on deep learning approaches for pneumonia detection using chest X-ray images. Multimedia Tools and Applications 83, 24101–24151.
[61] Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6, 1–48.
[62] Sitaula, C., Shahi, T.B., 2022. Monkeypox virus detection using pre-trained deep learning-based approaches. Journal of Medical Systems 46, 78.
[63] Stitson, M., Weston, J., Gammerman, A., Vovk, V., Vapnik, V., 1996. Theory of support vector machines. University of London 117, 188–191.
[64] Suthaharan, S., Suthaharan, S., 2016a. Decision tree learning. Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning , 237–269.
[65] Suthaharan, S., Suthaharan, S., 2016b. Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning , 207–235.
[66] Szostak, R., 2012. The basic concepts classification, in: Categories, Contexts and Relations in Knowledge Organization, Ergon-Verlag. pp. 24–30.
[67] Uysal, F., 2023. Detection of Monkeypox disease from human skin images with a hybrid deep learning model. Diagnostics 13, 1772.
[68] Vaughan, A., Aarons, E., Astbury, J., Brooks, T., Chand, M., Flegg, P., Hardman, A., Harper, N., Jarvis, R., Mawdsley, S., et al., 2020. Human-to-human transmission of Monkeypox virus, united kingdom, october 2018. Emerging Infectious Diseases 26, 782.
[69] Yang, S., Rothman, R.E., 2004. Pcr-based diagnostics for infectious diseases: uses, limitations, and future applications in acute-care settings. The Lancet Infectious Diseases 4, 337–348.
[70] Yolcu Oztel, G., 2024. Vision transformer and cnn-based skin lesion analysis: classification of Monkeypox. Multimedia Tools and Applications 83, 71909–71923.
[71] Zhang, L., Huang, J., Yan, W., Zhao, Y., Wang, D., Chen, B., 2024. Global prediction for Mpox epidemic. Environmental Research 243, 117748.