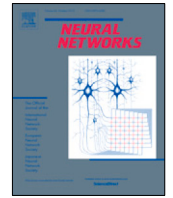




Elsevier has created a [Monkeypox Information Center](#) in response to the declared public health emergency of international concern, with free information in English on the monkeypox virus. The Monkeypox Information Center is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its monkeypox related research that is available on the Monkeypox Information Center - including this research content - immediately available in publicly funded repositories, with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the Monkeypox Information Center remains active.



MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification

Diponkor Bala^{a,f,1}, Md. Shamim Hossain^{b,1}, Mohammad Alamgir Hossain^a,
Md. Ibrahim Abdullah^a, Md. Mizanur Rahman^d, Balachandran Manavalan^f, Naijie Gu^b,
Mohammad S. Islam^{e,*}, Zhangjin Huang^{b,c,**}

^a Department of Computer Science and Engineering, Islamic University, Kushtia 7003, Bangladesh

^b School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, Anhui, 230026, China

^c USTC-Deqing Alpha Innovation Institute, Huzhou, 313299, China

^d School of Engineering, Design and Built Environment, Western Sydney University, Penrith, NSW 2751, Australia

^e School of Mechanical and Mechatronic Engineering, University of Technology Sydney (UTS), 15 Broadway, Ultimo, NSW 2007, Australia

^f Computational Biology and Bioinformatics Laboratory, Department of Integrative Biotechnology, College of Biotechnology and Bioengineering, Sungkyunkwan University, Suwon 16419, Gyeonggi-do, Republic of Korea

ARTICLE INFO

Article history:

Received 23 July 2022

Received in revised form 5 February 2023

Accepted 14 February 2023

Available online 22 February 2023

Keywords:

Monkeypox disease

Dataset

Machine learning

Deep learning

Convolutional neural network

Classification

ABSTRACT

The monkeypox virus poses a new pandemic threat while we are still recovering from COVID-19. Despite the fact that monkeypox is not as lethal and contagious as COVID-19, new patient cases are recorded every day. If preparations are not made, a global pandemic is likely. Deep learning (DL) techniques are now showing promise in medical imaging for figuring out what diseases a person has. The monkeypox virus-infected human skin and the region of the skin can be used to diagnose the monkeypox early because an image has been used to learn more about the disease. But there is still no reliable Monkeypox database that is available to the public that can be used to train and test DL models. As a result, it is essential to collect images of monkeypox patients. The “MSID” dataset, short form of “Monkeypox Skin Images Dataset”, which was developed for this research, is free to use and can be downloaded from the Mendeley Data database by anyone who wants to use it. DL models can be built and used with more confidence using the images in this dataset. These images come from a variety of open-source and online sources and can be used for research purposes without any restrictions. Furthermore, we proposed and evaluated a modified DenseNet-201 deep learning-based CNN model named MonkeyNet. Using the original and augmented datasets, this study suggested a deep convolutional neural network that was able to correctly identify monkeypox disease with an accuracy of 93.19% and 98.91% respectively. This implementation also shows the Grad-CAM which indicates the level of the model’s effectiveness and identifies the infected regions in each class image, which will help the clinicians. The proposed model will also help doctors make accurate early diagnoses of monkeypox disease and protect against the spread of the disease.

© 2023 Elsevier Ltd. All rights reserved.

* Corresponding author.

** Corresponding author at: School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, Anhui, 230026, China.

E-mail addresses: diponkor.b@gmail.com (D. Bala), shamim2@mail.ustc.edu.cn (M.S. Hossain), alomgir@cse.iu.ac.bd (M.A. Hossain), ibrahim@cse.iu.ac.bd (M.I. Abdullah), mdmizanur.rahman@westernsydney.edu.au (M.M. Rahman), bala2022@skku.edu (B. Manavalan), gunj@ustc.edu.cn (N. Gu), mohammadsaidul.islam@uts.edu.au (M.S. Islam), zhuang@ustc.edu.cn (Z. Huang).

¹ Equally contributed to this work.

1. Introduction

The very infrequent monkeypox disease is caused by a virus called the monkeypox virus. It belongs to the same family as the more well-known virus responsible for causing smallpox, and its name is orthopoxvirus (Lewin, 2010). In 1958, sick monkeys that had been sent from Singapore to a research center in Denmark provided the initial clues that led to the isolation and identification of the monkeypox virus (Kumar, Acharya, Gendelman, & Byrareddy, 2022). The number of monkeypox cases is going up, not just in Africa but also in other places where these illnesses have not been seen before (Duds et al., 2022). After being exposed to something, a person may not have symptoms for days or weeks. Early monkeypox symptoms include those similar to the

flu, such as fever, chills, headaches, aches, muscle pains, fatigue, and swollen lymph nodes (Fatima & Mandava, 2022). After a few days, a rash will typically begin to appear. The rash initially manifests itself as painful papules that are red and flat. These raised areas eventually develop into blisters, which are then filled with pus. The blisters harden and flake off in two to four weeks. The symptoms of monkeypox do not always appear in every person who has the disease (Adalja & Inglesby, 2022). The virus that causes monkeypox can be contracted by contacting an infected animal or person. There is a risk of transmission from person to person for monkeypox, despite the disease's rarity. Transmission from one person to another takes place when a person comes into contact with the sores, scabs, respiratory droplets, or oral fluids of an infected individual (Simpson et al., 2020). Expert healthcare professionals may initially suspect similar rash infections, such as measles or chickenpox, as the origin of their monkeypox because of how uncommon the condition is. But most of the time, swollen lymph nodes can tell the difference between monkeypox and other types of pox (Koenig, Bej, & Marty, 2022). The healthcare professionals will obtain a sample of tissue from someone with an active infection on one body in order to make a diagnosis of monkeypox. The sample is then sent to a laboratory for analysis using polymerase chain reaction (PCR) (Reed et al., 2004). This test has been known to be expensive and take a long time to return results. At this time, there is no antiviral therapy that will cure us enough to treat monkeypox (Reynolds, McCollum, Nguete, Shongo Lushima, & Petersen, 2017). Now the researchers need to find an effective way to identify monkeypox disease and do the data collection and research trial.

Recent developments in areas such as artificial intelligence and machine learning have made them one of the most helpful tools for clinicians (Bohr & Memarzadeh, 2020). Deep Learning is a subfield of artificial intelligence that assists in creating a model, automatically extracting the features without the need for human participation, training the model, and producing the results (Myszczynska et al., 2020). Imaging techniques of all kinds are already being put to use in the field of medicine, assisting medical professionals in making diagnoses of a wide variety of diseases, including brain cancer (Noreen et al., 2020) and other respiratory conditions like pneumonia and tuberculosis (Haloi, Rajalakshmi, & Walia, 2018), as well as COVID (Desai, Pareek, & Lungren, 2020) and other conditions. The analysis of medical images via deep learning has been extensively studied recently (Shen, Wu, & Suk, 2017). There is a need in this field because there are so many cases of monkeypox and not enough testing kits. Due to the low number of expert clinicians, it is been a challenging task to provide one to every hospital. Furthermore, the deep learning model may be able to address issues such as a lack of RT-PCR kits, inaccurate test results, high costs, and long wait times (Ozturk et al., 2020). Deep learning strategies have also been investigated to see if machine learning could provide a viable solution to the problem of developing an efficient triage strategy for the diagnosis of monkeypox sickness. The authors of this research deployed a deep learning model to improve the accuracy of the diagnosis of monkeypox.

Among some studies, several of the studies that were introduced captured our interest more than others. For the purpose of assisting with the diagnosis of Alzheimer's disease, Folego, Weiler, Casseb, Pires, and Rocha (2020) made use of 3D convolutional neural networks (CNN) for biomarker recognition in MRI. By using an ultrasound and a deep residual network (ResNet), Kuo et al. (2019) was able to accurately predict the start of chronic kidney disease. Apostolopoulos and Mpesiana (2020) developed a deep learning model that he termed "Darknet" by utilizing 224 positive image datasets from the Covid-19 database. The model had a 98.75% accuracy rate. Narin, Kaya, and Pamuk (2021)

achieved a 96% accuracy for COVID-19 detection using the ResNet50 ImageNet model. Rajpurkar et al. (2017) made CheXNet, which is a 121-layer convolutional neural network that is better at finding pneumonia than four working radiologists. There are some papers on chickenpox disease detection using deep learning techniques. A low-complexity convolutional neural network (CNN) was proposed by Sandeep, Vishal, Shamanth, and Chethan (2022) to identify skin illnesses such as psoriasis, melanoma, lupus, and chickenpox. They demonstrate that it is possible to diagnose skin diseases using image analysis with a level of accuracy of 71% by making use of the experimental VGGNet. On the other hand, the suggested method that they have come up with achieves the best outcomes by being approximately 78% accurate. A method that utilizes a smartphone and MobileNet to diagnose skin problems was proposed by Velasco et al. (2019). They came to the conclusion that people with chickenpox symptoms can be identified with 94.4% accuracy. Roy et al. (2019) used different segmentation strategies in their study to find skin diseases like acne, yeast infections, cellulitis, chickenpox, and others. Ali, Shams Nafisa et al. (2022) began by presenting a dataset referred to as "MSLD", which included a total of 228 images of three distinct kinds of skin lesions. They have increased the sample size by utilizing "data augmentation", which is a technique used to enhance the size of the sample, and they have established a 3-fold cross-validation study. In the subsequent stage, multiple different pre-trained deep learning models, such as VGG-16, ResNet50, and InceptionV3, are used to identify monkeypox and other diseases. Overall, ResNet50 is the most accurate model, with an accuracy of 82.96%. RManjurul Ahsan et al. (2022) presented the "Monkeypox2022" dataset, which was recently developed. They applied a modified deep learning model named VGG16 to the original and augmented datasets and obtained 97% and 88% test accuracy, respectively. However, in some papers, we demonstrated that the machine learning and deep learning results were very promising and accurate when it came to imaging datasets. Based on these results, we think that using deep learning techniques to classify monkeypox disease from an image-based dataset would give us benchmark results.

In this research, the authors have attempted to look into the performance of monkeypox identification from a newly developed dataset of images of monkeypox named *Monkeypox Skin Images Dataset*. The primary goal of this research is to develop a reliable framework for detecting patients with monkeypox using skin images obtained through the use of deep learning-based convolutional neural network models named *MonkeyNet* by developing a fully new image dataset. With its ability to learn on its own, CNN has recently emerged as the most popular deep learning approach for medical imaging classification. In order to detect patients with monkeypox using skin images, we used a relatively recent technology called CNN, in which the learning process is constructed using an arrangement of densely connected convolutional networks (DenseNet) (Huang, Liu, Van Der Maaten, & Weinberger, 2017). When it comes to deep learning models, one of the most important reasons for applying DenseNet-201 is that it helps to reduce the problem of vanishing gradients while also improving feature reuse and reducing parameter consumption. In order to optimize the flow of the architecture, the dense network would be driven by the concept of linking each layer to every other layer behind it. Instead of settling on a single final layer, this approach allows CNN to settle on a set of layers in total. Compared to other image processing technologies, DenseNet is more complicated and can take in more visual information.

As stated below, the following are some of the contexts in which this work makes a significant contribution to current research:

- Firstly, the authors have developed an entirely new image dataset named “Monkeypox Skin Images Dataset” that consists of four different image classes that are monkeypox, chickenpox, measles, and normal images. All of the images in the dataset came from different reliable online sources, such as reliable health websites, newspapers, journals, etc.
- Secondly, applied a new foster deep CNN model named *MonkeyNet* in combination with a data augmentation strategy to categorize and recognize monkeypox cases. The results of the RT-PCR tests may be a big part of any problems in very serious cases. The skin-based image classification method will be a good alternative because it has very higher accuracy for detecting monkeypox.
- Thirdly, the performance of our proposed deep CNN has been looked at by a number of different metrics, including test accuracy, precision, recall, AUC score, and F-1 score, as part of a wide range of experiments according to different hyperparameters and we selected the best value of the hyperparameters like batch size and learning rate for our presented original and augmented dataset.
- Fourthly, this research work filled in a conceptual gap by using a bigger set of medical imaging datasets than other research had used. This is because earlier research used datasets that were much smaller in size. The authors have tried to increase the size of the dataset.
- Finally, we have generated a Gradient-weighted Class Activation Map (CAM) for each class image for our presented dataset based on our proposed deep CNN model. In practice, the CAM image helps clinicians make quick and effective diagnoses and treatments by pointing the infected regions of the patient’s skin.

This remaining paper has been organized in the following way: The details of the experimental dataset are given in Section 2. After that, the architecture of the proposed model, methods used in this research, and the experimental setup used to evaluate the model’s performance are summarized in Section 3. The descriptions of all the evaluation metrics are illustrated in Section 4. Next, the experiments that were done to find a solution to the research gap are shown in Section 5 and the results of these experiments, the discussion of these results, and the interpretation of the data that was found are shown. Finally, in Section 6, a brief summary of the whole process is given.

2. Dataset collection

It is absolutely necessary to diagnose individuals who exhibit signs of monkeypox in this day and age, as the monkeypox disease is quickly spreading throughout many countries. Many artificial intelligence (AI) systems that interpret images are thought by many authorities in the medical sector to have the potential to make it simpler for doctors to diagnose outbreaks. Actually, the first time a pandemic situation spreads, it is very difficult to make a dataset for anyone (Fong, Li, Dey, Crespo, & Herrera-Viedma, 2020). For these reasons, any researcher depends on some reliable online-based resources. The journalists report various facts about that disease, statistics, or images of the pandemic during the pandemic in different journals. To solve the issues of the dataset, any researcher can depend on different reliable online-based health resources, as well as journals, newspapers, and any other resources. Meanwhile, at the moment this article was written, it was not possible to locate any publicly released Monkeypox database that was reliable, which makes it difficult to take advantage of implementing an AI-based technique to effectively diagnose and treat the Monkeypox disease. As a consequence of this, a significant number of researchers and experts are unable to

contribute to the detection of the monkeypox disease by utilizing cutting-edge AI approaches. In light of these constraints, the approach presented here involved the collection of patient pictures containing monkeypox. Our preliminary data set only includes a small number of samples, but this will not be a problem for the preliminary study. When researchers were building AI-based models in the early stages of COVID-19 disease (Khemasuwan & Colt, 2021; Lella & Pja, 2021), the authors were inspired by the large amount of published research that had only looked at small datasets before. Still, the database will always be updated with new information from many different organizations around the world.

Because there is not yet a cooperative dataset that is released to the public by a licensed hospital, clinic, or other reliable sources, the image data for monkeypox is put together from a variety of sources, such as reliable health websites, newspapers, online portals, and samples shared by the public resources (CDC, 2022; DermNet, 2022; IAC, 2022; NHS, 2022). In order to accomplish this, the Google search engine (Google, 2022) is utilized for the purpose of gathering the images that make up our dataset. Basically, the authors have collected two types of images: one is monkeypox and the other type is non-monkeypox images. Additionally, the non-monkeypox type includes three types of images: chickenpox, measles, and normal images. All these images have been obtained by searching for images of monkeypox, chickenpox, measles, and normal images from different reliable resources through Google. We have collected images of different parts of the body, like the face, hand, leg, fingers, etc. A total of 770 image samples have been collected, where chickenpox, measles, monkeypox, and normal classes contain 107, 91, 279, and 293 image samples, respectively. The authors named the dataset “MSID”, which is the short form of “Monkeypox Skin Images Dataset”. The augmentation method was used on the original dataset because of the need for a large number of images. We have uploaded the entire dataset to the “Mendeley Data” database for the purpose of further research (Bala & Diponkor, 2022). The dataset is easy to find and can be downloaded from the *Mendeley Data* database for free. The image samples of the different classes and the distribution of the images according to the classes is given in Figs. 1 and 2.

3. Methodology

Firstly, the experimental process started with the use of the cross-validation concept, which had to be adapted due to the small scale of the collected dataset to further prove the efficacy of our proposed dataset. In the next step, some machine learning classifiers were used to classify the image classes. Next, some pre-trained deep learning models (Wang, Fan, & Wang, 2021), and finally the dataset was trained and tested through the proposed deep learning-based CNN model. The proposed model is based on the implementation of a modified pre-trained model based on convolution neural networks (O’Shea & Nash, 2015) for the purpose of generating the output. The working environment is initialized by loading the monkeypox image dataset. Once the dataset has been loaded, it proceeds to execute data preprocessing procedures such as data normalization and data augmentation (Li, Wu, Lim, Belongie, & Weinberger, 2021), after which we will use our proposed CNN model for training in batches and allow it to run for a few epochs before moving on to the next section. To find out how well the model works, we need to test it with the testing dataset.

A number of major steps were taken in the proposed methodology, including:

- Data Preprocessing
- Model Development



Fig. 1. Samples of the image dataset.

- Training and Testing the Model
- Hyperparameters Settings

Fig. 3 illustrates the proposed methodological framework for the classification and prediction of monkeypox disease.

3.1. Data preprocessing

The data preprocessing step includes feature scaling, data re-sizing, splitting, and augmentation. These are illustrated in the following:

Feature Scaling: The procedure initially started by exporting the dataset so that we could work with it and modify the data so that it could be used by multiple classifiers to accurately forecast monkeypox disease detection. Data preprocessing consists of a number of operations, such as extracting the labels from images, turning the image into RGB format, resizing the image, and executing feature scaling. First, using OpenCV methods (Bradski &

Kaehler, 2008), the skin images in the dataset are transformed to RGB and then downsized to 224×224 pixels. The standard image size is in the range $[0,255]$, with 255 being the largest image size allowed by the system.

For feature scaling, images and labels are first loaded into two independent datasets, and then these datasets are turned into NumPy arrays for use in the next step. In this case, it is necessary to conduct maximum normalization on the images. Normalization techniques, such as feature scaling, are used to standardize input data into a fixed range by performing operations on the independent variables of the data. It is useful in tying together values between two numbers, particularly those between $[0,1]$. This is accomplished by dividing each image by the maximum size (255) and translating the data into the range $[0,1]$ (Pei & Lin, 1995). Using the one-hot encoding method, labels are given to the data that goes with each image. It follows that the normalized value is denoted by the symbol x' .

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x is the original intensity of the image.

Data splitting: As a result of the feature rescaling, all of the images in the dataset are resized into the range (224,224), where the image's height and width are both 224 pixels. The dataset must now be divided into two parts: a training portion and a testing portion. Specifically, in this study, we divided the dataset into two parts: 80% for training and 20% for testing our proposed model, respectively. From the training dataset, 20% of the image samples are utilized for the model validation. Table 1 provides further information regarding the manner in which the two datasets have been split according to the splitting ratio.

Data Augmentation: Data augmentation is a process that is used to expand the size of a dataset by applying random transformations to the original data (Shorten & Khoshgoftaar, 2019). ImageDataGenerator is a class in the Keras deep learning framework (Ketkar, 2017) that allows us to fit the model using image data. It is possible that we will have greater variability in our dataset as a result of this, and in addition to this, it will be utilized to boost the total number of training samples in an effort to prevent overfitting. In every epoch, all of the original images were transformed and augmented, and the resulting images were used for training in order to prevent overfitting. Because it was trained on numerous variants of the same image, the model was able to be more robust and accurate. It was determined that the number of images in each epoch was equivalent to the number of images in the original images. When training data is used, the ImageDataGenerator model (Bhandari, 2020) is used to enrich it with new information. We have applied mainly positional and color augmentation techniques. With the positional augmentation, the color augmentation technique has been utilized since every image was found online, where there is a wide range of lighting conditions. Fig. 4 depicts the augmented

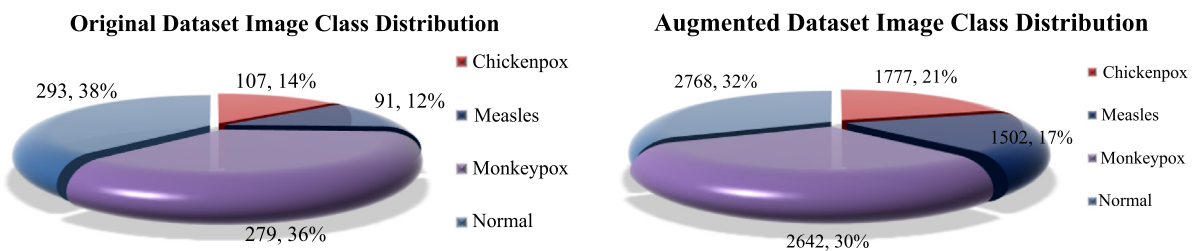


Fig. 2. Distribution for original (left) and augmented (right) dataset.

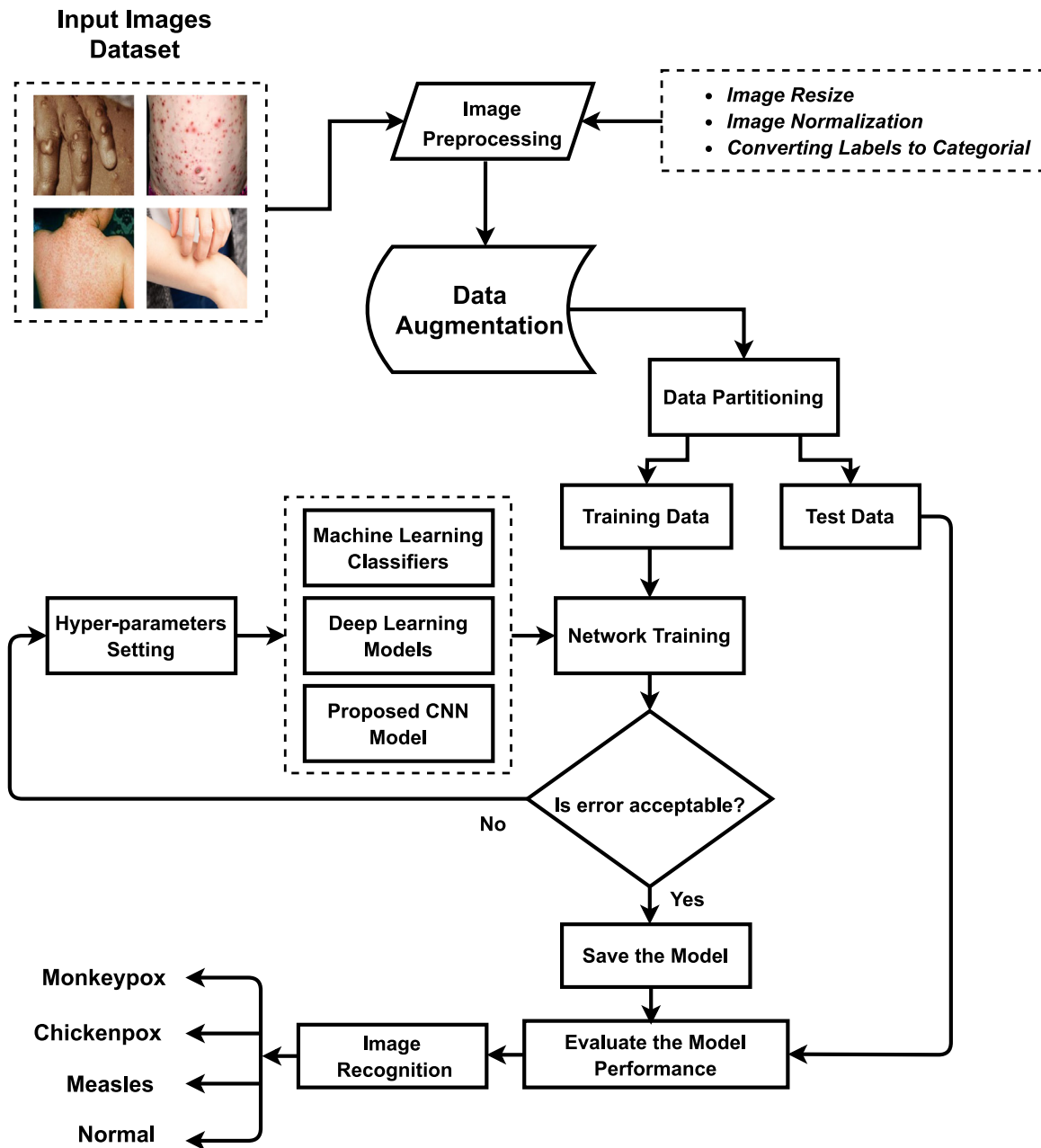


Fig. 3. Proposed methodological framework.

Table 1
Dataset splitting in details.

Image classes	Original dataset				Augmented dataset			
	Training	Validation	Testing	Total	Training	Validation	Testing	Total
Chickenpox	69	17	21	107	1138	284	355	1777
Measles	58	15	18	91	960	241	301	1502
Monkeypox	178	45	56	279	1691	423	528	2642
Normal	187	47	59	293	1771	443	554	2768
Total	492	124	154	770	5560	1391	1738	8689

images that were created from a single sample. The applied image augmentations parameters are tabulated in Table 2.

When dealing with class imbalances, this strategy was chosen over random oversampling, which was one method of dealing with the imbalances. When using random oversampling, it was

necessary to re-sample less frequent samples in order to adjust their amount in contrast to dominating samples. The distribution of classes, on the other hand, would alter dramatically, with the smaller classes having significantly less variance and the bigger classes having significantly more variety.

Table 2
The parameters for data augmentation that we utilized in this research.

Parameters name for data augmentation	Parameters value	Action
Rotation range	45	The input data are created by rotating −45 to 45 degrees.
Horizontal flip	True	Randomly flip inputs horizontally.
Vertical flip	True	Randomly flip inputs vertically.
Zoom range	[0.8, 1.25]	Zoom in or out by 0.8 to 1.25 distance from the middle.
Shear range	45	Images are randomly sheared by 0 to 45 degrees.
Height shift range	0.3	Randomly shifted in height by 30%.
Width shift range	0.3	Randomly shifted in width by 30%.
Brightness range	[0.1, 2]	Randomly changing brightness by the range of 0.1 to 2.
Contrast jitter	[0.5, 2]	Randomly changing contrast by the range of 0.5 to 2.
Hue jitter	0.5	Randomly changing hues by the value of 0.5.
Saturation jitter	[0.2, 3]	Randomly changing brightness by the range of 0.2 to 3.
Fill mode	Constant	The gaps are filled in with the pixel value of black.

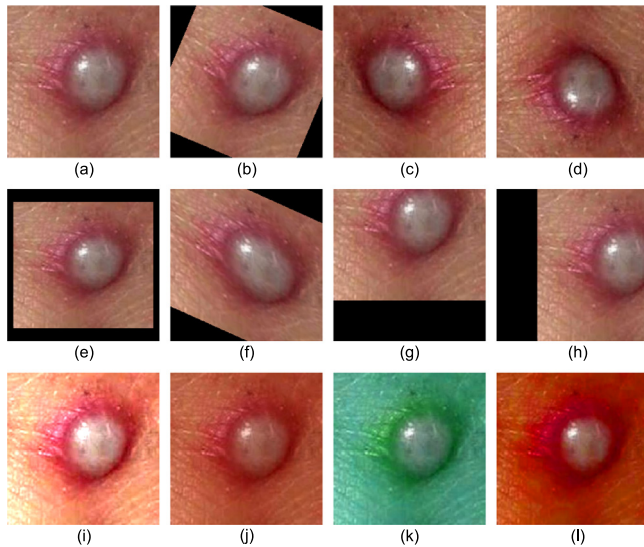


Fig. 4. Original monkeypox image (a) and its 11 types augmented images for the augmented dataset. The augmented images includes (b) Random rotation, (c) Horizontally flipped, (d) Vertically flipped, (e) Randomly zooming, (f) Randomly sheared, (g) Randomly height shifted, (h) Randomly width shifted, (i) Brightness jitter, (j) Color jitter, (k) Hue jitter, and (l) Contrast jitter.

3.2. Model development

The model development stage includes some machine learning classifiers, pre-trained deep learning models, and the proposed CNN model.

3.2.1. Machine learning classifiers

Firstly, five machine learning classifiers have used for the classification of monkeypox disease. The classifiers are briefly described in the following:

Logistic Regression: Logistic regression is a type of classification algorithm that requires supervision. When the predictor variable is categorical, which makes it simpler to apply and analyze, this method is the one that is utilized. The classes that are going to be predicted are more than two, so Multinomial Logistic Regression is needed. It is often called softmax regression, because it uses a generalization of the sigmoid function, the softmax function (Dreiseitl & Ohno-Machado, 2002). In Multinomial Logistic Regression, a document d is assigned to the class c , which is the one with the highest probability. These probabilities are computed via the softmax function:

$$P(y = c|x) = \frac{\exp(w_c x + b_c)}{\sum_{j=1}^C \exp(w_j x + b_j)} \quad (2)$$

where x is the input vector and w, b are the parameters.

Random Forest: Ultimately, the random forest is a collection of decision trees. A tree is constructed using random subsets of the dataset. A decision tree algorithm does not handle the problem of overfitting as effectively as a decision tree. There are many reasons for this, but the most common is that a single tree begins to construct rules for the minority class, which results in overfitting. Because each tree receives a vote in a random forest process, the final outcome is more generalized than if this problem were to arise. Entropy and Gini impurity are two of the most commonly used measures to ensure that the split is of high quality. Entropy will be used to determine how pure the split is Liu, Wang, and Zhang (2012). It is contained in the following formula:

$$E(S) = \sum_{i=1}^m -p_i \log_2(p_i) \quad (3)$$

where p_i is the probability of class ‘i’. The impurity of the split is measured by Gini impurity, which is akin to entropy. If Gini impurity equals 0, the split is considered ideal. Impurity in the Gini formula, p , shows how likely it is that each of the classes will split apart.

$$GiniIndex = 1 - \sum_{i=1}^n (p_i)^2 \quad (4)$$

The Gini impurity method uses less computer power than entropy does. Therefore, we use Gini impurity in our implementation of the random forest.

K-Nearest Neighbor: In addition to its application in classification and regression, the supervised learning model known as K-NN is also utilized. The result of k-Nearest Neighbors is dependent on the application in which it is utilized, whether for classification or regression. K-NN begins with loading the data and initializing the value of k , which is simply the number of neighbors. Calculating the distance between the query data and the current data, adding that distance, and indexing the example into the ordered collection are the three steps that are carried out in this process. After that, it selects the first k entries from the sorted collection and strips the labels off of those k entries that were chosen. If the procedure is being used to perform regression, it will return the mean of those k labels; if it is being used to perform classification, it will return the mode value of those k labels. However, if $k = 1 \dots n$, it will be allocated to a single or multiple nearest neighbor and equation is expressed as (Zhang, 2016)-

$$D_M = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (5)$$

where D_M is the Minkowski distance among the points.

Support Vector Machine: As a supervised learning model, an SVM, or support vector machine, can be used to categorize data

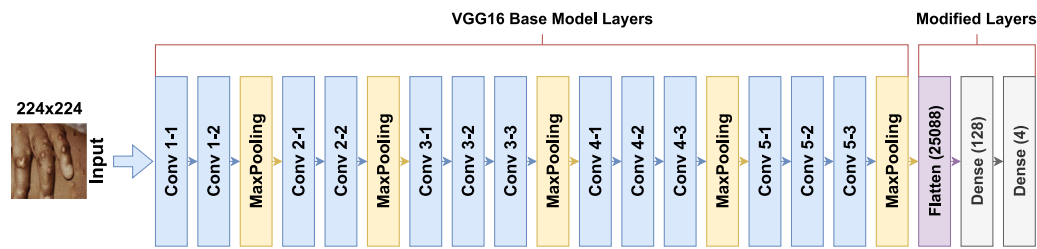


Fig. 5. VGG16 model architecture.

and solve regression issues. In order to categorize or split the data by a clear separation as broad as feasible, points in space are employed to represent the data. It forms a line, or what we may call a hyperplane, to help us categorize the data. Using the SVM technique, we can determine which points in each class are closest to the hyperplane, a concept known as support vectors. The margin is the distance between the support vectors and the hyperplane through which the hyperplane can be seen. If an n -dimensional vector or list of n elements has a single data point in each of two classes, then the new data point must be assigned to one of the two classes. The $(n - 1)$ dimensional hyperplane can be checked to see if there are any independent $(n - 1)$ dimensional points (Hearst, Dumais, Osuna, Platt, & Scholkopf, 1998).

Extreme Gradient Boosting: Boosting is a type of ensemble learning, and the model itself is often made up of a number of different decision trees. The correction of faults caused by an existing model can be made by incorporating other models into it. Models are added until there are no more ways in which they can be improved. When trying to fit new models to an existing one, gradient boosting employs the process known as gradient descent in order to try to minimize the loss function (Chen et al., 2015). The following equation is used to maximize the value of a cost objective function which is denoted by Ω .

$$\Omega(\theta) = \sum_{i=1}^n d(y_i, \hat{y}_i) + \sum_{k=1}^K \beta(f_k) \quad (6)$$

where \hat{y}_i represents the value of prediction, n is the total cases in the training sample, K is the total number of trees that need to be constructed, and f_k is a member of the group of trees known as the ensemble trees.

3.2.2. Deep learning models

In the second round of our monkeypox disease classification process, we deployed five different deep learning models. Each model is trained on the trained dataset with its pretrained weights and adds an extra three layers at the end of the model that are one flatten layer, one dense layer, and an output layer with four classes. We changed the input shape to 224×224 in each model. The following is a condensed description of the various models:

VGG16: The VGG16 is a deep convolution neural network proposed by Oxford University members. It is a very deep convolution neural network that helps with image classification and image recognition. The model achieved 93% accuracy in ImageNet, which is a collection of 14 million images that belong to more than 1000 classes. It overcomes the problems of AlexNet by replacing the size of kernel filters with multiples of 3×3 size. The VGG16 architecture comprises thirteen convolution layers, five maxpooling layers, and three dense layers. Those convolutional layers have very small receptive areas (3×3). A maximum pooling layer follows some of those layers. These layers are CNN model layers with different numbers of filters, sizes, and stride values (Simonyan & Zisserman, 2014). The Fig. 5 shows the utilized VGG16 model architecture.

ResNet-50: ResNet is an abbreviation for “Residual Network”, and this network brings to the field residual learning. It has 50 layers and is usually used for various computer vision tasks such as image classification, object localization, object detection, etc. It is also possible to use it for other tasks besides computer vision in order to provide an illusion of depth for those tasks. Generally, in a deep convolutional neural network, many layers are stacked together and are trained for a specific task at hand. The network is taught many low-to high-level features by the end of its layers. In residual learning, the network does not only try to learn some features, but it also tries to learn some residuals as well. Residual can be very simply understood as the subtraction of a feature learned from the input of that given layer. ResNet is able to do so using some shortcut connections that directly connect the input of the n th layer to some $(n + x)$ th layer (He, Zhang, Ren, & Sun, 2016). In this paper, the ResNet-50 model has been utilized with some modifications in the last layers. The Fig. 6 shows the utilized ResNet-50 model architecture.

MobileNetV1: MobileNet is a type of CNN created to be able to run on mobile devices with low processor power capabilities. They are built on a streamlined architecture that utilizes depth-separable convolutions. This helps in building a lightweight deep neural network that has low latency for mobile and embedded devices. MobileNet is made so that it mostly uses pointwise separable convolution instead of full convolution. MobileNetV1 consists of a regular 33 convolution layer followed by 13 blocks of 3×3 depthwise convolution, batch normalization, and ReLU, with 1×1 pointwise convolution, batch normalization, and ReLU. There are no pooling layers in between those depthwise separable convolution blocks. The stride of 2 is used to reduce the spatial dimension of those inputs. The number of output channels is also doubled in the pointwise layers. All these layers have batch normalization. MobileNet uses ReLU as its activation function. The architecture is completed with a global average pooling layer at the very end. The final three layers implemented with a flatten layer, a dense layer and final classification layer used softmax as activation function (Howard et al., 2017). The Fig. 7 shows the utilized MobileNetV1 model architecture.

Inception V3: Inception V3 is the enhanced version of Inception V1, also known as GoogleNet, which is assembled using a so-called Inception architecture. Inception V1 is not only deep in the direct sense, but the depth is also apparent within the used Inception modules. Inception modules create sparsity by applying 1×1 , 3×3 , and 5×5 convolutions after dense components. Additionally, 1×1 convolution is used as a non-linear dimension reduction before 3×3 and 5×5 convolutions to maintain sparsity and prevent an increase in computational requirements. The network mainly consists of stacked Inception modules. The occasional introduction of Max-Pooling down-samples the features by halving the resolution. These modules make it possible to focus on higher-level features at higher network levels, which makes it possible to process features on a large scale. Lastly, label smoothing is added to try to prevent overfitting by keeping the model from becoming too confident in a certain class (Szegedy,

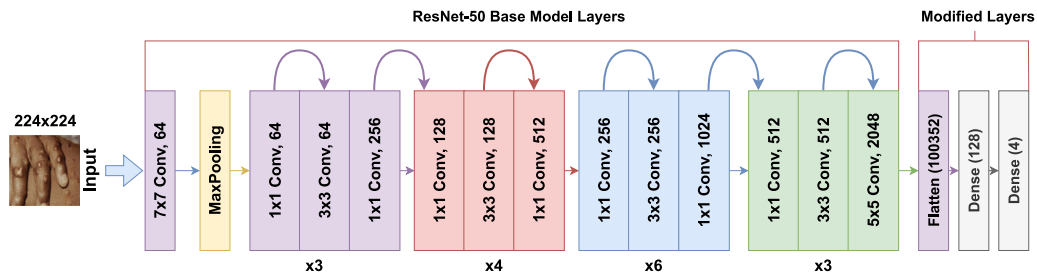


Fig. 6. ResNet-50 model architecture.

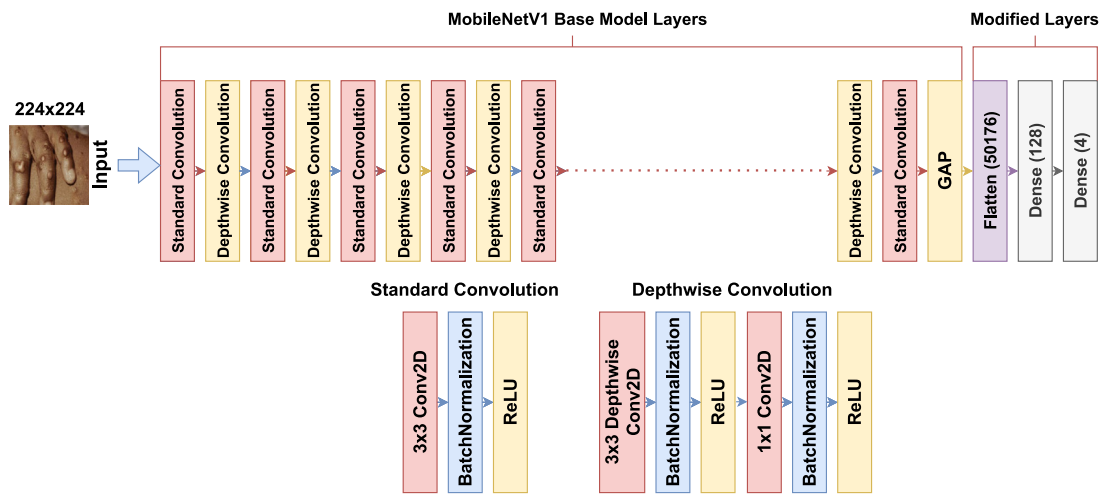


Fig. 7. MobileNetV1 model architecture.

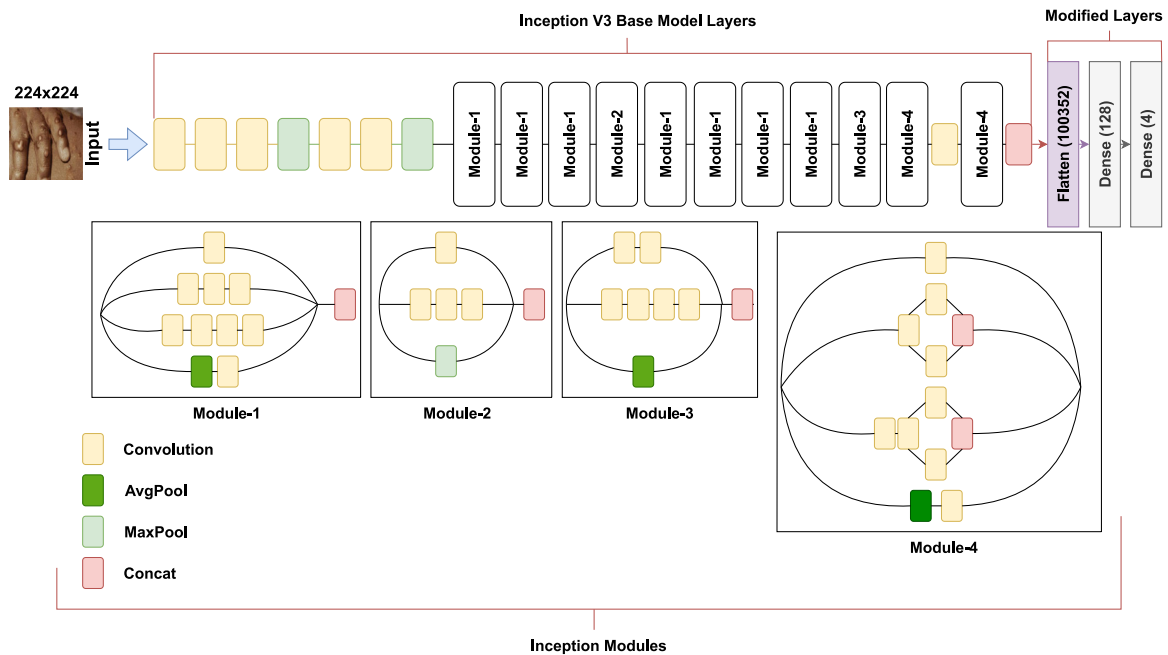


Fig. 8. Inception V3 model architecture.

Vanhoucke, Ioffe, Shlens, & Wojna, 2016). The Fig. 8 shows the utilized Inception V3 model architecture.

Xception: “Xception” is a “depthwise separable variant of Inception” where $n \times n$ spatially convolution channel-wise is referred to as “depthwise convolution”. The Xception neural network is a CNN architecture that has 71 layers of depth. The Xception architecture includes a total of 36 convolutional layers,

which serve as the foundation of the network’s feature extraction process. Convolutional layers are grouped into 14 modules, for a total of 36 layers. With the exception of the first and last modules, all of the modules have linear residual connections surrounding them. The Xception architecture is comprised of a linear stack of depthwise separable convolution layers that have

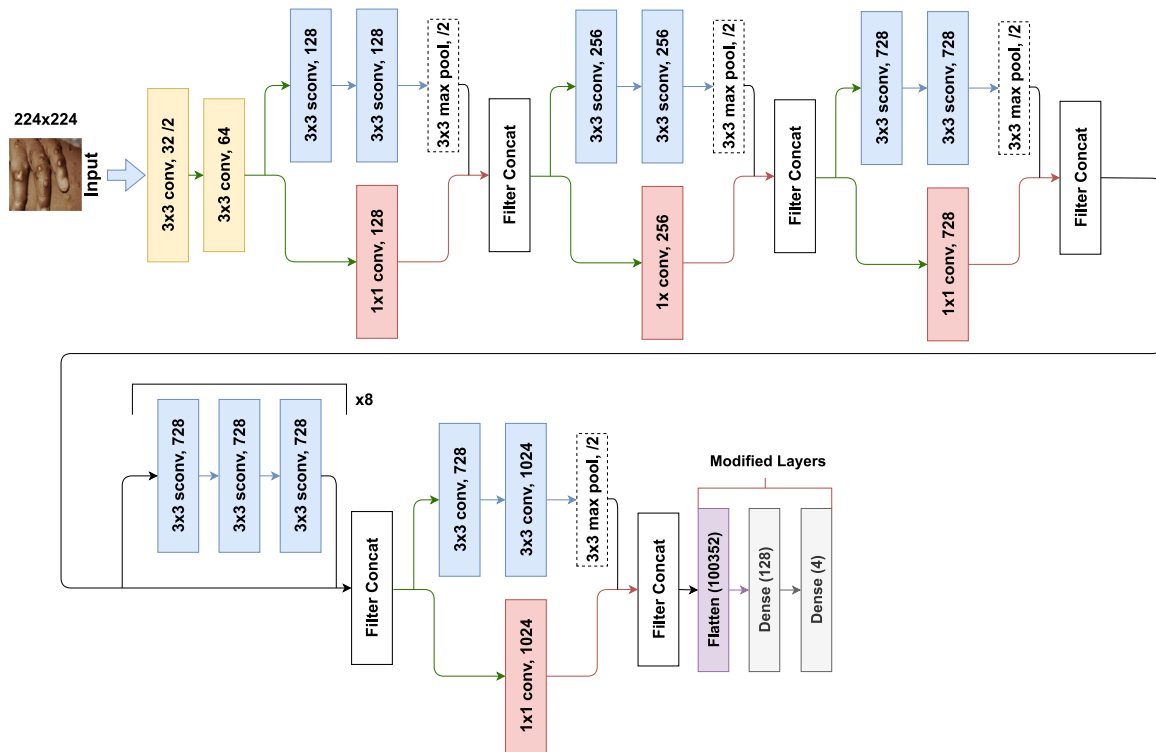


Fig. 9. Xception model architecture.

residual connections. The Xception model achieved 94.5% accuracy in ImageNet, which is a collection of 14 million images that belong to more than 1000 classes (Chollet, 2017). The Fig. 9 shows the utilized Xception model architecture.

3.2.3. Proposed model architectures

As a result of recent developments in deep learning methods, artificial intelligence has been transformed. The term “deep states” refers to the fact that the network layers in the model are growing in size. A convolution layer, a maxpooling layer, and a dense layer are all components of the CNN structure. Convolution layers gather features from input data through filters, and a max-pooling layer is used to lower the size of the layer, which improves computational efficiency. A dense layer assists in connecting the layers, which is referred to as a fully connected layer. A comprehensive CNN model is constructed by integrating all of these layers together (O’Shea & Nash, 2015). The hyperparameters of the CNN model are tweaked to perform specific tasks, such as object identification or object classification.

It is necessary to provide the input shape in order to develop our model since the models must know what kind of input they will be receiving. We have used a pretrained DenseNet-201 model as the foundation for our network in order to create our entire model and we named our proposed model as “MonkeyNet”. The following are the essential levels of the proposed model, as depicted in Fig. 10 and detailed further below:

Input Layer: Through this layer, the model transmits the information that was received at the input layer to the hidden layer and then onward to the output layer (Albawi, Mohammed, & Al-Zawi, 2017). In this study, the image shape for the input layer is (224, 224, 3), with the height and width of the image being 224 and 224, respectively, and three channels in the image.

Convolutional Layer: A convolution is defined as a mathematical process that describes a rule for combining two sets of information into a single new set of information. The convolution layer is the fundamental building component of a convolutional network, and it is responsible for the majority of the

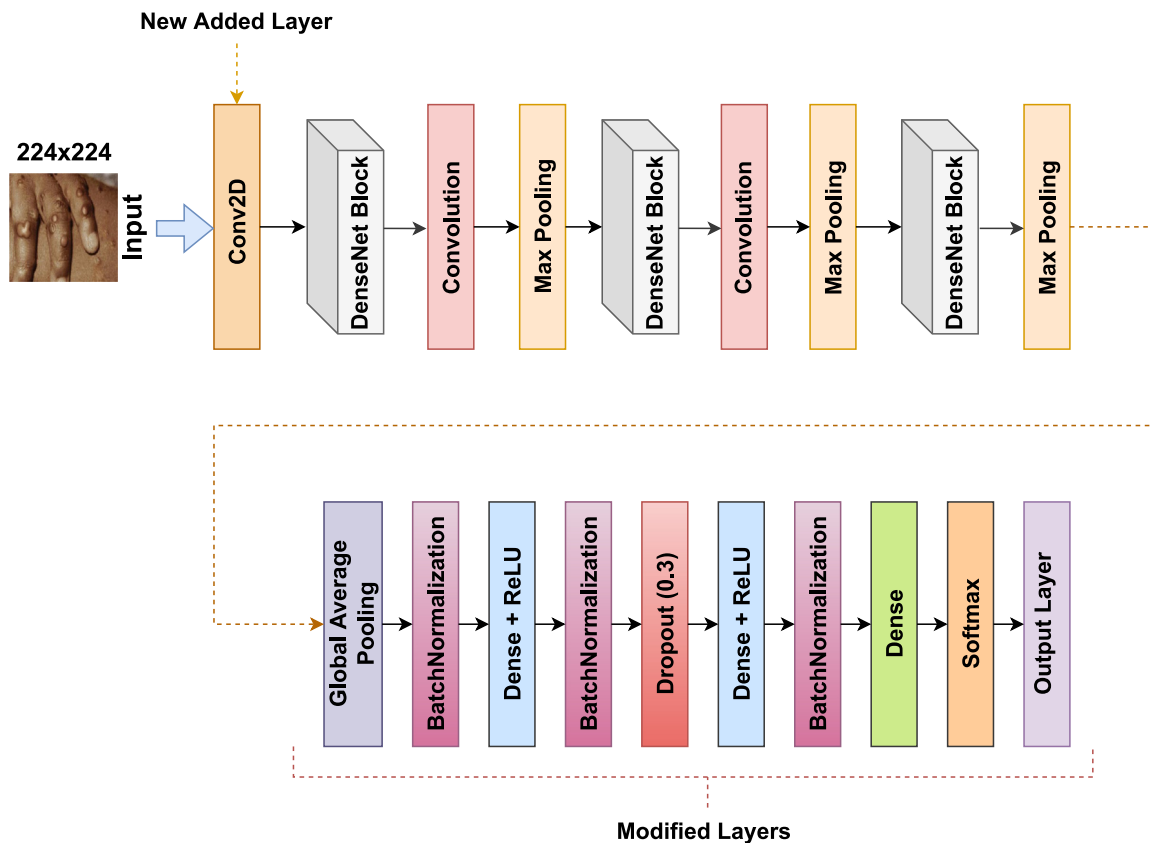
computation-intensive work. The convolution layer has some parameters and hyperparameters that are built up of filters, and it is through these filters that this layer pulls features and learns them. The main goal of this process is to use a method for extracting features to find out what the features in the images are like, which will then be stored as part of an arbitrary feature vector. In a nutshell, this is referred to as a feature extractor layer. The input images are compared with segments in order to determine the differences, and these segments are referred to as features. From the input images, this layer extracts one or more features and uses the image matrix to construct one or more matrices and dot products with the image matrix. In addition, the whole process comes up with a result called the convolution layer, which is the result of the entire process (Albawi et al., 2017).

For the input image (X) and kernel K , the 2D convolutional operator is defined as:

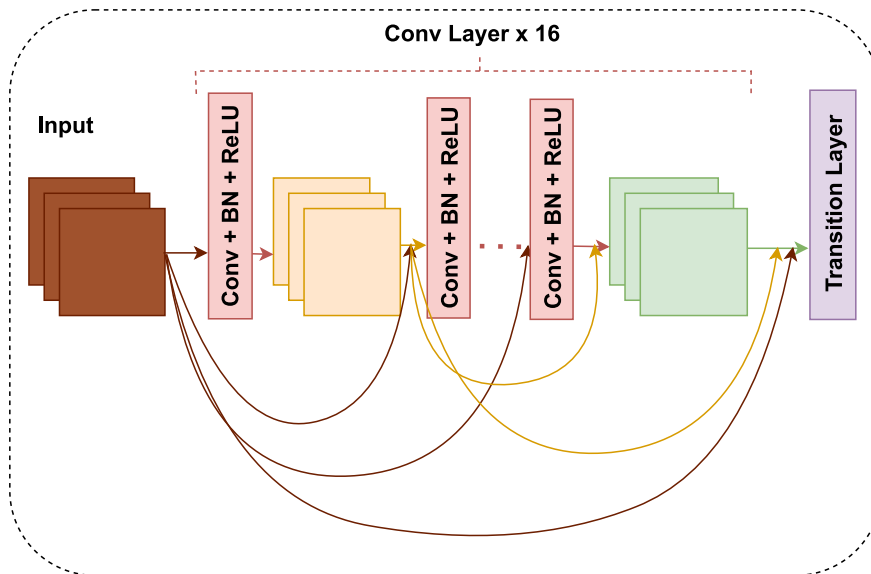
$$(X * K)(i, j) = \sum_m \sum_n K(m, n)X(i - m, j - n) \quad (7)$$

where $*$ represents mathematical representation of convolution operation, the k matrix moves over the input data matrix with stride parameter.

Base Model: In the field of neural networks for image classification and object recognition, DenseNet is one of the most recent discoveries in the field. With the exception of a few important differences, DenseNet is quite similar to ResNet. A common difficulty that occurs in deep CNNs is that a significant amount of information about the input tends to be lost by the time the network reaches its viewpoint (the vanishing gradient problem). DenseNet has improved the complexity of convolutional neural networks, which is a good thing. By connecting layers via concatenation within blocks of layers, DenseNet is specifically designed to improve the feed-forward characteristics of the network by increasing the amount of information that flows through CNN layers. A DenseNet is a type of CNN network that is specifically designed to maximize the amount of information that flows through CNN



(a) MonkeyNet Model



(b) DenseNet Block

Fig. 10. The proposed CNN model architecture.

layers. With max-pooling, the initial layer captures a large portion of the moving window while maintaining a low parameter count. The output of the dilated convolutional layer is routed into two dense blocks that are successively placed and joined by a transition layer. Convolution 1×1 followed by average-pooling are the building blocks of the transition layer. In order to get the expected class distribution, the output of the last dense block is sent through a single convolutional layer with max-pooling

and then a fully connected softmax layer (Huang et al., 2017). The Dense Block, which is used to make the DenseNet model, is shown in the Fig. 10(b).

The DenseNet-201 model was used in this work as a base model, which was chosen from among the different DenseNet (DenseNet-121, DenseNet-169, and DenseNet-201) architecture. The Densenet-201 is employed $5 + (6 + 12 + 48 + 32) \times (2) = 201$ layers. Details of the DenseNet-201 is following:

- 5—convolution and pooling layers
- 3—transition layers (6,12,48)
- 1—Classification layer (32) and
- 2—denseblock (1 × 1 and 3 × 3 conv).

Consider the case of a network with L layers, every single one of which produces a non-linear transformation H_l . The outcome of the l th layer of the network is denoted by the symbol x_l , while the input image is denoted by the symbol x_0 . Conventional feed-forward networks, as we all know, connect the output of the l th layer to the output of the $(L + 1)$ th layer. Furthermore, the skip connections can be stated as follows:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{8}$$

When compared to a regular CNN, DenseNets requires less parameters, as a result of which duplicate feature maps are eliminated, allowing for feature reutilization. As a result, the feature-maps of all previous levels, x_0, \dots, x_{l-1} , are passed to the l th layer as input:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \tag{9}$$

where the concatenated of the feature-maps is represented by $[x_0, x_1, \dots, x_{l-1}]$. In order to make process easy, the numerous inputs of H_l are integrated into a single tensor. An H_l function is defined as a composite function that performs three sequential operations: batch normalization (BN), a rectified linear unit (ReLU), and a convolution (Conv). The operation H_l generates a total of k feature maps, hence the l th layer has a total of k feature maps. After that, we have the input feature-maps.

$$k_l = k_0 + k * (l - 1) \tag{10}$$

where x_0 denotes how many channels are available on the input layer.

The design of DenseNet solves the vanishing gradient problem, strengthens feature propagation, encourages data reuse, reduces the number of parameters used, and hence presents an extremely powerful learning model (Hegde et al., 2021).

Activation Layer: The Rectified Linear Unit layer, often known as the ReLU layer, is a layer with an activation function that is used in CNN models to increase nonlinearity. The ReLU activation function was used in the convolution layers to enhance the quality of the results. Many types of neural networks now use ReLU as their default activation function, which is a significant advancement. Models that are initiated with ReLU may be trained quickly and efficiently, and they also perform better. If the result of the operation is greater than 0, the convolution layer passes it as input; otherwise, it returns 0. The ReLU layer is free of any parameters or hyperparameters of any kind (Agarap, 2018). The equation of the ReLU function is represented as:

$$ReLU(x) = \max(0, x) \tag{11}$$

where x represents the value that is input into the neuron.

Batch Normalization Layer: Batch Normalization was developed to address the problem of internal covariate shift, which occurs when the parameters of layers change as the parameters of prior layers change during training. The normalization by batching between layers enables us to employ higher learning rates and, as a result, accelerates the training process significantly (Albawi et al., 2017).

Dropout Layer: Dropout is a regularization method that reduces model complexity and helps solve the overfitting problem. During neural network training, it randomly eliminates units by setting a layer's activation to 0 (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Fig. 11 depicts an example of a typical dropout situation. In the default configuration, the likelihood that a neuron will remain on or off is set to 0.5. In the

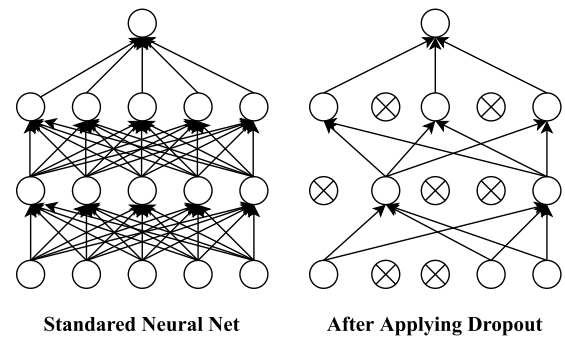


Fig. 11. Dropout example.

proposed design, we used one dropout layer with a drop rate of 30%, which means that 30% of random neurons were turned off in order to prevent our proposed model from becoming overfitted.

Dense Layer: The dense layer function is responsible for determining the relationship between all of the characteristics that have been provided to it without the use of any more input parameters than convoluted layers (Albawi et al., 2017).

Output Layer: This layer is responsible for achieving the final predicted class. Because of this, the sigmoid function is at the heart of the probabilistic approach, and it works best when distinguishing between two classes. The softmax function is applied to the multiclass classification, and softmax ensures that the total of the probabilities of the outcomes is one. As a result of this consideration, the softmax activation function has been chosen for the proposed model. After being transformed into a probability array of all four classes by using softmax, the input data is compared to the actual output to determine whether or not the likelihood of occurrence is higher (Albawi et al., 2017). The following is the definition of the softmax function:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{i=1}^K e^{z_i}} \text{ for } i = 0, 1, \dots, K \tag{12}$$

This function accepts a K -dimensional input vector z and returns a K -dimensional vector containing values within the range of 0 to 1 that sum up to 1.

3.3. Training and testing the model

Using the adam optimizer (Kingma & Ba, 2014) and categorical crossentropy loss function (Ho & Wookey, 2019), we are able to compile the model, and after that, the model is trained. When we were training our model, we used the training dataset, and when we were validating it, we used the validation dataset. The accuracy metric is used to assess the model's overall performance. Finally, the model is put through its paces on the test data.

3.4. Models hyperparameters

The proposed models used included a large number of parameters, which meant there were a large number of possible alterations in the architecture. When optimizing the model, hyperparameters are tweaked to achieve the best results. They also assist in determining the hyperparameter value that is closest to the one that provides the best performance. As a result, we used several commonly used hyperparameter values and looked into other ways to produce a more accurate model evaluation and prediction. Table 3 summarizes the hyperparameter values that were employed throughout the proposed deep CNN model. According to the suggested model architecture, there are a total

Table 3
Models Hyperparameters.

Hyperparameters	Values
Optimizer	Adam
Loss fFunction	Categorical cross-entropy
Epochs	100
Batch size	8, 16, 32
Learning rRate	0.03, 0.003, 0.0003

of 18,351,640 trainable parameters to be estimated, with 233,280 non-trainable parameters.

The following explains the reasoning behind the values of the hyperparameters selected:

- Optimizer:** Adam (Kingma & Ba, 2014) is currently the most widely utilized optimization algorithm for training deep neural networks. This is due to the fact that it is simple to use, has high computational efficiency, and is particularly effective when dealing with enormous amounts of data and parameters. One way to think about Adam is combining RM-Sprop (Zou, Shen, Jie, Zhang, & Liu, 2019) with the stochastic gradient descent algorithm and adding momentum. In this study, we have used Adam optimizer with the β values of 0.9 and 0.999, ϵ of 0.1 and weight decay of 0.01. In this strategy, the weights are updated by employing the following procedure:

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * \nabla_{w_t} \tag{13}$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * (\nabla_{w_t})^2 \tag{14}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{15}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{16}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} * \hat{m}_t \tag{17}$$

- Loss Function:** The loss function known as categorical cross-entropy is utilized for the purpose of single-label classification. This occurs when there is only a single category that can be applied to any individual data sample (Ho & Wookey, 2019). This worked perfectly here as one example could only belong to one of the two class categories.

$$\text{Loss} = - \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \tag{18}$$

where \hat{y}_i and y_i is the i th scalar and corresponding target value respectively, and “output size” refers to the number of scalar values that are produced by the model.

- Epochs:** Upon multiple initial trials with values of 20, 50, and 100, 100 epochs was sufficient to get to the most optimum results.
- Batch Size:** Upon multiple initial trials with values of 8, 16, and 32, a batch size of 8 and 32 produced the most optimal results in corresponding dataset.
- Learning Rate:** A learning rate annealer was utilized here. A decreasing learning rate during training enabled the global minimum of a loss function to be reached efficiently (Balles, Romero, & Hennig, 2016). Learning rate started at 0.003 and decreased by factor of 0.7 if the validation accuracy was not improved after 10 epochs (patience).

The entirety of the research was carried out on a compiler named Jupyter Notebook. This notebook includes a number of different packages, and it is powered by an “Intel[®] Core™ i7-10510U (1.8 GHz, up to 4.9 GHz, 8 MB cache, 4 cores) and NVIDIA GeForce MX330 (2 GB), 16 GB DDR4 RAM”, as well as a Windows 10 based 64-bit operating system. For training and testing data, we made use of the Google Colabratory platform. As a result of this, we evaluated the performance of our proposed model, which was built on the predictions generated by our trained model on the test dataset.

4. Evaluation metrics

Evaluation metrics calculated in the model included the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These numbers were later used to calculate our four main quantitative measurements for the classification performance on the proposed methods in terms of precision, recall, F1 score, and accuracy metrics. All these quantitative measurements have been chosen due to their effectiveness for the classification task, as well as their frequency in closely related research. The degree to which the model accurately categorizes the images contained within the test dataset set is referred to as its accuracy. The term “precision” refers to the ability to accurately estimate the true positive outcomes for both categories. The recall is a prediction of cases that are true positives in both groups, and it represents those cases. The F-1 score is a metric that reflects the connection between precision and recall. The F-1 score goes from 0 to 1, with 0 being the worst model and 1 being the best. The performance of the models in response to a wide range of situations is shown by the area under the curve (AUC) (Hossin & Sulaiman, 2015). The mathematical expressions for these metrics are as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{19}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{20}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$F - 1 \text{ score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{22}$$

$$\text{AUC} = \frac{\sum r_i(x_p) - x_p((x_p + 1)/2)}{x_p + x_n} \tag{23}$$

5. Results and discussion

Firstly, the cross-validation technique has been adapted due to the small scale of the collected dataset to prove its efficacy. The entire data set is divided into k sub-folders of the same size as part of the k -fold cross-validation strategy. After that, the learning function is trained utilizing $(k - 1)$ sub-folders of training data, and the remaining fold is utilized to test the model. Each individual subset of the data is put to use for either training or validation. After using all of the data subsets in the sequence, a cumulative average validation score is calculated. We have applied the 5-fold cross-validation technique, where four folds (80% of the data) have been used for training and the remaining two folds (20% of the data) have been used for the evaluation of the test accuracy of the model. According to these experimental tasks, our proposed model performed with a precision, recall, F-1 score, and accuracy of 90.38%, 89.84%, 89.85%, and 89.84% on the original dataset and 97.60%, 97.61%, 97.60%, and 97.61% on the augmented dataset, respectively, as well as lower standard deviation values for each metric on both datasets. Based on the

Table 4
5-fold cross-validation results for original dataset.

Number of fold	Evaluation metrics			
	Precision (%)	Recall (%)	F-1 score (%)	Test accuracy (%)
Fold 1	90.71	90.21	90.22	90.21
Fold 2	92.02	91.91	91.92	91.91
Fold 3	93.24	93.16	93.10	93.16
Fold 4	88.68	87.18	87.28	87.18
Fold 5	87.27	86.75	86.74	86.75
Mean	90.38	89.84	89.85	89.84
Standard deviation	0.0217	0.0253	0.0250	0.0253

Table 5
5-fold cross-validation results for augmented dataset.

Number of fold	Evaluation metrics			
	Precision (%)	Recall (%)	F-1 score (%)	Test accuracy (%)
Fold 1	89.94	89.98	89.94	89.98
Fold 2	98.75	98.74	98.74	98.74
Fold 3	98.41	98.41	98.41	98.41
Fold 4	100	100	100	100
Fold 5	99.91	99.91	99.91	99.91
Mean	97.60	97.61	97.60	97.61
Standard deviation	0.0386	0.0384	0.0385	0.0384

results of our experiments using the cross-validation method, we could state that both the dataset we presented and the deep CNN model are good for detecting and classifying monkeypox disease. The experimental results using the cross-validation technique for both datasets are tabulated in Tables 4 and 5.

Secondly, we performed the classification of monkeypox disease by utilizing some machine learning classifiers. For this experiment, we have used the pre-trained deep learning model as a feature extractor. After the feature extraction, machine learning classifiers have been used for the classification of different image classes. The classification results for the machine learning classifiers are given in the following Table 6. From Table 6, we see that the Linear Regression classifier provides the all the highest metric values among all other ML classifiers when the MobileNetV1 model architecture has been used as a feature extractor. In rows and columns, bold face values show the maximum value of the metrics for the classifiers and feature extractors.

Thirdly, for the deep learning approach, the model that has been trained is saved using the Hierarchical Data Format version 5, denoted by the extension “.h5” in the file name. These models are now available to be used in any area of application for the purpose of classifying monkeypox. With its adaptability and capacity for handling any number of data points, this saved model can be used to check for monkeypox on the patient’s skin images.

Now we will discuss the evaluation procedure carried out on the deep learning models and the proposed model to analyze its efficiency. This section explains the results obtained on various steps that were performed for classifying and predicting the severity score from skin images of the patients and performance analysis of the models on training and validation datasets. The accuracy and loss for both classification and scoring for the training and validation sets are plotted. The Scikit python library is utilized for this purpose.

The models were trained with a sample of 492 and 5560 samples from the training dataset, 124 and 1391 samples used for the model validation, and then the model was tested with

Table 6
Machine learning classifier based classification results.

Feature extractor	Metrics (%)	LR	RF	SVM	K-NN	XGBoost
VGG16	Precision	87.19	74.65	85.52	74.49	82.77
	Recall	87.23	74.47	85.53	73.62	82.13
	F-1 score	87.20	74.24	85.51	72.60	82.28
	Test accuracy	87.23	74.47	85.53	73.62	82.13
ResNet50	Precision	64.34	66.21	68.82	61.43	73.56
	Recall	62.55	65.53	68.09	61.28	72.76
	F-1 score	62.89	65.55	68.16	60.89	72.97
	Test accuracy	62.55	65.53	68.09	61.28	72.77
MobileNetV1	Precision	90.74	78.87	88.95	70.56	80.68
	Recall	90.64	78.29	88.94	64.68	80.43
	F-1 score	90.65	77.92	88.88	62.70	80.34
	Test accuracy	90.64	78.30	88.94	64.68	80.43
Inception V3	Precision	83.18	73.55	84.70	69.12	80.14
	Recall	82.98	73.62	84.26	66.81	80.00
	F-1 score	82.99	73.45	84.31	65.98	79.94
	Test accuracy	85.83	73.62	84.26	66.81	80.00
Xception	Precision	86.14	76.41	83.94	72.65	81.62
	Recall	85.96	76.17	83.83	71.06	81.70
	F-1 score	85.99	76.07	83.87	70.85	81.59
	Test accuracy	85.96	76.17	83.83	71.06	81.70

154 and 1738 images, which were split into a proportion of 80:20 for the original and augmented datasets, respectively. The model has been trained with the preprocessed images from the dataset with an epoch count of 100 and the model has been saved as a separate file which is utilized to classify the images obtained from the patient’s skin images. Every epoch is validated for accuracy and loss using the validation set, and then the validated accuracy and loss are tallied up. This process is repeated until all epochs have been processed. This procedure has been applied to the original and augmented datasets respectively. The accuracy and loss curve for the train and validation dataset for the original and augmented dataset have been shown with the plotted graph in Figs. 12 and 13. The graph in Figs. 12 and 13 represents the accuracy, validation accuracy, loss, and validation loss plotted against every epoch while training the proposed model. The validation accuracy achieved at the end of the training model is 91.91% and 98.91%. The accuracy of the model has started to increase after the second epoch and continues exponentially and reaches a maximum accuracy of around 91.91% and 98.91% by the end of the epochs for the original and augmented datasets, respectively, and the validation accuracy is slightly lower than the training accuracy in every epoch. And the loss goes down in a way that is proportional to the number of epochs, reaching its lowest point at the end of training. This demonstrates that the model has received adequate training and that the classification of monkeypox disease can be performed well.

The results for precision and recall, as well as the F-1 score, were determined to be the evaluation criteria for the model. The term “precision” refers to the ability to accurately anticipate the real positive cases across all classes. The recall is a prediction of real positive cases across all classes, and it represents those cases. The F-1 score not only illustrates how well the model works but also demonstrates the relationship that exists between precision and recall. The test data of the original and augmented datasets have been tested, and their accuracy has been measured to be 91.91% and 98.91%, respectively. The precision, recall, F-1 score, and AUC are 91.88%, 91.91%, 91.86%, and 0.9850 respectively for the original dataset and 98.92%, 98.91%, 98.91%, and 0.9997 respectively for the augmented dataset. The batch size of 32 and learning rate of 0.003 has been used to obtain these metric values. All the experimental results for the deep learning models and the proposed model have been tabulated in Table 7. From Table 7, we can see that the overall accuracy of our proposed model is higher than that of all the other deep learning models. The bolded values represent the highest values in each metric.

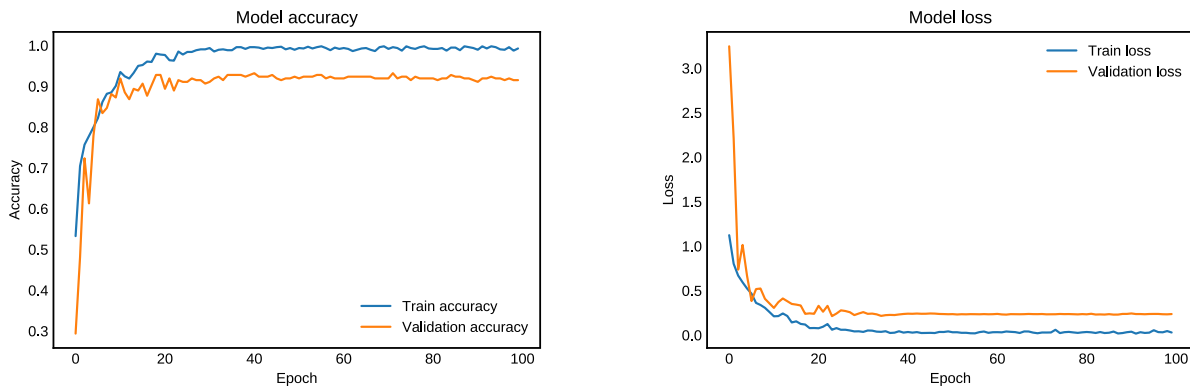


Fig. 12. Proposed model accuracy (left) and loss (right) curve for original dataset.

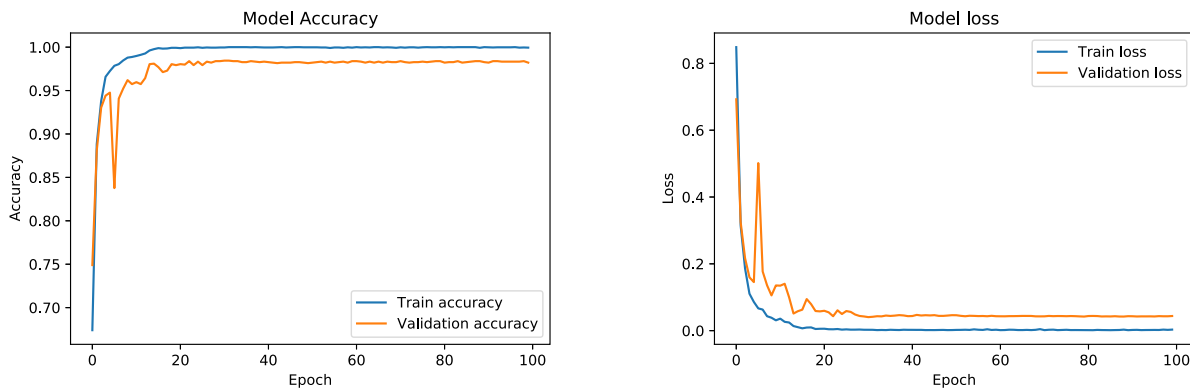


Fig. 13. Proposed model accuracy (left) and loss (right) curve for augmented dataset.

Table 7
Classification results for different deep learning models.

Model	Metrics (%)	Original dataset	Augmented dataset
VGG16	Precision	88.07	94.48
	Recall	88.09	94.43
	F-1 score	88.03	94.44
	Test accuracy	88.09	94.43
	AUC	0.9667	0.9931
ResNet50	Precision	91.41	95.89
	Recall	91.06	95.86
	F-1 score	90.94	95.87
	Test accuracy	91.06	95.86
	AUC	0.9829	0.9962
MobileNetV1	Precision	89.95	96.48
	Recall	89.36	96.44
	F-1 score	89.36	96.44
	Test accuracy	89.36	96.44
	AUC	0.9839	0.9979
Inception V3	Precision	89.47	97.71
	Recall	89.36	97.70
	F-1 score	89.33	97.70
	Test accuracy	89.36	97.70
	AUC	0.9819	0.9989
Xception	Precision	88.59	96.53
	Recall	88.51	96.49
	F-1 score	88.47	96.50
	Test accuracy	88.51	96.49
	AUC	0.9740	0.9989
Proposed model	Precision	91.88	98.92
	Recall	91.91	98.91
	F1 score	91.86	98.91
	Test accuracy	91.91	98.91
	AUC	0.9850	0.9997

Confusion matrices were also generated and this was done so in order to better understand the results. After the model

has been evaluated, the genuinely positive and negative impacts are displayed on a confusion matrix. This provides us with a clear understanding of the flawed model as well as the number of genuine negatives or false positives it produced. Concluding that the majority of the predictions generated by the model are accurate is facilitated by the utilization of the confusion matrix. Fig. 14 provides a representation of the confusion matrix for the original and augmented dataset. However, because many of the images are very similar to one another, it is possible to spot multiple inaccuracies.

We have also plotted the ROC curve for the model performance of both datasets, as shown in Fig. 15. Plots in two dimensions, which are known as ROC curves, are frequently utilized in the process of analyzing and evaluating the performance of classifiers. ROC graphs clearly depict a classifier’s precision or specificity for all feasible classification thresholds. This enables the evaluation and selection of classification models based on unique user needs, which are typically tied to changeable mistake costs and efficiency assumptions. The area under the curve (AUC) is a reflection of the amount of differentiation, whereas the ROC is a probability curve. It provides an indication of how well the classifier can differentiate across various groups. From Fig. 15, we see that our proposed model performs outstandingly on the original and augmented datasets.

We also trained and tested the datasets on a variety of learning rates and batch sizes for the analysis of our proposed model. For this experiment, we have assigned the learning rates of 0.03, 0.003, and 0.0003 and the batch sizes are 8, 16, and 32. After training the proposed model with these parameters, all the obtained results are tabulated in Tables 8 and 9. When we take a look at the Table 8, we can see that a learning rate of 0.003, which was utilized, produces the best results and has the maximum accuracy. As can be seen in Fig. 13, this method maintains the

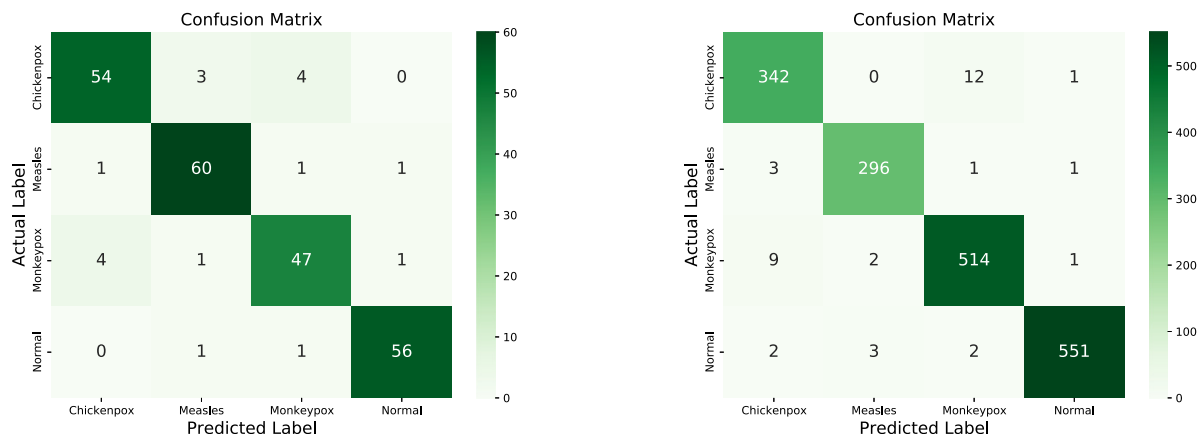


Fig. 14. Confusion matrix on original (left) and augmented (right) dataset.

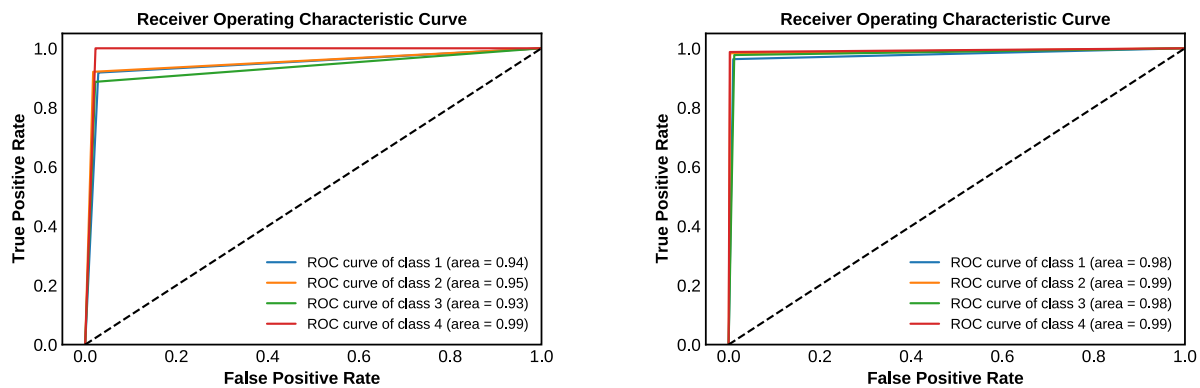


Fig. 15. ROC curve on original (left) and augmented (right) dataset.

highest level of training accuracy over all of the epochs, which ultimately leads to the highest level of training and validation accuracy overall. A learning rate of 0.003 results in a speedier model convergence and a lower final loss, as indicated in the table. This is something that we can see for ourselves. In Table 8, we find that a learning rate of 0.003 greatly surpasses the other two learning rates when examined on the testing set. This finding further validates our findings and provides further support for their accuracy. After doing the studies on the learning rate, we will evaluate each batch size by employing a learning rate of 0.003 for each of them. The F-1 score, precision, recall, accuracy, and area under the curve (AUC) will be compared across all batch sizes as we do the evaluation. When we take a look at the Table 9, we can see that batch sizes of 8 perform well with the limited data with an accuracy of 93.19%, while batch sizes of 16 and 32 perform correspondingly well but lower than the batch size of 8. On the other hand, batch sizes of 32 perform well for the augmented dataset with an accuracy of 98.91%, while batch sizes of 8 and 16 perform around similarly well but lower than the batch size of 32. Looking at the table, we can conclude that the batch size of 8 performs well for the small-scale original dataset, while batch size of 32 perform well for the augmented dataset, respectively.

Finally, we have accomplished another task where a comparison has been done between the original DenseNet-201 and our proposed MonkeyNet model to verify the robustness of our suggested model. In this task, we replaced the final classifier with only one fully connected layer with several fully connected layers plus batch normalization layers and dropout layers from our proposed model and then trained and tested them on the original and augmented datasets. Our proposed MonkeyNet model provides the highest accuracy in each evaluation metric compared

Table 8

Classification results for different learning rates.

Learning rate	Metrics (%)	Original dataset	Augmented dataset
0.03	Precision	90.55	98.01
	Recall	90.49	97.99
	F-1 score	90.46	97.99
	Test accuracy	90.49	97.99
	AUC	0.9750	0.9986
0.003	Precision	91.88	98.92
	Recall	91.91	98.91
	F-1 score	91.86	98.91
	Test accuracy	91.91	98.91
	AUC	0.9850	0.9997
0.0003	Precision	88.91	96.36
	Recall	88.51	96.32
	F-1 score	88.05	96.33
	Test accuracy	88.51	96.32
	AUC	0.9789	0.9967

to the original DenseNet model. The comparative results for the two models are given in Table 10.

In addition, Gradient-weighted Class Activation Mapping, or Grad-CAM for short, is a technique that is often used to depict the “thinking” process that a model goes through, which in turn enables users to have a better understanding of the model’s predictions. This creates a heatmap representation that shows areas of “focus” by analyzing the gradient information going into the last convolutional layer of a given network. Because the model must look there to differentiate between the image classes, the Grad-CAM image may “focus” on the most significant elements of each image class. In our particular scenario, the Grad-CAM ought to concentrate with priority on the contaminated areas of the

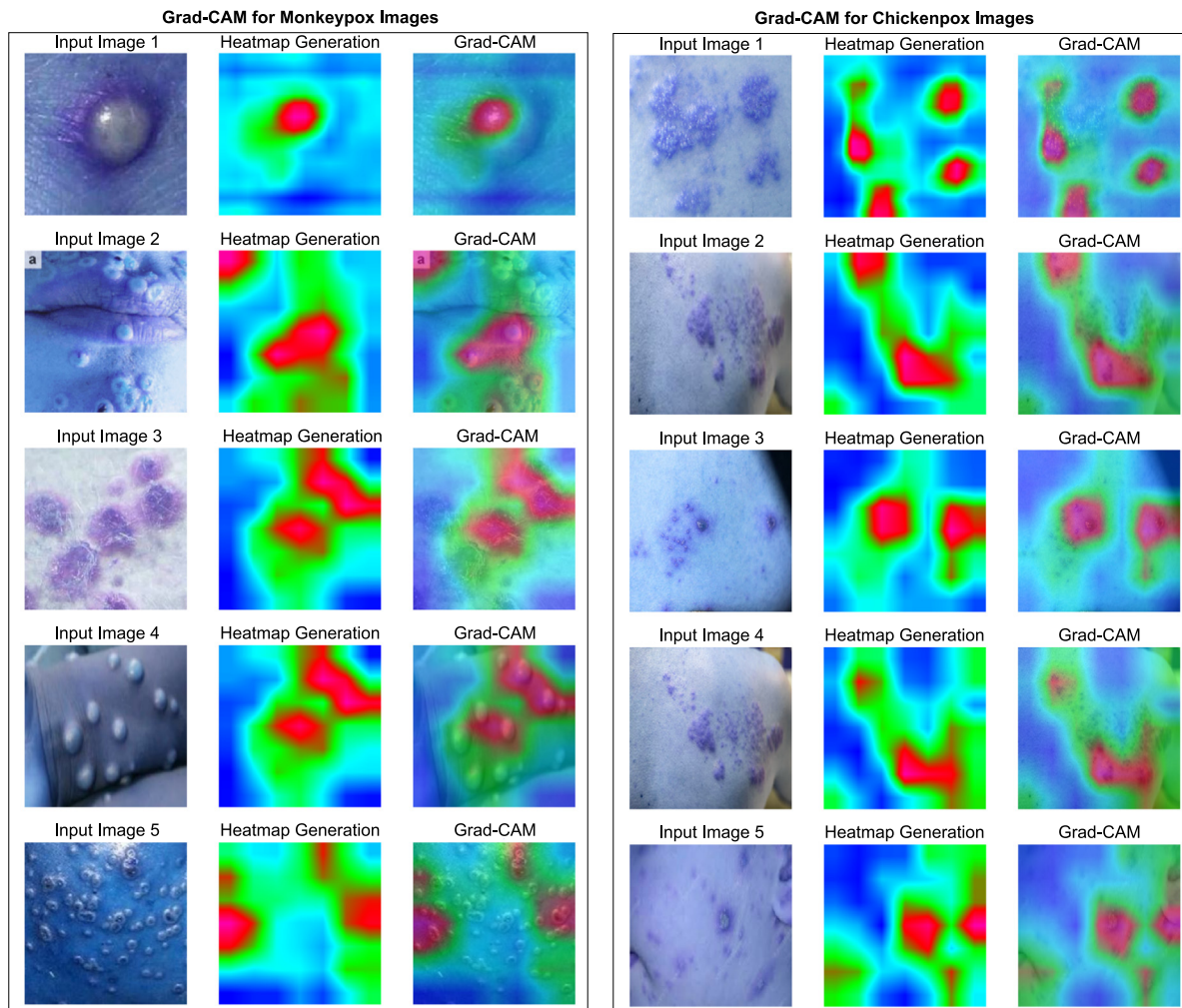


Fig. 16. Grad-CAM of monkeypox (left) and chickenpox (right) image samples.

Table 9
Classification results for different batch sizes.

Batch size	Metrics (%)	Original dataset	Augmented dataset
8	Precision	93.19	97.47
	Recall	93.97	97.47
	F-1 score	93.15	97.47
	Test accuracy	93.19	97.47
	AUC	0.9918	0.9974
16	Precision	92.99	98.80
	Recall	92.77	98.79
	F-1 score	92.73	98.79
	Test accuracy	92.77	98.79
	AUC	0.9882	0.9990
32	Precision	91.88	98.92
	Recall	91.91	98.91
	F-1 score	91.86	98.91
	Test accuracy	91.91	98.91
	AUC	0.9850	0.9997

various disease classes because that is where the primary signs of sickness can be discovered. For the purpose of the Grad-CAM analysis, we have displayed five CAM outcomes for each class. When we take a look at Figs. 16 and 17, we can plainly see that the suggested model is producing accurate predictions and correctly identifying the contaminated region of the skin images. The professionals will find these Grad-CAM images very helpful when they are trying to figure out which parts of the patient’s skin are contaminated.

Table 10
Comparison of results between the proposed and original DenseNet-201 model.

Model	Metrics (%)	Original dataset	Augmented dataset
DenseNet-201	Precision	91.73	97.88
	Recall	91.53	97.87
	F-1 score	91.54	97.87
	Test accuracy	91.53	97.87
	AUC	0.9791	0.9991
	Proposed model	Precision	93.19
	Recall	93.19	98.91
	F-1 score	93.15	98.91
	Test accuracy	93.19	98.91
	AUC	0.9918	0.9997

There has already been one more case study of research that has already made use of the monkeypox classification tasks. The investigations made use of skin images with varying levels of accuracy. The implemented DenseNet-201 framework uses a much larger dataset than many of the strategies that are used now, which use a relatively small set of data. A variety of input images and techniques have been validated using various state-of-the-art methods and techniques, which have resulted in several different images and methodologies. As shown in Table 11, the size of the sample and classification strategies that have been used by the previous researchers differed. There have been a few different models that have been trained and tested over time, and

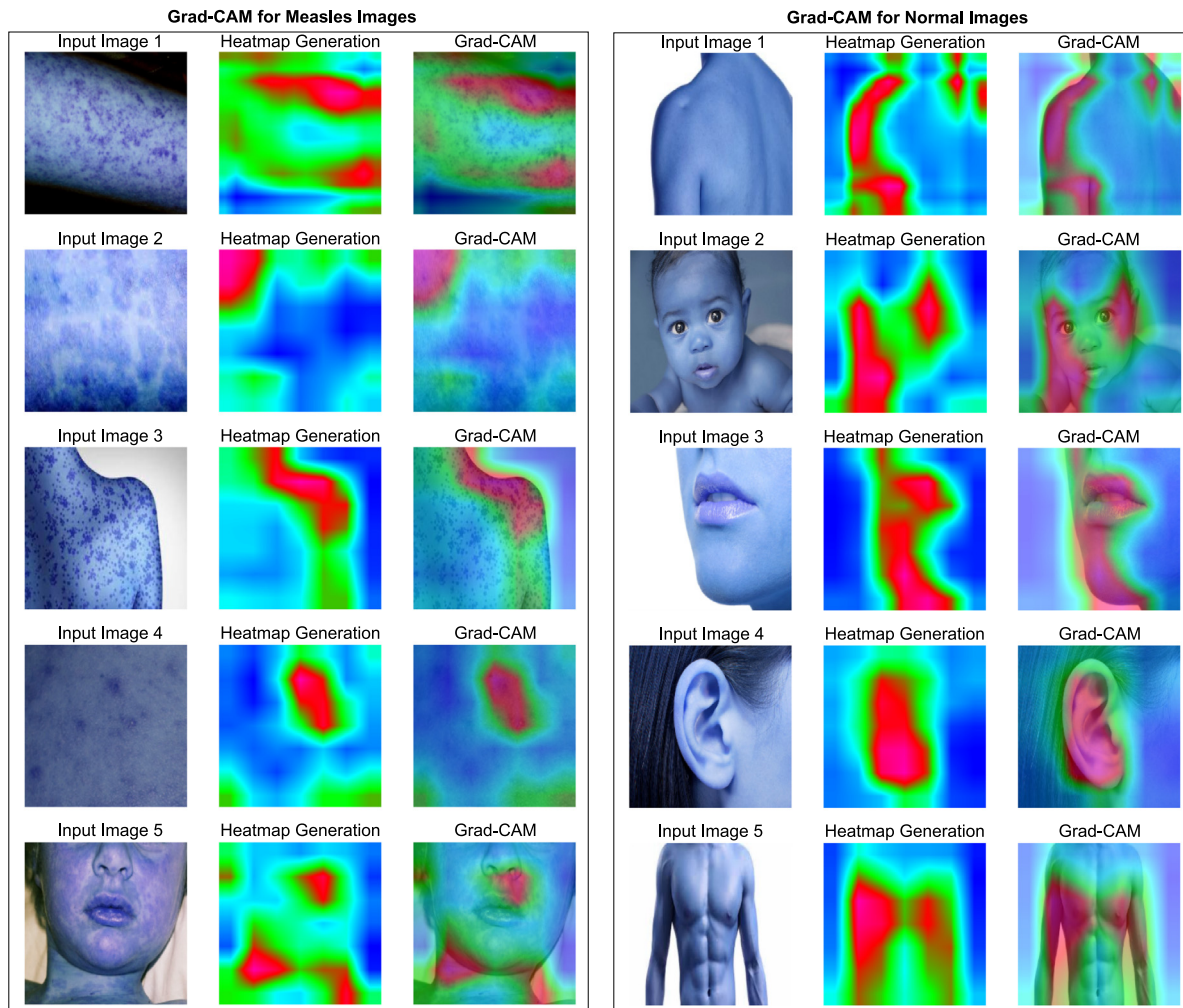


Fig. 17. Grad-CAM of measles (left) and normal (right) image samples.

Table 11 Comparison with the previous works.

Authors	Model	Number of class	Number of samples	Accuracy (%)
(Ali et al., 2022)	ResNet50	3	228	82.96%
(RManjurul Ahsan et al., 2022)	Modified VGG16	2	90	97%
Proposed	Deep CNN	2	1754	88%
		4	770	93.19%
		4	8689	98.91%

most of them have been able to correctly classify monkeypox cases.

One of the advantages of using DenseNet-201 is that it can recognize monkeypox occurrences with an efficiency of 93.19% and 98.91% and a reduced computational cost, making it more accurate than the existing traditional PCR test procedure.

The experimental framework can also be used in conjunction with other approaches for the monkeypox clinical diagnosis. Due to the fact that they are immediately available and have high efficiency, skin images are more effective for patients in severe situations. The model seems to be capable of recognizing monkeypox in a matter of seconds. As a result, a deep learning model

based on skin visual images is highly suggested due to the fact that it is a more reliable learning process. The use of skin images to train deep learning methods has the potential to not only improve the classification of images but also assist clinicians in the fight against an emerging, prevalent diagnosis by permitting them to predict the outcome of a diagnosis.

Furthermore, the following are some of the most significant advantages of this research:

- According to the researchers, skin image-based classification outperforms other types of images in terms of monkeypox disease classification performance. Also, when it comes to classification accuracy, the DenseNet-201-based deep CNN model does better than many other research methods.
- The experimental framework does not necessitate the use of a hand-crafted extraction procedure.
- This research showed that a classification method could be used to help doctors and others in the healthcare system figure out if a patient has monkeypox and spot abnormalities on the skin right away.

6. Conclusion

Monkeypox has global ramifications that will affect our lives and mankind will continue to experience the effects for years, but we will try to fix the problem in different ways, and current circumstances necessitate fresh solutions. In this research paper, an

effective deep learning-based way to find and classify the monkeypox disease pandemic early on has been looked into. In our case, a first-ever database was developed named “MSID” for the detection and classification of monkeypox disease. Then advanced artificial intelligence has been applied that can support the medical diagnosis of monkeypox disease effectively and save lives. The augmentation technique has been applied to the original dataset for the large number of images. In this paper, a modified DenseNet-201-based deep CNN model named “MonkeyNet” has been presented to classify monkeypox from skin images, which is multiclass. To show the classification efficiency of the trained model, evaluation metrics such as precision, recall, F-1 score, accuracy percentages, AUC, and confusion matrices have been presented, and it is possible to show that the predictions made with the images of the test set are sufficiently outstanding. The proposed model has correctly classified the image classes with an accuracy of 93.19% and 98.91% in the multiclass classification of the original and augmented datasets, respectively. In the real world, the model could work well with image-based technology and find and classify the monkeypox disease well.

The findings of this study will improve the knowledge of the medical diagnosis of monkeypox disease. The advanced AI-based detection method would improve the knowledge of the field. In future work, this work will have stepped into a study that will be improved in the future. This experiment could be done on a large number of clinical data and skin images. The current work is able to implement the model in a mobile application that is reliable and will truly support the diagnosis of medical personnel.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (Nos. 2022YFB3303402 and 2021YFF0500901), and the National Natural Science Foundation of China (Nos. 71991464 and 61877056). The authors thankfully acknowledge the web portal organization and journalists who submitted monkeypox images to the online resources.

References

- Adalja, A., & Inglesby, T. (2022). A novel international monkeypox outbreak. *Annals of Internal Medicine*.
- Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International conference on engineering and technology ICET*, (pp. 1–6). Ieee.
- Ali, & Shams Nafisa (2022). Monkeypox skin lesion detection using deep learning models: A feasibility study. arXiv preprint arXiv:2207.03342.
- Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2), 635–640.
- Bala, & Diponkor (2022). Monkeypox skin images dataset (MSID), mendeley data, V3. <http://dx.doi.org/10.17632/r9bfpnvxyr.3>, [Online available at:] <https://data.mendeley.com/datasets/r9bfpnvxyr>.
- Balles, L., Romero, J., & Hennig, P. (2016). Coupling adaptive batch sizes with learning rates. arXiv preprint arXiv:1612.05086.
- Bhandari, A. (2020). *Image augmentation on the fly using keras imagedatagenerator*. Analytics Vidya.

- Bohr, A., & Memarzadeh, K. (2020). The rise of artificial intelligence in healthcare applications. In *Artificial intelligence in healthcare* (pp. 25–60). Academic Press.
- Bradski, G., & Kaehler, A. (2008). *Learning opencv: Computer vision with the opencv library*. O'Reilly Media, Inc.
- CDC (2022). *CDC Works 24/7*. Centers for Disease Control and Prevention, www.cdc.gov. <https://www.cdc.gov/>.
- Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., & Chen, K. (2015). Xgboost: extreme gradient boosting. *R Package Version 0.4-2*, 1(4), 1–4.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).
- DermNet, N. Z. (2022). DermNet NZ – All about the skin | DermNet NZ. DermNet NZ – All about the Skin | DermNet NZ; dermnetnz.org. <https://dermnetnz.org/>.
- Desai, S. B., Pareek, A., & Lungren, M. P. (2020). Deep learning and its role in COVID-19 medical imaging. *Intelligence-Based Medicine*, 3, Article 100013.
- Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, 35(5–6), 352–359.
- Duds, D., Page, D., Care, D. P., Diet, K., Course, F. W. L. M., Boy, B., & Virilis, S. (2022). COVID-19 vaccine informed consent. In *Humor*.
- Fatima, N., & Mandava, K. (2022). Monkeypox-a menacing challenge or an endemic? *Annals of Medicine and Surgery*, 79, Article 103979.
- Folego, G., Weiler, M., Casseb, R. F., Pires, R., & Rocha, A. (2020). Alzheimer's disease detection through whole-brain 3D-CNN MRI. *Frontiers in Bioengineering and Biotechnology*, 8, Article 534592.
- Fong, S. J., Li, G., Dey, N., Crespo, R. G., & Herrera-Viedma, E. (2020). Finding an accurate early forecasting model from small dataset: A case of 2019-ncov novel coronavirus outbreak. arXiv preprint arXiv:2003.10776.
- Google (2022). Google. www.google.com. <https://www.google.com/>.
- Haloi, M., Rajalakshmi, K. R., & Walia, P. (2018). Towards radiologist-level accurate deep learning system for pulmonary screening. arXiv preprint arXiv:1807.03120.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4), 18–28.
- Hegde, G., Pharral, T., Jahagirdar, S., Nargund, V., Tabib, R. A., Mudanagudi, U., & Dhiman, A. (2021). DeepDNet: Deep dense network for depth completion task. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2190–2199).
- Ho, Y., & Wookey, S. (2019). The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8, 4806–4813.
- Hossain, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
- IAC (2022). Immunization action coalition (IAC): Vaccine information for health care professionals. www.immunize.org. <https://www.immunize.org/>.
- Ketkar, N. (2017). Introduction to keras. In *Deep learning with python* (pp. 97–111). Berkeley, CA: A Press.
- Khemasuwana, D., & Colt, H. G. (2021). Applications and challenges of AI-based algorithms in the COVID-19 pandemic. *BMJ Innovations*, 7(2).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Koenig, K. L., Beý, C. K., & Marty, A. M. (2022). Monkeypox 2022 identify-isolate-inform (3I): A tool for frontline clinicians for a zoonosis with escalating human community transmission. *One Health*, Article 100410.
- Kumar, N., Acharya, A., Gendelman, H. E., & Byrareddy, S. N. (2022). The 2022 outbreak and the pathobiology of the monkeypox virus. *Journal of Autoimmunity*, Article 102855.
- Kuo, C. C., Chang, C. M., Liu, K. T., Lin, W. K., Chiang, H. Y., Chung, C. W., & Chen, K. T. (2019). Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning. *NPJ Digital Medicine*, 2(1), 1–9.
- Lella, K. K., & Pja, A. (2021). A literature review on COVID-19 disease diagnosis from respiratory sound data. arXiv preprint arXiv:2112.07670.
- Lewin, S. (2010). Gender differences in emerging infectious diseases. In *Principles of gender-specific medicine* (pp. 497–515). Academic Press.
- Li, B., Wu, F., Lim, S. N., Belongie, S., & Weinberger, K. Q. (2021). On feature normalization and data augmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12383–12392).

- Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random forest. In *International conference on information computing and applications* (pp. 246–252). Berlin, Heidelberg: Springer.
- Myszczyńska, M. A., Ojames, P. N., Lacoste, A., Neil, D., Saffari, A., Mead, R., & Ferraiuolo, L. (2020). Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. *Nature Reviews Neurology*, 16(8), 440–456.
- Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 24(3), 1207–1220.
- NHS (2022). *The NHS website*. UK: Nhs, www.nhs.uk. <https://www.nhs.uk/>.
- Noreen, N., Palaniappan, S., Qayyum, A., Ahmad, I., Imran, M., & Shoaib, M. (2020). A deep learning model based on concatenation approach for the diagnosis of brain tumor. *IEEE Access*, 8, 55135–55144.
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
- Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Acharya, U. R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*, 121, Article 103792.
- Pei, S. C., & Lin, C. N. (1995). Image normalization for pattern recognition. *Image and Vision Computing*, 13(10), 711–723.
- Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., & Ng, A. Y. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225.
- Reed, K. D., Melski, J. W., Graham, M. B., Regnery, R. L., Sotir, M. J., Wegner, M. V., & Damon, I. K. (2004). The detection of monkeypox in humans in the Western Hemisphere. *New England Journal of Medicine*, 350(4), 342–350.
- Reynolds, M. G., McCollum, A. M., Nguete, B., Shongo Lushima, R., & Petersen, B. W. (2017). Improving the care and treatment of monkeypox patients in low-resource settings: applying evidence from contemporary biomedical and smallpox biodefense research. *Viruses*, 9(12), 380.
- RManjural Ahsan, M., Ramiz Uddin, M., Farjana, M., Nazmus Sakib, A., Al Momin, K., & Akter Luna, S. (2022). Image Data collection and implementation of deep learning-based model in detecting monkeypox disease using modified VGG16. arXiv e-prints, arXiv:2206.
- Roy, K., Chaudhuri, S. S., Ghosh, S., Dutta, S. K., Chakraborty, P., & Sarkar, R. (2019). Skin disease detection based on different Segmentation Techniques. In *2019 International conference on opto-electronics and applied optics (Optronix)* (pp. 1–5). IEEE.
- Sandeep, R., Vishal, K. P., Shamanth, M. S., & Chethan, K. (2022). Diagnosis of visible diseases using CNNs. In *Proceedings of international conference on communication and artificial intelligence* (pp. 459–468). Singapore: Springer.
- Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(221).
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Simpson, K., Heymann, D., Brown, C. S., Edmunds, W. J., Elsgaard, J., Fine, P., & Wapling, A. (2020). Human monkeypox—After 40 years, an unintended consequence of smallpox eradication. *Vaccine*, 38(33), 5077–5081.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
- Velasco, J., Pascion, C., Alberio, J. W., Apuang, J., Cruz, J. S., Gomez, M. A., & Jorda Jr, R. (2019). A smartphone-based skin disease classification using mobilenet cnn. arXiv preprint arXiv:1911.07929.
- Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 141, 61–67.
- Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11).
- Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W. (2019). A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11127–11135).