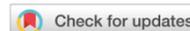




## Monkeypox Detection Using Hyper-Parameter Tuned Based Transferable CNN Model

V. Gokula Krishnan<sup>1\*</sup>, B. S. Liya<sup>2</sup>, S. Venkata Lakshmi<sup>3</sup>, K. Sathyamoorthy<sup>4</sup>  
and Sangeetha Ganesan<sup>5</sup>



<sup>1</sup>Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Thandalam, Chennai, Tamil Nadu, India; <sup>2</sup>Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India; <sup>3</sup>Department of AIDS, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India; <sup>4</sup>Department of Computer Science and Engineering, Panimalar Engineering College, Poonamallee, Chennai, Tamil Nadu, India; <sup>5</sup>Department of AIDS, R M K College of Engineering and Technology, Kavaraipettai, Tamil Nadu, India

E-mail/Orcid Id:

VGK,  gokul\_kris143@yahoo.com,  <https://orcid.org/0009-0005-6819-6729>; BSL,  bsliyapraveen@gmail.com; SVL,  venkatalakshmi227@gmail.com; KS,  pitsathyamoorthy@gmail.com; SG,  gsangeethakarthik@gmail.com

### Article History:

Received: 25<sup>th</sup> Jul., 2023

Accepted: 17<sup>th</sup> Sep., 2023

Published: 30<sup>th</sup> Sep., 2023

### Keywords:

Chickenpox, Convolutional Neural Network, Modified Gear and Steering based Model, Monkeypox Virus, Rider Optimization Algorithm, Ttransferable

### How to cite this Article:

V. Gokula Krishnan, B. S. Liya, S. Venkata Lakshmi, K. Sathyamoorthy and Sangeetha Ganesan (2023). Monkeypox detection using hyper-parameter tuned based transferable CNN model. *International Journal of Experimental Research and Review*, 33, 18-29.

DOI: <https://doi.org/10.52756/ijerr.2023.v33spl.003>

**Abstract:** The reemergence of Monkeypox, a communicable illness resulting from the Monkeypox virus, has raised apprehensions about a potential swift global pandemic similar to the COVID-19 epidemic as COVID-19 infections diminish globally. The prompt emphasizes the criticality of prompt action within communities to mitigate the development of the phenomenon. The timely identification and accurate categorization of Monkeypox cutaneous manifestations are crucial for the successful implementation of containment strategies. This paper presents a novel methodology for detecting Monkeypox by utilizing a transferrable Convolutional Neural Network (CNN) model that has been optimized utilizing hyper-parameter tuning techniques. The proposed methodology initiates by improving the quality of the original Monkeypox images, with a specific emphasis on boosting edge details to increase visual clarity. Texture qualities are obtained through an energy layer, enhancing distinctive traits. Our methodology's cornerstone is utilizing the Hyper-parameter-based transferable Convolutional Neural Network (HPT-TCNN), specifically designed to enhance classification accuracy.

In contrast to traditional methods, we enhance the architectural design by replacing the pooling layer with a configuration comprising three convolutional layers and one energy layer. The hyper-parameter tuning procedure is optimized by employing the Optimisation Algorithm known as MGS-ROA. In order to enhance the process of model training and validation, we have assembled the "Monkeypox Skin Lesion Dataset (MSLD)," which consists of a collection of images depicting human skin lesions produced by Monkeypox. The dataset in question is vital in evaluating and improving our methodology. In a comparison analysis conducted on other deep learning models, the suggested model has superior performance compared to other models, obtaining a notable accuracy, sensitivity, and specificity, all reaching a value of 93.60%. The outstanding performance shown in this study highlights the methodology's effectiveness in adequately classifying skin lesions associated with Monkeypox. This approach shows potential for physicians and healthcare workers since it facilitates early detection, a crucial factor in preventing the spread of Monkeypox.

### Introduction

The etiological agent responsible for Monkeypox, known as the Monkeypox virus, is classified under the P

family and the orthopoxviral genus. The variola virus is known to induce the development of smallpox within individuals belonging to the same familial lineage

\*Corresponding Author: [gokul\\_kris143@yahoo.com](mailto:gokul_kris143@yahoo.com)



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

(Samaranayake & Anil, 2022; Zumla et al., 2022). The aetiology of bovine smallpox may be attributed to the Cowpox virus, whereas the Vaccinia virus is employed in developing smallpox vaccines. It is noteworthy that, contrary to its nomenclature, the virus responsible for the manifestation of Monkeypox has undergone evolutionary development in rodent species. The nomenclature "Monkeypox" (Adalja & Inglesby, 2022) was used in 1958 upon the initial discovery of the virus during two distinct instances of outbreaks, whereby the exhibited symptoms bore a resemblance to those employed in scientific investigations. The discovery that the virus may infect people was made in 1970 by researchers. Following this, the incidence of Monkeypox has been relatively infrequent throughout Africa. This phenomenon has predominantly been documented in West and Central Africa, areas distinguished by vast tropical rainforests (Otu et al., 2022; Català et al., 2022). Nevertheless, during the past several years, the disease has exhibited an increased scope, impacted a more heterogeneous demographic and manifested in a broader geographic expanse compared to previous occurrences. Given the profound ramifications of the pandemic, there has been a heightened level of attentiveness in the surveillance of Monkeypox occurrences despite the absence of an epidemic scale thus far (Zhang et al., 2021).

Monkeypox is characterized by a rash that develops over 1-5 days. The rash initially appears on the face and subsequently spreads to other body regions. Lesions in the vaginal region, eyes, and intraoral mucosa have been reported in certain patients (Suganyadevi et al., 2022). The rashes caused by this condition can seem like those caused by chickenpox, leading to misdiagnoses. These rashes start out as water-filled blisters but eventually heal into crusty areas. Some people get hundreds of blisters all over their bodies, whereas others only have a few (Ayca et al., 2022). Lesions may join together to form widespread rashes on the skin's surface in severe situations. In 2-4 weeks, contingent on the harshness of the sickness, the rashes subside, and the disease recovers.

CNNs are widely employed in academic research in learning (Li & Du, 2021), which is revealed when we look at the most cutting-edge technologies in image classification. In most cases, images provide the input data for CNN, a deep-learning model. It records the results of several processes on the picture in order to categorize potential future judgements. LeNet, initially proposed by Yann LeCun in 1988 and refined until 1998 (Savas, 2022), was the first structure for a convolutional neural network. Many industries, including NLP and

biology, make use of CNN algorithms, particularly in the realm of picture and sound processing. The best consequences have been achieved, particularly in image processing. The error rate was decreased to 0.23% using CNN on the MNIST dataset.

The skin sores caused by Monkeypox are the disease's most noticeable symptoms. In order to start therapy as soon as possible, it is crucial to quickly distinguish skin lesions from other lesion diseases (Saanat et al., 2022). Mobile devices should be able to tell the difference between Monkeypox and other illnesses that cause skin lesions, reducing the likelihood of transmission. The end user may determine whether or not they have Monkeypox by taking a picture with their phone and running it through the transfer learning-trained TFLite model (Yue et al., 2022). Since Monkeypox has begun to spread rapidly worldwide, we hope that our research will help scientists swiftly and accurately categorize the impact of this virus on skin lesions (Uysal, 2023).

In modern CAD systems, deep learning methods have been integrated to improve the accuracy of skin lesion identification and categorization. When it comes to diagnosing Monkeypox, computer-aided design (CAD) approaches based on image processing are on the rise. The following are some of the benefits of our proposed method for early identification of Monkeypox:

- We present an HPT-TCNN for Monkeypox classification, with hyper-parameter tuning handled by an MGS-ROA model.
- In the suggested HPT-TCNN architecture, we use the energy layer (EL). By doing so, we can maintain textural information and restrict the model's general capacity for learning.

Here is how the break of the paper is structured: In Section 2, we describe the relevant literature, and in Section 3, we offer a summary of the suggested model. In Section 4, the trial analysis and validation are presented, and in Section 5, the conclusion is illustrated.

### Related works

Uysal et al. (2023) have created a hybrid AI system that can identify photos of Monkeypox on the skin. Images of skin were taken from a publicly available picture resource. The chickenpox measles, classes make up the multi-class structure of this dataset. The original dataset had an uneven distribution of data across classes. Many data augmentation and pre-processing methods were used to correct this discrepancy. Following these steps, state-of-the-art deep learning models, including CSPDarkNet, RepVGG, were used to search for signs of Monkeypox. By combining the two best-performing

models with the (LSTM) model, this study developed a novel hybrid deep-learning model with improved classification results. The built and suggested hybrid AI system for Monkeypox detection achieved an impressive 87% test accuracy and a Cohen's kappa value of 0.8222.

Altun et al. (2023) envision using deep learning to rapidly and precisely identify monkeypoxes from skin lesions during the pandemic. Tools for hyperparameter optimisation and transfer learning were made available to support deep learning techniques. Through careful tuning of the hyperparameters, we created a hybrid function learning model for use within the CNN architecture. Models, ResNET50, and Xception were used in the implementation. In this research, we compared and contrasted different methods using loss and F1-score. The optimized hybrid MobileNetV3-s model obtained the best results, which had an F1-score of 0.98, an average. In this research, a bespoke CNN model was constructed using convolutional neural networks, hyperparameter optimization, and a hybrid function transfer learning model, yielding impressive results. Our proposed custom CNN model architecture demonstrates the efficiency and effectiveness of deep learning techniques for classification and discrimination.

Using computer vision, (Almufareh et al., 2023) propose a smarter and more secure alternative to conventional ways of diagnosing MPX by analyzing photographs of skin lesions. The suggested approach utilizes deep learning strategies to identify MPXV positivity in skin lesions. We test our approach on two datasets, including images and descriptions of Monkeypox lesions: the (MSID). Multiple deep learning models' performance was measured by their sensitivity to change, specificity, and overall accuracy. Results from using the suggested approach to identify Monkeypox have been very encouraging, showing that it is probable to be used on a large scale. This clever and low-cost option may be put to good use even in underdeveloped regions where there is a need for more laboratory facilities.

To identify the attendance of the Monkeypox virus in skin lesion photos, (Pramanik et al., 2023) offer an ensemble learning-based system. First, we focus on fine-tuning the Monkeypox dataset using one of three pre-trained base Xception, or DenseNet169. We also use the deep models to extrapolate probabilities that are fed into the ensemble framework. To learn information collected from the sum rule-based ensemble, we offer a Beta function-based normalization scheme of probabilities to combine the results. Using a five-fold cross-validation configuration, the framework is extensively tested on a

publicly accessible Monkeypox skin lesion dataset. The average values for the representation's accuracy, precision, recall, and F1 are 93.39, 88.91, 96.78, and 92.35.

Yasmin et al. (2023) have set out to solve this issue by using machine learning and image processing techniques to create a model for diagnosing Monkeypox. Data augmentation methods have been used to achieve this goal and prevent the model from becoming overfit. Six distinct Deep Learning (DL) models were then trained using the pre-processed dataset using the transfer-learning approach. We settled on the best one after comparing each model's precision, recall, and accuracy performance matrices. After doing fine-tuning on the best-performing model, a new model named "PoxNet22" was suggested. Compared to other approaches, PoxNet22's categorization of Monkeypox is superior since it achieves perfect results in accuracy. Clinicians will find the findings of this study instrumental in the classification and diagnosis of Monkeypox disease.

Ariansyah et al. (2023) presented an image classification to differentiate measles. In order to model images, researchers employed a deep learning approach based on learning. With transfer learning, a model learned on one dataset may be applied to another. This enabled the model to generalize insights from one data set to another. Because deep learning is so effective for recognizing patterns in similar photos, researchers have proposed using it to forecast fresh data. Consequently, the VGG-16 model achieves a respectable 83.333% accuracy at epoch = 15.

### Proposed work

Monkeypox images are fed into a DL model optimized using the hybrid rider optimization approach for binary classification.

### Dataset

The fast spread of Monkeypox to more than 65 countries has caused public health officials to worry. Stopping its fast development requires prompt clinical identification. However, there needs to be more ready access to a large sum of (PCR) tests and other biochemical assays (Nolen et al., 2016). Computer vision techniques might help in detecting Monkeypox from pictures of skin lesions. However, at present, no such data is available. Therefore, the "(MSLD)" is created by compiling and analyzing images from various online resources (such as websites, portals, and public case reports). To distinguish Monkeypox patients from similar non-Monkeypox instances, the "Lesion Dataset" was created. Due to their resemblance to the Monkeypox rash

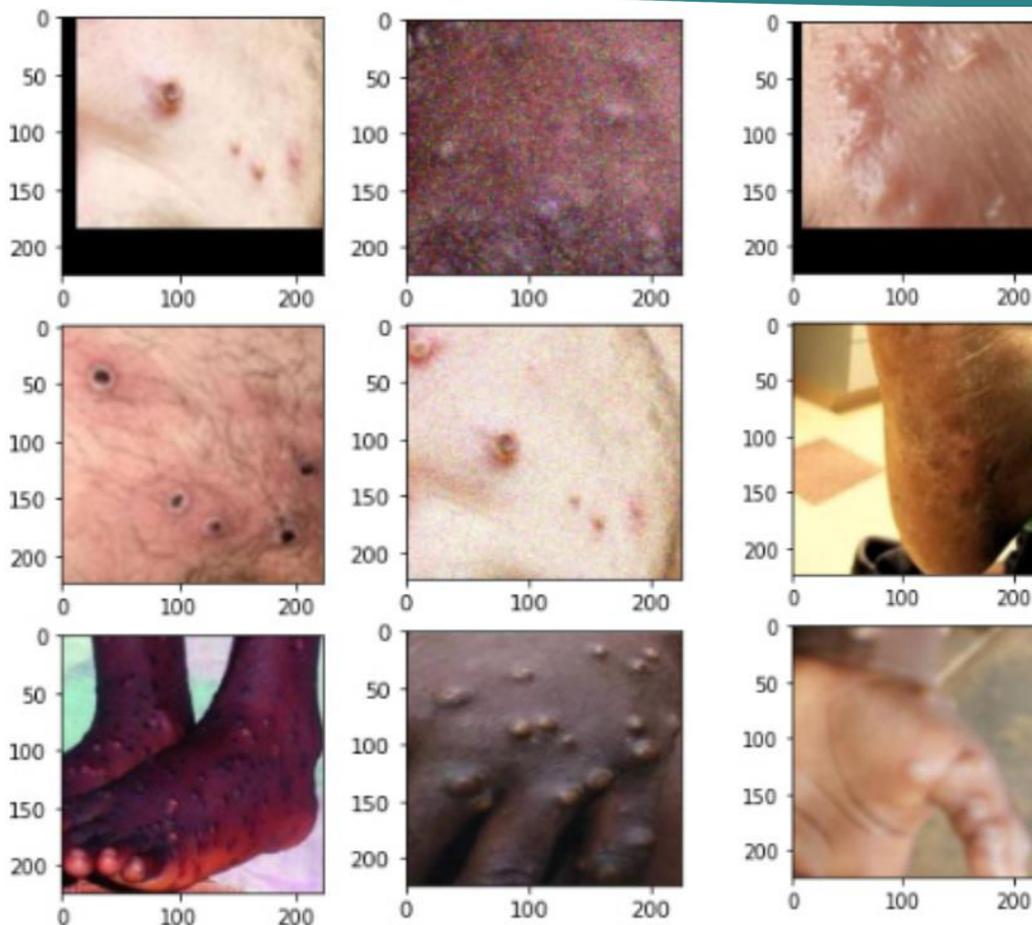


Figure 1. Model dataset

and first-state pustules, we included lesion images of the "Chickenpox" category to create a classifier. For example, the Monkeypox Skin Lesion Dataset has 228 images, 102 of which are labelled as "Monkeypox" and the residual 126 as "Others," which comprise suitcases of non-Monkeypox. Raw data samples are exposed in Figure 1.

**Classification using DL network architecture**

We introduce the HPT-TCNN framework for classifying cases of Monkeypox. The proposed deep CNN considers three critical aspects of the picture: First, certain description filters can still discover them if their size is the same as the size filter's mask. Second, distinct input image regions might use distinct forms or patterns. Convolution of the full source picture is another way to define such models. Finally, the max-pooling layer relies heavily on down-sampled pixels, which do not alter the original image's overall form. Figure 2 depicts the suggested HPT-TCNN architecture for the categorization of Monkeypox.

In the proposed HPT-TCNN, a third convolution layer regulates the EL after two convolution layers and a pooling layer. After the softmax layer is added. Elastic net (EL) summarises the feature maps generated by the

output of the corrected activation layer. For each feature map, you get a number that stands for the energy response of a filter bank. This layout reduces memory and computational requirements and increases efficiency while learning texture functions, EL speed and processing time. The primary motivation for implementing this layer is to flow. After the final pooling layer, the output of EL is flattened and sent to the layer. Because of this link, a new, simplified vector representing the image's contours and textures is generated and sent to the fully linked layer. Equation (1) delivers a size:

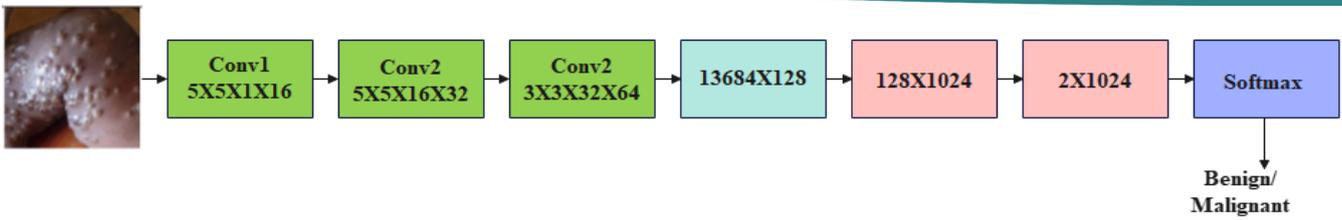
$$Output = \frac{I_a - I_b + 2S}{\varrho + 1} \dots\dots\dots(1)$$

where  $I_a$  and  $I_b$  characterize filter size correspondingly,  $S$  denotes the stuffing, and  $\varrho$  is the stride value.

After that, 16 and 32 channels are produced by the layers, with a kernel scope of 5 5 for the first two layers. Using 33 kernel and 64 output channels, the convolution layer is analyzed as a potential intermediary layer for extracting texture attributes. From the convolution layer, we can only parameters, which we do so by solving the following equations:

$$\xi_v = \zeta_v \times (I_k \times \varrho + 1) \dots\dots\dots(2)$$

$$\xi_v = \zeta_v + X_k \times \varrho \times \zeta_v \dots\dots\dots(3)$$



**Figure 2. The framework of the projected HPT-TCNN for Monkeypox classification**

Where  $\xi_v$  symbolizes the CNN parameters,  $l_k$  characterizes the kernel extent, and  $\zeta_v$  denotes the channel quantity.

The neuron output linked to the input is intended at each convolution layer. The computation is the dot product of the object's mass and the width of its smallest input field. The result of the layer is a 16-kernel 32-by-32-by-16 matrix. The output of the neurons in the first is calculated using Equation (4):

$$S_\vartheta = \sum_\vartheta C_\vartheta \times T_\vartheta + P_\vartheta \dots \dots \dots (4)$$

where  $S_\vartheta$  characterizes the output feature maps,  $C_\vartheta$  epitomizes the maps, and T denotes the weighted map. After that, the layer's output is transformed into an energy descriptor. After the descriptor. It performs similarly to a texture explanation for a cluttered, thick surface. In Equation (5), we see the relationship stated as:

$$EL(\xi, \vartheta) = \rho[\sum_{i=1}^j T_i^\omega \vartheta_i + P] \dots \dots \dots (5)$$

where  $EL(\xi, \vartheta)$  characterizes the EL layer, j characterizes the input, and T signifies the EL weighted vector. The link between the EL and FC layers is substantially smaller compared to the parameters. In addition, EL remembers the energy state of the previous layer and acquires new knowledge as signals travel both forward and backwards in time. In addition to enhancing the learning capacity and simplifying the projected system, EL helps lower the vector scope of the following FC layer. To determine which EL parameters may be taught, use Equation (6):

$$\xi_{EL} = \eta^m \times \eta^{m-1} \dots \dots \dots (6)$$

where  $\xi_{EL}$  is the EL learnable limits,  $\eta^m$  is FC layer neuron, and  $\eta^{m-1}$  is the preceding FC layer neuron.

Between the convolution layer and the rectified linear unit (ReLU) layer, a batch normalization and activation function is utilized to expedite the shift, which can be eliminated by normalization. The deviation can be normalized to accomplish this. Mean and Variance are strongminded using Equations (7) and (8) used in the bulk normalization computation.

$$\tau_Q = \frac{1}{n} \sum_i^n l_i \dots \dots \dots (7)$$

$$v_Q = \frac{1}{n} \times \sum_i^n (l_i - \tau_Q)^2 \dots \dots \dots (8)$$

where  $\tau_Q$  and  $v_Q$  characterize the mean and alteration correspondingly, n is the size of  $l_i$  element of features. Normalization is intended in Equation (9) as:

$$\lambda_i = \frac{\vartheta_i - \tau}{\sqrt{v^2 + \phi}} \alpha + A \dots \dots \dots (9)$$

Where a and A are the two starting parameters of the output layers that can be learned. The activation function for the ReLU layer can be found in Equation (10) and its output can be found in Equation (11), as:

$$\lambda_{i,j,k} = \max(0, \vartheta_{i,j,k}) \dots \dots \dots (10)$$

$$\lambda_{ReLU} = \text{ReLU}(\text{Bnorm}(\text{Conv}(w, x))) \dots \dots (11)$$

Where  $\lambda_{i,j,k}$  represents the output features and  $\vartheta_{i,j,k}$  symbolizes the feature of the input element. Afterwards, The control network is over fit because the pooling layer averages out data from the feature maps, weights, and computations. The formula is as follows, and it is used to determine the maximum pooling layer:

$$M_{pool} = \max(0, \sum_Q \vartheta^{k-1} T_\vartheta) \dots \dots \dots (12)$$

Where  $M_{pool}$  signifies the production feature maps,  $\vartheta$  designates the maps, Q means the pooling size, and T stands for the maximum pooling layer for the kernel vector. In this study, we employ a maximum of two pooling layers, each having a kernel size of two by two.

To avert the model from overfitting the training data, the layer is utilized during the weighted update phase to repeatedly eliminate a sample of accidental parameters. To prevent overfitting training data, drop editing is performed during the weighted update phase to periodically eliminate a subset of random parameters. Over-compatibility of training data is especially problematic in FC layers since they include the most network-wide properties. The dropout layer is the result since the FC layer is established later. The softmax layer is used as a loss-based classifier. Softmax accepts and one, between [0, 1]. In Equation (13), the loss function is expressed mathematically as:

$$k_l = \delta_j + \log \sum_i \exp(\delta_i) \dots \dots \dots (13)$$

Where  $k_l$  signifies the entire loss and  $\delta_i$  consuming the class d which is i-th course element. The classifier's goal is to reduce the likelihood between the true label and its projected counterpart, as computed by the function in Equation (14):

$$\lambda_i = \frac{\exp^{\delta_j}}{\sum_i \exp(\delta_j)} \dots\dots\dots(14)$$

The next step of HPT-TCNN, hyper-parameter tweaking, is described below once this phase is complete. Input and output dimensions for the proposed network are listed in Table 1.

**Table 1. Projected HPT-TCNN structure layers**

Types	Input Size	Padding	Kernel Size to Form Respectively Feature Map	Stride	Output Size
EL	16×16×64	-	-	-	128×1
Dropout	128×1	-	-	-	128×1
FC1	128×1	-	-	-	1024×1
Dropout	1024×1	-	-	-	1024×1
FC2	1024×1	-	-	-	2×1
Convolutional Layer 1	64×64×1	[1 1 1 1]	5×5	[1 1]	62×62×16
Max Pooling Layer 1	62×62×16	[1 1 1 1]	2×2	[2 2]	32×32×16
Convolutional Layer 2	32×32×16	[1 1 1 1]	5×5	[1 1]	30×30×32
ReLU					
Max Pooling Layer 2	30×30×32	[1 1 1 1]	2×2	[2 2]	16×16×32
Convolutional Layer 3	16×16×32	[1 1 1 1]	3×3	[1 1]	16×16×64
ReLU					
Classification Layer	-	-	-	-	-

**Hyper-parameter tuning process using MGS-ROA**

The cluster of riders is progressing toward the goal, which inspired the algorithm ROA (Binu and Kariyappa, 2019). Let's pretend many groups of cyclists are headed in the same general direction. The cyclists are divided into four groups, each including an equal number of riders. Bypass riders follows, overtakers and attackers make up the four groups of riders. Each faction has its unique strategy to reach its goal. The rider attempting to pass the leader does so by focusing on the leader's position relative to his own and then moving in that direction. The assailant arrives quickly and stands in the rider's path to the objective. Each rider should follow the steps outlined below, which comprise this algorithm.

Group and rider parameters are set to their default values. Riding groups (RG) are first set up with a random distribution of riders among four groups. Eq. (15) is a representation of the cluster initialization. Here, the number of riders (RN) is synonymous with the riders' group (RG). The coordinate number (CN) represents the number of dimensions.  $L_{it}(c, d)$  also represents the location of the  $c^{th}$  rider. The total number of riders may be determined by adding up the riders in each section, and the related equation is given by the symbol Eq. (16).

$$L_{it} = \{L_{it}(c, d)\}; 1 \leq c \leq RN; 1 \leq d \leq CN \dots\dots(15)$$

$$RN = ByR + Fol + Ovr + Att \dots\dots\dots(16)$$

In Eq. (16), *ByR*, *Fol*, *Ovr*, and *Att* are the “bypass rider, supporter, overtaker and attacker”, correspondingly. Furthermore, parameters which are cluster loading. The steering angle of the vehicle at a period is given in Eq. (17), which  $ST_{c,d}^{it}$  is the angle of  $c^{th}$

rider’s vehicle, and the exact equation is signified in Eq. (18).

$$ST_{it} = \{ST_{c,d}^{it}\}; 1 \leq c \leq RN; 1 \leq d \leq CN \dots\dots(17)$$

$$ST_{c,d} = \begin{cases} \theta_c; & \text{if } d \neq 1 \\ ST_{c,d-1} + \varphi; & \text{if } d \neq 1 \text{ and } ST_{c,d-1} + \varphi \leq 360 \dots\dots(18) \\ ST_{c,d-1} + \varphi - 360; & \text{otherwise} \end{cases}$$

In addition, from Eq. (18),  $\theta_c$  and  $\varphi$  is the site angle of the  $c^{th}$  rider’s vehicle. Based on the sum of riders and the extreme angle 360°, the site angle of  $c^{th}$  rider’s vehicle is strongminded, which is given in Eq. (19). In find and it is meant in Eq. (20).

$$\theta_c = c * \frac{360}{RN} \dots\dots\dots(19)$$

$$\varphi = \frac{360}{CN} \dots\dots\dots(20)$$

Counting how often something works In Eq. (21), where  $Lo_{ta}$  is the target's position, we see that the success rate is specified in terms of the distance mid the target and the rider's current location. For the rider's success rate to increase as a function of distance, the rider must shorten the distance.

$$sur_c = \frac{1}{\|L_c - Lo_{ta}\|} \dots\dots\dots(21)$$

Finding the top finisher requires heavily considering the rider's achievement rate. The rider with the highest achievement rate will be considered the leader since he or she is more likely to reach the goal. Each time period has a different leading rider since the location of the objective

moves around. So, it stands to reason that the rider with the highest success record will take the reins. Where are the riders now? Each time around, the riders' relative positions inside their respective clusters are revised to determine who has taken the lead and ultimately won the race. The following procedures inform the rider's position inside each cluster based on the characteristics. Update sidestep: Since these riders avoid following the leaders, their update process is represented by Eq. (22), which indicates that they choose a different route. Here, 0 and 1 represent two random numbers chosen at random. The values and determine which random number between 0 and RN is selected. The value of is 1 CN. So, the winning riders are the ones who regularly update their position.

$$L_{it+1}^{ByR} = \delta[L_{it}(\eta, d) * \beta(d) + L_{it}(\xi, d) * [1 - \beta(d)]] \dots (22)$$

(a) Follower update: The trailing rider may arrive at the target in record time by tracking the leader's movements and constantly updating its position. The update equation for the location is given by Eq. (23), and it selects values already present in CN.

$$L_{it+1}^{Fol}(c, crs) = L^{Ler}(Ler, crs) + [cso(ST_{c,crs}^{it}) * L^{Ler}(Ler, crs) * dst_c^{it}] \dots (23)$$

In Eq. (23), the site of the rider is given by  $L^{Ler}$ ; the selector is designated by  $crs$ . The steering angle of the  $c^{th}$  rider in  $cs^{th}$  co-ordinate is signified by  $ST_{c,crs}^{it}$ , and the distance traveled by the  $c^{th}$  rider is given by  $dst_c^{it}$ . This is calculated by multiplying the rider's speed by the off-time rate, with the matching equation being provided in Eq. (24). The maximum amount of time is shown below.

as  $T_{i\text{off}}$  and  $ver_c^{it}$  is the velocity of  $c^{th}$  rider.

$$dst_c^{it} = ver_c^{it} * \left(\frac{1}{T_{i\text{off}}}\right) \dots (24)$$

In addition, Eq. (25) represents the velocity equation, which is a direct relationship between the vehicle speed and the rider characteristics other than the angle.

$$ver_c^{it} = \frac{1}{3}[ge_c^{it} * sp_c^{ge} + sp_{max}^c * acc_c^{it} + (1 - br_c^{it}) * sp_{max}^c] \dots (25)$$

In the overhead equation,  $ge_c^{it}$ ,  $br_c^{it}$ , and  $acc_c^{it}$  are the gear, brake, and throttle of the  $c^{th}$  rider's vehicle, correspondingly. The speed boundary of the gear of  $c^{th}$  rider is given by  $sp_c^{ge}$ .

(b) The overtaker is updated based on the comparative achievement rate, the direction indication, and the co-ordinate selector. For the mathematicians out there, the formula is Eq. (26), in which the site of  $c^{th}$  rider in  $crs^{th}$  coordinate is given by  $L_{it}(c, crs)$ , the direction pointer of  $c^{th}$  rider at time it is signified by  $di_{it}(c)$ . In addition, the rate, which is obtained in Eq. (27) when Eq. (28) is used, serves as a measure of the direction. In this case, we can calculate the relative

success degree of  $c^{th}$  rider at that moment by  $(Res_{it}^{RS}(c))$ .

$$L_{it+1}^{Ovr}(c, crs) = L_{it}(c, crs) + [di_{it}(c) * L^{Ler}(Ler, crs)] \dots (26)$$

$$di_{it}(c) = \left[\frac{2}{1 - \text{lof}(Res_{it}^{RS}(c))}\right] - 1 \dots (27)$$

$$Res_{it}^{RS}(c) = \frac{sur_{it}(c)}{\max_{c=1}^{RS} sur_{it}(c)} \dots (28)$$

(c) Attacker inform: The assailant attempting to take the leader's place will behave just like a follower. The attacker's iterative update procedure is indicated by Eq. (29), where the front-runner's position is supplied by  $L^{Ler}(Ler, d)$  and the angle of  $c^{th}$  rider in  $d^{th}$  organize is denoted as  $ST_{c,d}^{it}$ .

$$L_{it+1}^{Att}(c, d) = L^{Ler}(Ler, d) + [\cos(ST_{c,d}^{it}) * L^{Ler}(Ler, d)] + dst_c^{it} \dots (29)$$

Establishing the Proportion of Success After the updating procedure is complete, each rider's success rate is calculated to have the best possible chance of finishing the race in the position they started in.

### Projected MGS-ROA

The traditional ROA relies on the number of riders who ultimately reach the goal, a novel approach to computing. The ROA strategy entails incremental gains relative to the leaders on each time step (Jadhav et al, 2021). In this example, the attacker takes care of the local neighbourhood to avoid local minima, while the pursuer takes care of the global neighbourhood to achieve rapid convergence. However, this approach needs help when solving issues involving discrete optimization, rather than general optimization. The suggested MGS-ROA refines the revised gear and steering angle pattern for enhanced performance. It is calculated using the extreme fitness values  $F\_MAX$  and the current solution's fitness function  $F\_C$ . The maximum gear (19) is used to revise the gear  $ge\_cit$ , and the steering angle is revised using Eq. (30) if  $F\_C < F\_MAX$ . However, the minimal gear is used to update the gear  $ge\_cit$ , and the angle is revised using Eq. (31).

$$ST_{it+1} = ST_{it} + Distn \dots (30)$$

$$ST_{it+1} = ST_{it} - Distn \dots (31)$$

The term  $Distn$  is a role, which is intended by Eq. (32),

$$Distn = \frac{abs(L_{it+1}^* - L_{it})}{\max(L_{it+1}^*)} \dots (32)$$

In Eq. (32),  $L_{it+1}^*$  is the best site and  $L_{it}$  is the current site.

### Results and discussion

Keras (Chollet, 2018) was used to realize the study's recommended architecture. One NVIDIA GeForce GTX 1070 was used for training, with a batch size of 4. Our

tests are conducted on an RTX2060 graphics processing unit (GPU) with 6 GB of RAM and an AMD CPU R7-4800 with 2.9 GHz and 16 GB of RAM.

The recall is intended by the subsequent Equation (35):

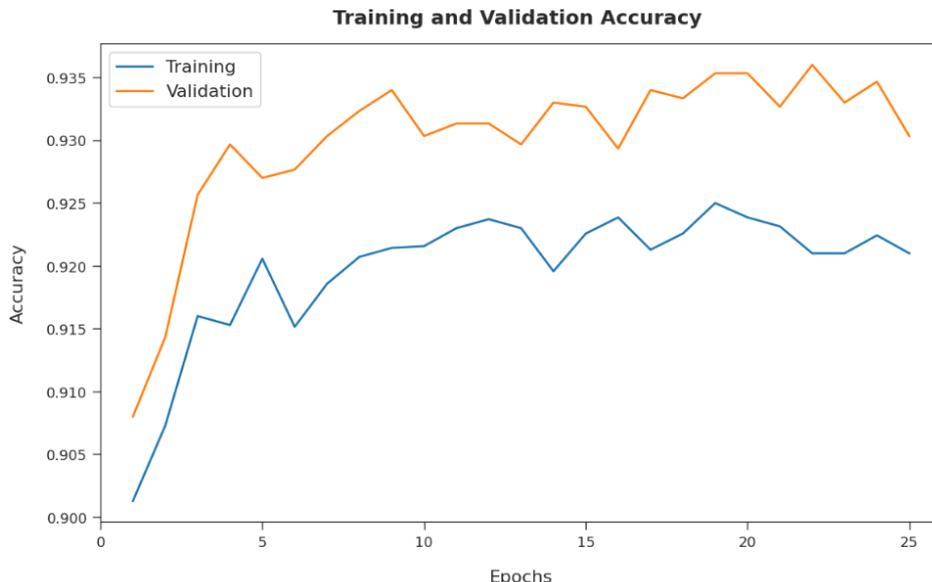


Figure 3. Accuracy graph investigation of Proposed procedure



Figure 4. Loss graph investigation of the proposed procedure

**Performance metrics**

We utilized the measures to gauge how well our model performed. Equation (33) may be used to determine the precision:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(33)$$

The four outcomes are a True Positive (TP), True Negative (TN), a False Positive (FP), and a False Negative (FN).

The precision is intended by the subsequent Equation (34):

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(34)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(35)$$

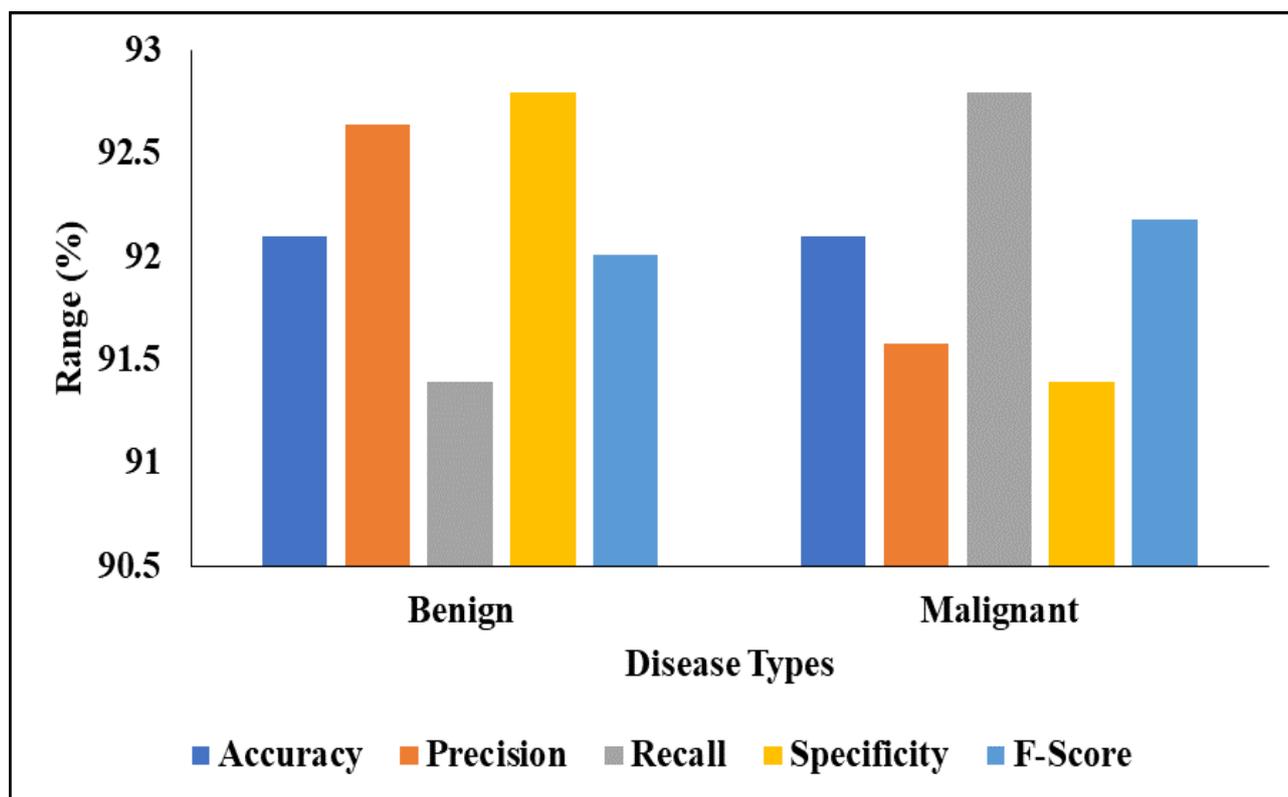
The F1 is designed by the subsequent Equation (36):

$$F1 = 2 \frac{Precision*recall}{Precision+recall} \dots\dots\dots(36)$$

The AUC curves evaluate the false positive and true positive rates at various cut-offs. Precision at each threshold is weighted equally in AP's summary of the curve's recall. Accuracy and loss of the projected model on training and testing data are shown in Figures 3 and 4, respectively.

**Table 2. Analysis of the proposed model for binary classification**

Class Labels	Accuracy	Precision	Recall	Specificity	F-Score
Training Set (70%)					
Benign	92.10	92.64	91.39	92.80	92.01
Malignant	92.10	91.58	92.80	91.39	92.18
Average	92.10	92.11	92.10	92.10	92.10
Testing Set (30%)					
Benign	93.60	93.95	93.33	93.88	93.64
Malignant	93.60	93.25	93.88	93.33	93.56
Average	93.60	93.60	93.60	93.60	93.60

**Figure 5. Graphical analysis of a proposed model for training set**

In the Training Set (70%) of splitting condition, the Benign reached an accuracy of 92.10, a precision value of 92.64, a recall value of 91.39, and a specificity of 92.80 and the F1-score of 92.01, respectively. Malignant reached an accuracy of 92.10, a precision value of 91.58, a recall value of 92.80, a specificity of 91.39 and an F1-score of 92.18, respectively. The average value reached an accuracy of 92.10, a precision value of 92.1, a recall value of 92.10, and a specificity of 92.10 and the F1-score of 92.10, respectively. After the Testing Set (30%)

condition, the Benign reached an accuracy of 93.60, a precision value of 93.95, a recall value of 93.33 and a specificity of 93.88 and the F1-score of 93.64, respectively. Malignant reached an accuracy of 93.60, a precision value of 93.25, a recall value of 93.88, a specificity of 93.33, and an F1-score of 93.56, respectively. The average reached an accuracy of 93.60, precision value of 93.60, recall value of 93.60, and specificity of 93.60, respectively.

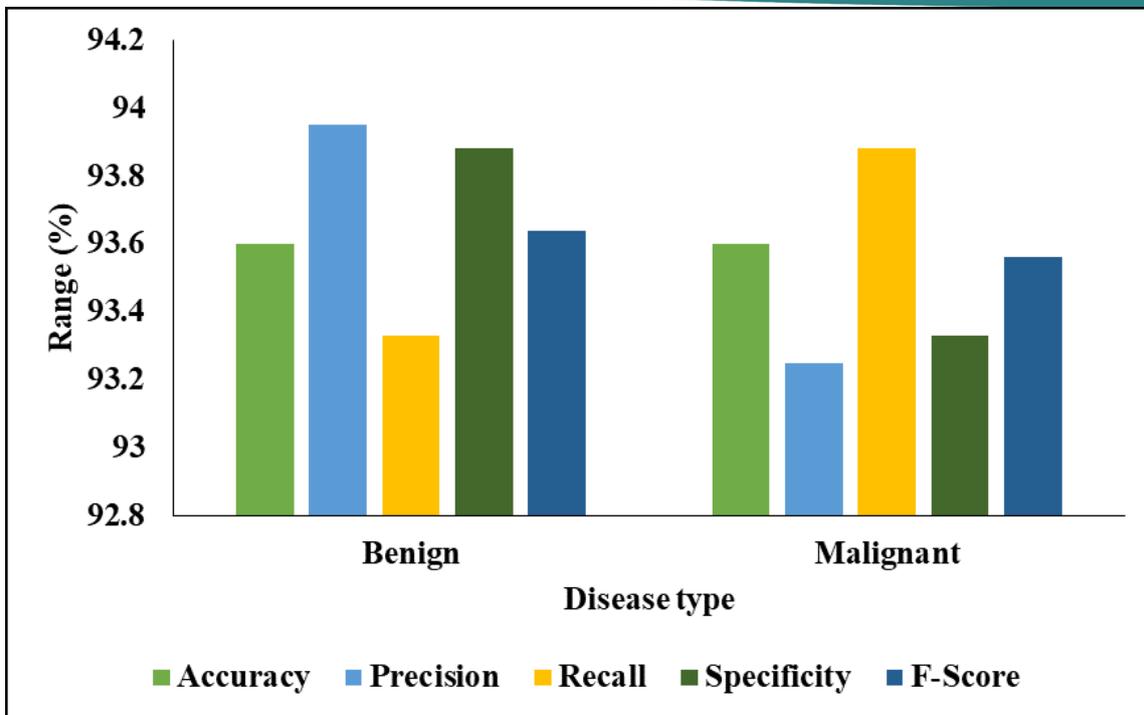


Figure 6. Analysis of the proposed model for testing set

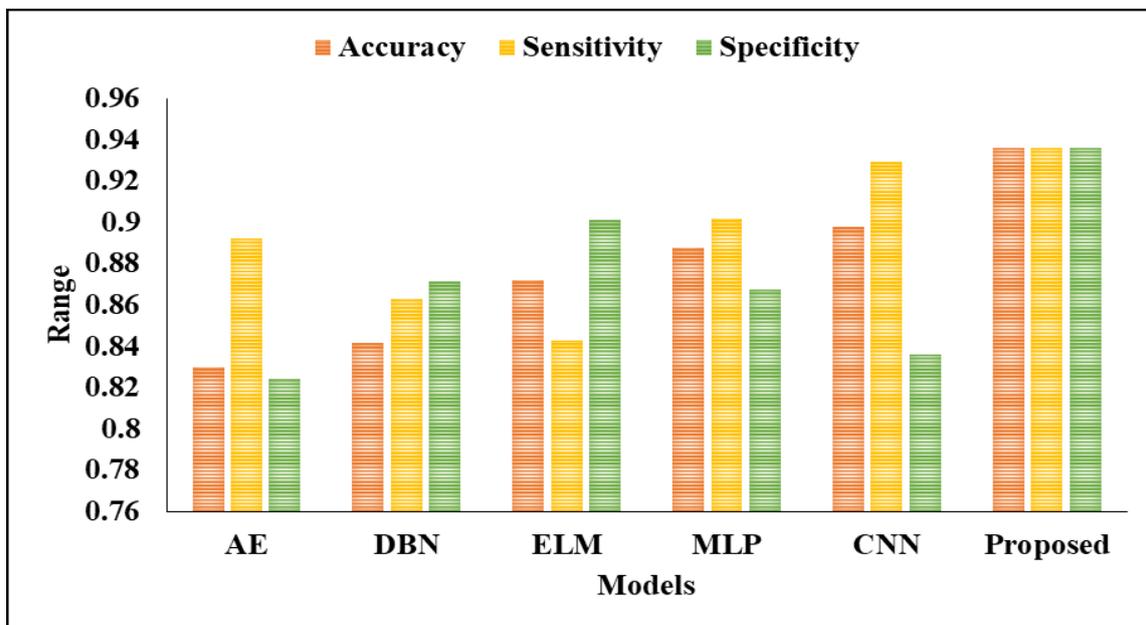


Figure 7. Graphical comparison for various DL models

Table 3. Comparative analysis of various DL models

Methods	Accuracy	Sensitivity	Specificity
AE	0.8301	0.8923	0.8242
DBN	0.8416	0.8631	0.8715
ELM	0.8722	0.8428	0.9016
MLP	0.8880	0.9020	0.8676
CNN	0.8979	0.9293	0.8358
Proposed	0.9360	0.9360	0.9360

The comparative analysis of different DL Models is shown in Table 3. The AE model analysis's accuracy, sensitivity, and specificity were 0.8311, 0.8923, and 0.8242, respectively. After that, the DBN model attained accuracy and sensitivity values of 0.8416 and 0.8631, respectively. The ELM model achieved an accuracy of 0.8722, a sensitivity of 0.8428, and a specificity of 0.9016. The MLP model then achieved an accuracy of 0.8880, sensitivity of 0.9020, and specificity of 0.8676, respectively. The accuracy, sensitivity, and specificity values for the CNN model were then 0.8979, 0.9293, and 0.8358, respectively. Finally, the suggested model

achieved the following results: accuracy of 0.9360, sensitivity of 0.9360, and specificity of 0.9360.

## Conclusion

In order to increase the precision and effectiveness of Monkeypox detection, this study presents a unique strategy that combines several approaches. The process begins by enhancing the original Monkeypox photos by enhancing edge detail. Next, texture features are extracted using an energy layer. Then, the Hyper-parameter-based transferable convolutional neural network (HPT-TCNN) is introduced to improve classification performance even further. Notably, this method streamlines the process by substituting just three convolutional layers and one energy layer for the conventional pooling layer. The model is more approachable and effective because of the usage of the Optimisation Algorithm (MGS-ROA), which makes hyper-parameter adjustment easier.

Additionally, the "Monkeypox Skin Lesion Dataset (MSLD)" was developed by gathering pictures of Monkeypox-related skin lesions on people. This database is a valuable tool for training and validation. The comparison of several deep learning models highlights the suggested model's higher performance. It performs better than models like AE, DBN, ELM, MLP, and basic CNN with accuracy, sensitivity, and specificity, all at 93.60 percent. The model's exceptional accuracy, well-balanced sensitivity, and specificity values indicate how well it can categorize Monkeypox skin lesions. This research reveals a novel method for detecting Monkeypox and offers convincing proof of its higher efficacy compared to other models. For doctors and other healthcare workers, its application offers enormous potential since it can speed up early detection and eventually help limit Monkeypox outbreaks.

## Funding Statement

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

## Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

## Data Availability

Data sharing does not apply to this article as no datasets were generated or analyzed during the current study.

## References

- Adalja, A., & Inglesby, T. (2022). A Novel International Monkeypox Outbreak. *Annals of Internal Medicine*, 175(8), 1175-1176. <https://doi.org/10.7326/M22-1581>.
- Altun, M., Gürüler, H., Özkaraca, O., Khan, F., Khan, J., & Lee, Y. (2023). Monkeypox Detection Using CNN with Transfer Learning. *Sensors (Basel)*, 23(4), 1783. <https://doi.org/10.3390/s23041783>.
- Almufareh MF., Tehsin S., Humayun M., & Kausar S. (2023). A Transfer Learning Approach for Clinical Detection Support of Monkeypox Skin Lesions. *Diagnostics (Basel)*, 13(8), 1503. <https://doi.org/10.3390/diagnostics13081503>.
- Ariansyah, M. H., Winarno, S., & Sani, R. R. (2023). Monkeypox and Measles Detection using CNN with VGG-16 Transfer Learning. *Journal of Computing Research and Innovation*, 8(1), 32-44. <https://doi.org/10.24191/jcrinn.v8i1.340>
- Ayca, A. K., Vedat, T., Ipek, M. (2022). Motor imagery EEG signal classification using image processing technique over GoogLeNet deep learning algorithm for controlling the robot manipulator. *Biomedical Signal Processing and Control*, 72(A), 103295, ISSN: 1746-8094. <https://doi.org/10.1016/j.bspc.2021.103295>.
- Binu, D., & Kariyappa, B. S. (2019). RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits. *IEEE Transactions on Instrumentation and Measurement*, 68(1), 2-26. <https://doi.org/10.1109/TIM.2018.2836058>.
- Català, A., Clavo-Escribano, P., Riera-Monroig, J., Martín-Ezquerria, G., Fernandez-Gonzalez, P., ... & Fuertes, I. (2022). Monkeypox outbreak in Spain: clinical and epidemiological findings in a prospective cross-sectional study of 185 cases. *The British Journal of Dermatology*, 187(5), 765-772. <https://doi.org/10.1111/bjd.21790>.
- Chollet, F. (2018). Keras: The python deep learning library. *Astrophys. Source Code Library*, Tech. Rep. Rec. Ascl:1806.022.
- Jadhav, A. S., Patil, P. B., & Biradar, S. (2021). Optimal feature selection-based diabetic retinopathy detection using improved rider optimization algorithm enabled with deep learning. *Evolutionary intelligence*, 14(4), 1431-1448. <https://doi.org/10.1007/s12065-020-00400-0>
- Khalifa, N. E., Loey, M., & Mirjalili, S. (2022). A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial*

- Intelligence Review*, 55(3), 2351-2377. <https://doi.org/10.1007/s10462-021-10066-4>.
- Li, D. & Ling, D. (2021). Recent advances of deep learning algorithms for aquacultural machine vision systems with emphasis on fish. *Artificial Intelligence Review*, 55(4). <https://doi.org/10.1007/s10462-021-10102-3>.
- Nolen, L. D., Osadebe, L., Katomba, J., Likofata, J., Mukadi, D., ... & Reynolds, M. G. (2016). Extended Human-to-Human Transmission during a Monkeypox Outbreak in the Democratic Republic of the Congo. *Emerging Infectious Diseases*, 22(6), 1014-1021. <https://doi.org/10.3201/eid2206.150579>.
- Out, A., Ebenso, B., Walley, J., Barceló, J. M., & Ochu, C. L. (2022). Global human Monkeypox outbreak: atypical presentation demanding urgent public health action. *The Lancet, Microbe*, 3(8), e554-e555. [https://doi.org/10.1016/S2666-5247\(22\)00153-7](https://doi.org/10.1016/S2666-5247(22)00153-7).
- Pramanik, R., Banerjee, B., Efimenko, G., Kaplun, D., & Sarkar, R. (2023). Monkeypox detection from skin lesion images using an amalgamation of CNN models aided with Beta function-based normalization scheme. *PLoS One*, 18(4), e0281815. <https://doi.org/10.1371/journal.pone.0281815>.
- Samaranayake, L., & Anil, S. (2022). The Monkeypox Outbreak and Implications for Dental Practice. *International Dental Journal*, 72(5), 589-596. <https://doi.org/10.1016/j.identj.2022.07.006>.
- Sanaat, A., Shiri, I., Ferdowsi, S., Arabi, H., & Zaidi H. (2022). Robust-Deep: A Method for Increasing Brain Imaging Datasets to Improve Deep Learning Models' Performance and Robustness. *Journal of Digital Imaging*, 35, 469-481. <https://doi.org/10.1007/s10278-021-00536-0>.
- Savaş, Serkan. (2022). Detecting the Stages of Alzheimer's Disease with Pre-trained Deep Learning Architectures. *Arabian Journal for Science and Engineering*, 47(2), 2201-2218. <https://doi.org/10.1007/s13369-021-06131-3>.
- Suganyadevi, S., Seethalakshmi, V., & Balasamy, K. (2022). A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval*, 11(1), 19-38. <https://doi.org/10.1007/s13735-021-00218-1>.
- Uysal, F. (2023). Detection of Monkeypox Disease from Human Skin Images with a Hybrid Deep Learning Model. *Diagnostics (Basel)*, 13(10), 1772. <https://doi.org/10.3390/diagnostics13101772>.
- Yasmin, F., Hassan, M. M., Hasan, M., Zaman, S., Kaushal, C., El-Shafai, W., & Soliman, N. F. (2023). PoxNet22: A fine-tuned model for the classification of Monkeypox disease using transfer learning. *IEEE Access*, 11, 24053-24076. <https://doi.org/10.1109/ACCESS.2023.3253868>.
- Yue, X., Li, H., Shimizu, M., Kawamura, S., & Meng, L. (2022). YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots. *Machines*, 10(5), 294. <http://dx.doi.org/10.3390/machines10050294>
- Zhang, J., Li, C., Rahaman, M. M., Yao, Y., & Ma, P. (2021). A comprehensive review of image analysis methods for microorganism counting: from classical image processing to deep learning approaches. *Artificial Intelligence Review*, 55(4), 2875-2944. <https://doi.org/10.1007/s10462-021-10082-4>.
- Zumla A., Valdoleiros, S.R., Haider, N., Asogun, D., & Ntoumi, Petersen, E., & Kock, R. (2022). Monkeypox outbreaks outside endemic regions: scientific and social priorities. *The Lancet Infectious Diseases*, 22(7), 929-931. [https://doi.org/10.1016/S1473-3099\(22\)00354-1](https://doi.org/10.1016/S1473-3099(22)00354-1).

### How to cite this Article:

V. Gokula Krishnan, B. S. Liya, S. Venkata Lakshmi, K. Sathyamoorthy and Sangeetha Ganesan (2023). Monkeypox detection using hyper-parameter tuned based transferable CNN model. *International Journal of Experimental Research and Review*, 33, 18-29.

**DOI:** <https://doi.org/10.52756/ijerr.2023.v33spl.003>



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.