Classification of Monkeypox Using Greylag Goose Optimization (GGO) Algorithm

Ahmed Eslam[*1], Mohamed G. Abdelfattah[2], El-Sayed M. El-Kenawy[1,3], Hossam El-Din Moustafa[4]

[1] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
[2] Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[3]MEU Research Unit, Middle East University, Amman 11831, Jordan
[4] Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Egypt
Emails: ahmedeslam@std.mans.edu.eg; eng.mo.gamal@mans.edu.eg; skenawy@ieee.org; hossam_moustafa@mans.edu.eg

Abstract

After the COVID-19 epidemic, public health awareness increased. A skin viral disease known as monkeypox sparked an emergency alert, leading to numerous reports of infections across numerous European countries. Common symptoms of this disease are fever, high temperatures, and water-filled blisters. This paper presents one of the recent algorithms based on a metaheuristic framework. To improve the performance of monkeypox classification, we introduce the GGO algorithm. Firstly, we employ four pre-trained models (AlexNet, GoogleNet, Resnet-50, and VGG-19) to extract the most common features of monkeypox skin image disease (MSID). Then, we reduce the number of extracted features to select the most distinguishing features for the disease. We make it by using GGO in binary form, which has an average fitness of 0.60068 and a best fitness of 0.50248. Lastly, we apply various optimization algorithms, including the (WWPA) waterwheel plant algorithm, the (DTO) Boosted Dipper Throated Optimization, the (PSO) particle swarm optimizer, the (WAO) whale optimization algorithm, the (GWO) gray wolf optimizer, the (FA) firefly algorithm, and the GGO algorithm, all based on the Convolution Neural Network (CNN), to achieve the best performance. Best Performance is indicated in accuracy and sensitivity; it reached 0.9919 and 0.9895 by GGO. A rigorous statistical analysis test was applied to confirm the validity of our findings. We applied Analysis of Variance ANOVA, and Wilcoxon signed tests, and the results indicated that the value of p was less than 0.005, which strongly supports our hypothesis.

Keywords: Monkeypox; water-filled blisters; pre-trained model; classification; GGO algorithm.

1. Introduction

Monkeypox is a viral disease caused by the Orthopox virus. The disease first emerged in western Africa in the late 1950s, and in the 1970s, the Congo Democratic Republic recorded the first human infection transmission, quickly spreading to neighboring countries. For the last four decades, this disease has not caused any headaches, according to the World Health Organization (WHO). Nowadays, many countries all over the world have reported that there is an outbreak of monkeypox and that it has spread rapidly, which indicates a new threat for the world after COVID-19 [1,2].

This disease's difficulty in detection was due to its similarity to other smallpox diseases (chickenpox, measles, and cowpox). They all have many shared symptoms on the first days of infection, like flu, fever, muscle pain,

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

and a high temperature. After a few days, a painful rash appeared all over the body. Any person can be infected by direct transmission to another. Lesions take 2-4 weeks to recover their beauty and shape. Although the disease is spreading rapidly, the percentage of fatalities is not high compared to COVID-19. As a result, early diagnosis of the disease is critical to limiting its spread. It is also critical to continue showing the common symptoms of monkeypox and its transmission mechanisms to reduce its danger to people [3-5].

Different tests, such as computed tomography scan images and polymerase chain reaction tests, are conducted to diagnose the disease. PCR testing is essential in defining the infection early, especially for COVID-19, and identifying monkeypox. The steps of the PCR process include the following: The clinical samples are collected from the infected body, either from the skin (from the lesion area), blood, or respiratory secretions, given the stage of the disease. After going through PCR amplification, the DNA or RNA fragments examined are as follows to check how specific they are to the Mpox virus. A positive result suggests detecting the requisite Mpox virus genetic material in the sample and helps confirm a diagnosis [4-5].

However, based on this literature review, our study will be more specific in using AI, specifically ML and DL, for disease analysis. It has been proved that various DL techniques are highly effective in image analysis and pattern identification, including disease classification. DL is a branch of ML that uses artificial neural networks (ANNs) to identify the features from the images and pass desired predictions. DL algorithms, mainly CNNs, have also been similarly applied in medical image processing, where they participated in skin lesion classification, tumor identification, breast cancer detection, and lung nodule identification. Feature selection methods used in this study assist in excluding some of the generative features, resulting in an effective general model and faster computations [5-8].

Therefore, this paper presents a binary classifier approach to forecasting Monkeypox disease by exploiting a new metaheuristic optimization algorithm. The proposed dataset contains skin images of monkeypox, which is relatively small, and we hence augment the data using several techniques to reduce the chances of overfitting. Feature engineering is performed to extract image features using four DL pre-trained models, Which are the most famous deep learning architectures, namely, AlexNet, GoogleNet, ResNet-50, and VGG-19. Reducing the number of features entails coming up with the most optimal features to be extracted from the extracted features. It is done by applying five different binary optimization algorithms. Next, we classify monkey pox again by fine-tuning CNNs and using the GGO algorithm, which provides an accuracy level of 0. 9919.

We then test it for validity by analyzing the results statistically, as shown below, thus providing emphatic support for our hypothesis. In the ANOVA test and the Wilcoxon rank test, both statistical analyses suggest that the P value is less than 0. 005, consequently evidencing the study's statistical significance and accurate hypothesis.

The main contributions of this paper can be concluded in several steps:

- Pre-processing of MSID dataset to avoid overfitting problem.
- Applying Different pre-trained models (AlexNet, GoogleNet, VGG-19, ResNet-50) to extract most common features.
- Introducing GGO Optimization Algorithm.
- Binary Classification of monkeypox based on CNN-GGO.
- Statistical analysis of classification results to check the validity of the assumed hypothesis.

This paper is organized and structured in several sections: related work presented in section 2, proposed methodology introduced in section 3, results detailed in section 4, and lastly, conclusion and feature work.

2. Related Work

As previously mentioned, after the COVID-19 pandemic, AI, ML, and DL have shown their effectiveness in diagnosing and categorizing the seriousness of medical conditions using high-resolution images from the area of medicine, such as chest X-rays, chest ultrasounds, and computed tomography (CT) scans. Therefore, researchers and scientific communities use AI, ML, and DL methods to diagnose or classify monkeypox disease based on digital images of infected patients' skin.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

Collecting sufficient amounts of clinically validated imaging datasets for this condition is challenging owing to its recent inception. The available literature on monkeypox is contemporary with other relevant research in comparable domains [9,10].

The Chaos Game Optimization (CGO) metaheuristic optimization algorithm is the foundation for a novel method of detecting monkeypox, which was introduced by authors [11]. They utilized two standard datasets, MSLD [12] and MSID [13]. MSLD has a total of 3192 images splitted into two parts: monkeypox and monkeypox; MSID has 790 images splitted into four classes: monkeypox, average, chickenpox, and measles. They implemented data preprocessing to steer clear of common data issues. Five deep learning models (MobileNet V1, MobileNet V2, InceptionV3, DenseNet 121, and DenseNet 169) were used for the feature extraction process, and the extracted features were integrated into residual blocks to reach the learning process. Then, extracted features were fed to different optimization algorithms whale optimization algorithm (WOA), grey wolf optimizer (GWO), sin cosine algorithm (SCA), differential evolution, and traditional genetic algorithm) to reduce the number of features. The detection results for MSLD reached 100% for only 45 samples, and for MSID, they reached 94.16% for a fitting model with 156 samples. (Grad-CAM) Gradient Class Activation Mapping was used to understand the decision-making basics fully. They tried to generalize their model using the Brain Tumor MRI Dataset, and the result reached 97.98%.

Detection of monkeypox using mobile phones was introduced in [14]. The Monkeypox Close Skin Images (MCSI) dataset consists of 400 images separated into four classes: Chickenpox, Monkeypox, Healthy, and Acne [15]. MCSI was used for two studies: the first for binary classification, which reached an accuracy of, and the second study for multi-class detection, which reached an accuracy of .93 by using MobileNetV3Large. Different pre-trained models (VGG16, InceptionResNetV2, NASNetMobile, MobileNetV3Small, and MobileNetV3Large) were applied. Grad-CAM was also applied to comprehend the fundamental principles of decision-making.

In [16], monkeypox classification was introduced using four different studies according to three different datasets: dataset 1 comprises only 60 images, dataset 2 encompasses 1754 images, and dataset 3 contains 2524 images [17,18]. There were three modified versions of different deep learning models (VGG16, ResNet50, and ResNet101). Modified VGG16 achieved the best accuracy of 88%, 76%, and 77% in Studies One, Two, and Four; modified ResNet50 achieved an accuracy of 89% in the third study. Local Interpretable Model-Agnostic Explanations (LIME) was applied to explain predictions visually.

Federated learning (FL) based on deep learning approaches (MobileNetV2, Vision Transformer (ViT), and ResNet50) was introduced in the study [19]. To enhance the classification of monkeypox, combined two public datasets, MSLD and MSID [12, 20] and made a new one consisting of 886 images separated into 4 classes: Chickenpox, Monkeypox, Healthy, and Measles. They applied Cycle Generative Adversarial Networks (CycleGAN) for the data augmentation process to avoid overfitting. ViT-B32 deals with a patch size 32 image and gives the best accuracy of 97.90.

In [21], we introduced a novel approach to detecting monkeypox. This was achieved by combining two datasets [17, 22], resulting in a new dataset of 910 images classified into 5 classes: chickenpox, monkeypox, healthy, smallpox, and zoster zona. Our method involved multiple nested patches: DenseNet for detection; DenseNet 201 for feature extraction; NCA, Chi2, and ReliefF for feature selection; and Iterative Hard Majority Voting (IHMV) for classification. The classification accuracy reached an impressive 91.87, and when we attempted to generalize our model, we applied it to public data, achieving an even higher accuracy of 94.74%.

In [23], we utilized binary classification to detect monkeypox using the Monkeypox Skin Lesion Dataset (MSLD), which is divided into two phases: typical and monkeypox [12]. We achieved impressive results by applying different deep learning models (ResNet50V2, MobileNetV2, and Xception) to the dataset. ResNet50V2, fine-tuned with the RMSprop optimizer, demonstrated the highest accuracy at 0.9874. Xception, tuned with sgd, had an accuracy of 0.9546, and MobileNetV2, tuned with the Adam optimizer, reached an accuracy of 0.9452. Authors of [24] presented an enhancement of feature extraction for monkeypox with only other viral diseases utilized in this study using Residual Network-50-Zero Phase Component Analysis ResNet50 (ZCA), also known as MXGBoost (modified extreme gradient boosting),

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

was used in the classification phase of images. They used a public dataset split into two phases (monkeypox and non-monkeypox) [25]. It reached an accuracy of 97.41, and they used the Wilcoxon rank sum test to check their hypothesis, and the p-value was less than 005. In a study [26], the classification of monkeypox according to feature extraction was introduced using two different datasets: MSID and Darmnet [13, 27]. Darmnet has 8 classes (Monkeypox, Normal, Lupus, Chickenpox, Eczema, Scabies, Measles, and Molluscum contagiosum) with 1285 images. The extended MSID was combined of MSID with Darmnet. Attention-empowered MobileNetV2 was proposed in their study and had the best accuracy compared to six models (ShuffleNetV2, GoogleLeNet, AlexNet, MobileNetV2, ResNet-152, and VGG-19). The proposed model had the best performance accuracy for MSID, reaching 98.19% and 92.28% for EMSID. They also applied MSLD to their proposed model and gave an accuracy of 93.33%.

The classification of monkeypox was introduced based on optimizing CNNs by applying Biruni Earth Radius stochastic fractal search (BERSFS) in [28]. MSID was used in their study. It consisted of 770 images separated into 4 classes with an initial size of 227*227 [29]. To make data preprocessing and feature engineering, The classification results of BERSFS-CNN had an accuracy of .9883 and were the best compared to other classifiers (SVM, K-NN, and DT). They also compared results with four different pre-trained models (AlexNet, GoogleNet, Resnet-50, and VGG-19), which still had the best result. Statistical analysis was applied to check if the result was statistically significant, and statistical results confirmed it. Table 1 comprehensively studies the relevant related work on monkeypox disease.

Table 1: Comprehensive Study of the Recent Work

Ref	data	Analysis tools	Accuracy	Model	classification	Contribution	Data split
[11]	[12] [13]	Grad-CAM	100% 94.16%	CGO-Ensemble	Binary Multi-class (4 classes)	Monkeypox detection using optimization algorithm	80% train 20% test
[14]	[15]	Grad-CAM	93% 88.2%	MobileNetV3Large	Binary Multi-class (4 classes)	Detection of monkeypox using a phone	75% train 25% validation
[16]	[17] [18]	LIME	89%	M-ResNet-50	Binary	Monkeypox diagnosing	80% train 20% test
[19]	[12] [20]	--------	97.90%	ViT-B32	4Multi class (4-classes)	Detection of monkeys based on GAN-Augmentation	80% train 20% test
[21]	[17] [22]	-------	91.87%	MNPDenseNet201	5Multi class (5-classes)	Monkeypox classification and detection	---
[23]	[12]	--------	98.74%	ResNet50V2	Binary	Detection of monkeypox	80% train 10% test 10% validation
[24]	[25]	-------	97.41%	ZCA-ResNet-50,	4Multi class (4-classes)	Enhancing feature selection	-----
[26]	[13] [27]	Grad-CAM LIME	98.19% 92.28%	attention-empowered MobileNetV2	4Multi class (4-classes) 8Multi class (8-classes)	Monkeypox diagnosing	80% train 20% test
[28]	[29]	-------	98.83%	BERSFS-CNN	Binary	Monkeypox classification	-----

3. The proposed methodology

The proposed methodology consists of five main steps: collecting the desired sample, data cleaning and normalization, feature extraction, selecting a set of features that is optimal for the model, classifying the model, and testing for statistical significance. When collecting images and information about monkeypox, there are several sources to use, namely, websites, newscasts, and relevant platforms. However, any chosen dataset, especially the ones in computer vision, usually does not contain raw data; therefore, in the augmentation process, we perform a data preprocessing step to increase and balance the size and quantity of images. When performing a feature transformation in this study, we aim to gain advanced knowledge of the most salient aspects of the disease. In

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

feature extraction, we feed the preprocessed images to the selected models, and we also downsample the number of features used to ease the evaluation of models and reduce the processing time. The GGO-CNN algorithm is employed here in subsequent network architecture optimization. Last, the results obtained by the given approach are assessed based on accuracy, precision, recall, and F1 score. This proposed model is further elucidated in Figure 1, where various steps and procedures are clearly shown.

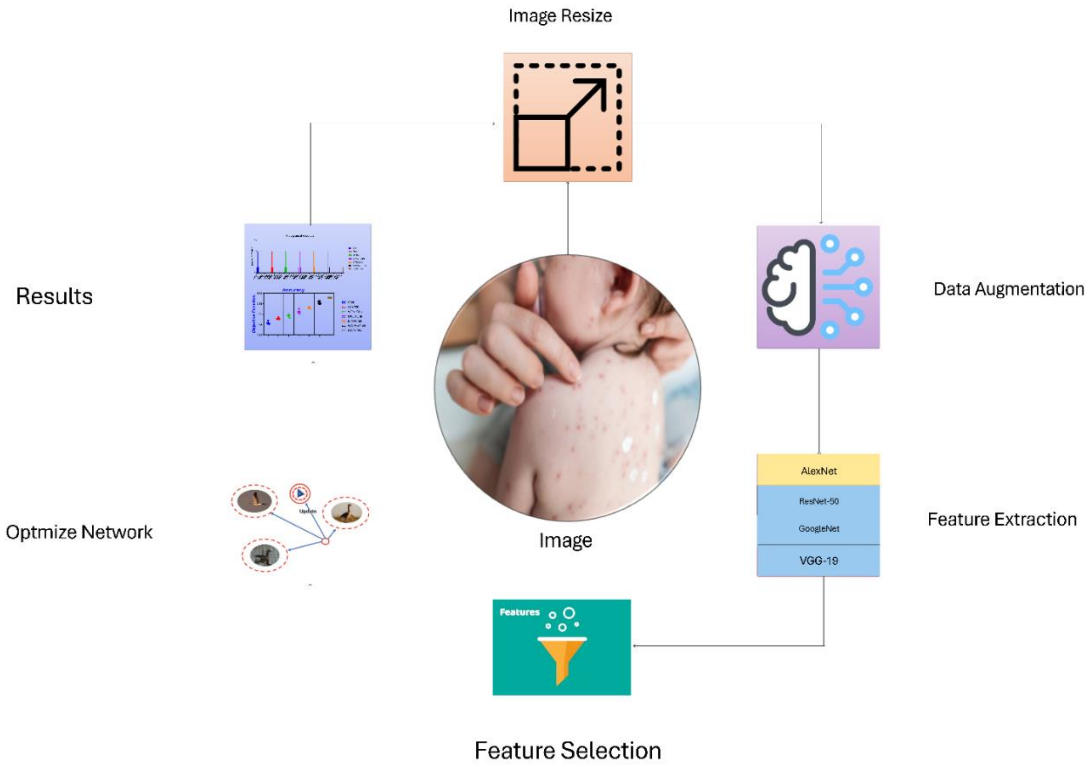

Figure 1: The Main Steps for the Proposed Methodology

3.1 Data collection

Many specialists worldwide view artificial intelligence as crucial to human lives, particularly medicine. Several efficient and uncomplicated techniques exist to get authorized data from websites and platforms across various research fields. Our study uses a highly regarded and trustworthy dataset from the reputable Kaggle platform. The Monkeypox Skin Image Dataset (MSID) comprises 770 RGB photos categorized into four classes: chickenpox, monkeypox, measles, and expected, as shown in Figure 2. The number of images in each class is as follows: chickenpox (107), monkeypox (279), measles (91), and normal (293) [29].

Figure 2: Sample of MSID Dataset

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

3.2 Data pre-processing

The preprocessing phase is a crucial step in image analysis since it enhances the accuracy and unity of the data. The preprocessing procedures used in this article included image scaling, image normalization, and data augmentation.

3.2.1 Image Normalization and Image Resize

In this section, the preprocessing of feature images is described by previous experiments, and their images are scaled and resized to conform to deep learning algorithms. In these experiments, the approach mimics a live experiment by incorporating the fully convolutional layer to perform the resizing task as the experiment suggests despite doing it before feeding the photos into the models. More specifically, to make the processing easier due to the uniformity of all the images, the photos were resized to the cuts of 224 per 224 pixels. This resizing helps standardize the dimension of the images because CNNs applied later in this analysis require images to be of similar sizes.

Further, the preprocessing of the photos also involves the usage of a Normalizing filter that brings more or less equal intensity to the photos and helps in getting rid of shadow caused by extra brightness or contrast, which might lead to noise that affects the actual model. Normalization focuses on the pixel values of the images and entails the scaling of pixel values from a range of 0 – 255 to a range of 0 – 1. This transformation plays a vital role in stabilizing the form of the training process and the convergence of deep learning models where the input data should be kept in a standard form [30].

3.2.2 Data Augmentation

To enhance the performance of our model, we applied various data augmentation techniques such as rotation, blurring, flipping, and zooming. These techniques not only expand the dataset but also provide more certain results. They play a crucial role in reducing the risk of overfitting and increasing the generalizability of the model. By applying these transformations, we created additional variations of the initial images, leading to a more diverse dataset.

Several data augmentation techniques were applied to tackle the problems aligned with overfitting and the need for more data variety. These strategies include zoom range modulation, rotation range modulation, height shift range modulation, shear range modulation, width shift range modulation and vertical flip projection. All of the mentioned enhancements bring some adjustments to the primary photos while providing greater adaptability and generalization to the model in question as it operates with the set of transformed images. They augmented the data source and also helped improve the understanding of the model on the elemental features of images, thus increasing its performance in classification tasks.

3.3 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are a subcategory of artificial neural networks (ANNs) that excel in dealing with matrices, particularly images. They are now powerful tools for diagnosing and classifying medical diseases, underscoring their significance in the field. CNNs mimic how the human brain processes visual information and comprise several levels that perform different tasks. The two layers outlined combine the outcomes of the convolution and pooling layers before making final determinations. As depicted in Figure 3, the architecture of CNNs showcases their efficiency in handling and analyzing the features of image datasets, making them invaluable in applications such as Medical Image Analysis and Disease diagnosis and prognosis [31, 32].

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

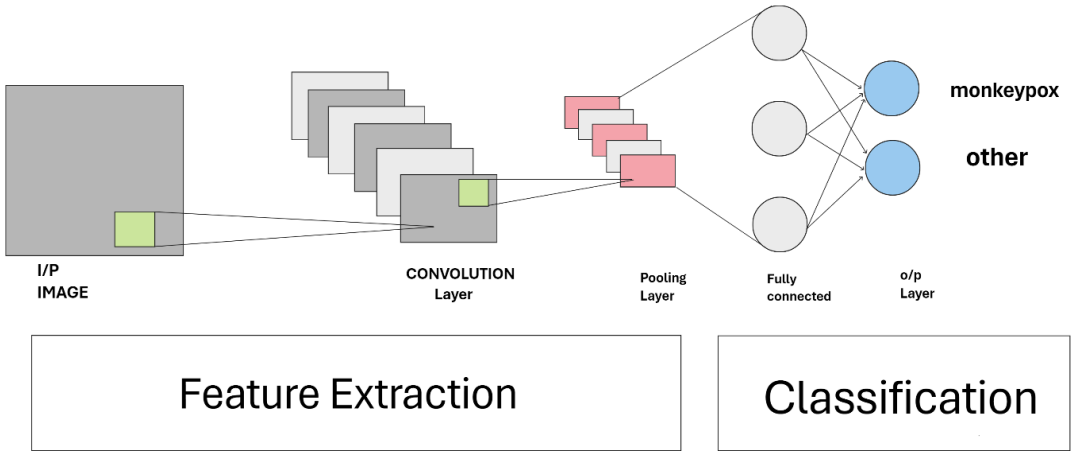

Figure 3: Simple Form of CNN Network Architecture

3.4 Greylag Goose Optimization (GGO) algorithm

Our study employs the Greylag Goose Optimization (GGO) algorithm for optimization. As detailed in Algorithm 1, the GGO algorithm begins by randomly generating a population of individuals, each representing a potential solution to the problem. This population, denoted as Xi (i = 1, 2,..., n), has a size n, referred to as a gaggle. Each individual is evaluated using a chosen objective function, Fn. After calculating the objective function for each individual (agent), Xi, the best solution or leader, is identified and denoted by P. The GGO algorithm then dynamically divides the population into an exploitation group (n2) and an exploration group (n1). In each iteration, the number of solutions in each group is adjusted based on the best solution found. Initially, the GGO algorithm splits the population evenly, with 50% in the exploration group and 50% in the exploitation group. As iterations progress, the number of agents in the exploration group (n1) decreases while the number of agents in the exploitation group (n2) increases. However, the objective function value of the best solution remains unchanged after three consecutive iterations. In that case, the algorithm increases the number of agents in the exploration group (n1) to search for a better solution and avoid local optima [33].

3.4.1 Exploration operation

As we will cover in more detail later, exploration advances toward the optimal solution to prevent the local optimum from stagnating and finds intriguing regions of the search space.

Approaching the optimal resolution: Using this strategy, the goose explorer will look for interesting new areas to investigate close to its present position. We accomplish this by consistently assessing the numerous local options to identify the most optimal representation in terms of fitness.

$\mathbf{A}$ and $\mathbf{C}$ vectors updated as $\mathbf{A} = 2\mathbf{a} \cdot r_1 - \mathbf{a}$ and $\mathbf{C} = 2.r_2$Throughout iterations with the parameter altered linearly from 2 to 0, the GGO algorithm employs the following equations to accomplish this:

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A} \cdot |\mathbf{C}.\mathbf{X}^*(t) - \mathbf{X}(t)| \quad (1)$$

X(t) indicates the iteration of an agent. The optimal solution (the boss) position is denoted by X(t+1) and represents the updated position of the agent. Random variations occur for r1 and r2, with values ranging from zero to one. To prevent agents from being influenced by a single leader's position and promote broader exploration, we will use the following equation, which involves selecting three randomly designated search agents (paddlings) and then adjusting the current search agent's position as follows:

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

$$\mathbf{X}(t+1) = w_1 * \mathbf{X}_{\text{Paddle 1}} + \mathbf{z} * w_2 * (\mathbf{X}_{\text{Paddle 2}} - \mathbf{X}_{\text{Paddle 3}}) + (1-\mathbf{z}) * w_3 * (\mathbf{X} - \mathbf{X}_{\text{Paddle 1}}) \quad (2)$$

where w_1, w_2, and w_3 values have been updated between [0,2]. The given formula may be used to compute the parameter z, which is decreasing exponentially.

$$\mathbf{z} = 1 - \left(\frac{t}{t_{\max}}\right)^2 \quad (3)$$

where t denotes the iteration number and $t_{\max}$the maximum number of iterations.

For $r_3 \geq 0.5$. The second updating procedure looks like this, where the values of the a and A vectors are reduced.

$$\mathbf{X}(t+1) = w_4 * |\mathbf{X}^*(t) - \mathbf{X}(t)| \cdot e^{bl} \cdot \cos(2\pi l) + [2w_1(r_4 + r_5)] * \mathbf{X}^*(t), \quad (5)$$

where l is an arbitrary number in [-1,1], and b parameter is a constant. While $r_4$and r_5 are updating in [0,1], the $w_4$parameter is updating in [0,2].

3.4.2 Exploitation operation

The exploitation team is responsible for enhancing the current solutions. At the end of each cycle, the GGO determines who is the most fit and rewards them appropriately. As explained below, the GGO uses two approaches to accomplish its exploitation goal.

Approaching the optimal resolution: To get closer to the ideal answer, utilize the following equation. The three sentries ($\mathbf{X}_{\text{Sentry1}}$, $\mathbf{X}_{\text{Sentry2}}$,and $\mathbf{X}_{\text{Sentry3}}$) direct other people ($\mathbf{X}_{\text{NonSentry}}$) To reposition themselves in the direction of the prey's anticipated position. The procedure for updating positions is depicted in the following equations:

$$\mathbf{X}_1 = \mathbf{X}_{\text{Sentry 1}} - \mathbf{A}_1 \cdot \left|\mathbf{C}_1 \cdot \mathbf{X}_{\text{Sentry 1}} - \mathbf{X}\right|,$$

$$\mathbf{X}_2 = \mathbf{X}_{\text{Sentry 2}} - \mathbf{A}_2 \cdot \left|\mathbf{C}_1 \cdot \mathbf{X}_{\text{Sentry 2}} - \mathbf{X}\right|, \quad (5)$$

$$\mathbf{X}_3 = \mathbf{X}_{\text{Sentry3}} - \mathbf{A}_3 \cdot \left|\mathbf{C}_1 \cdot \mathbf{X}_{\text{Sentry 3}} - \mathbf{X}\right|,$$

where the calculations for$\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ are calculated as $\mathbf{A} = 2\mathbf{a}.r_1 - \mathbf{a}$ and $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ are calculated as $\mathbf{C} = 2r_2$. An average of the three solutions of $\mathbf{X}_1, \mathbf{X}_2$ and $\mathbf{X}_3$maybe used to represent the updated locations for the population $\mathbf{X}(t+1)$

$$\mathbf{X}(t+1) = \overline{\mathbf{X}}_i|_0^3 \quad (6)$$

Looking for the region around the perfect response When flying, the most hopeful option is situated near the boss, who provides the most effective reaction. That leads some people to look for improvements by examining areas called. $\mathbf{X}_{\text{Flock 1}}$At is near the optimal answer, the GGO uses the following equation to carry out the previously described procedure:

$$\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1+\mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{\text{Flock 1}}) \quad (8)$$

3.4.3 Selection of the best solution

The Greylag Goose Optimization (GGO) algorithm, which refers to a special kind of optimization, has proven to possess outstanding exploration abilities by examining the members of the exploration group and, consequently, employing the mutation approach. This ability aids in increasing distance from convergence: Corporate exploration capacity This robust exploration capacity is a strong indication of how firms can delay convergence. Algorithm 1 below gives the pseudo-code for the GGO in its simplest form following the detailed breakdown indicated above. First, we reveal the group size, mutation rate, and number of iterations to the GGO. The GGO then categorizes its members into two groups: The first is the operational level, which involves exploitative tasks, and the second one is exploratory. To find the best solution, the GGO then evaluates the size of each group used during a fixed period. Each group uses two procedures regarding their tasks, although they share goals.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

The exploitation group (n_2) and the exploration group (n_1) are updated by applying the GGO algorithm's stages. During iterations, the parameter r_1 is modified as follows:

$r_1 = c\left(1 - \frac{t}{t_{\max}}\right)$Re, t is the present iteration, c is a constant, and t_{max} is the total number of iterations. GGO updates the agents in the search space after each iteration, and the agents' positions in the exploration and exploitation groups are switched around randomly. GGO provides the optimal solution in the last stage.

Algorithm 1: GGO Algorithm

Initialize GGO population $\mathbf{X}_i(i = 1,2,\ldots,n)$, size n, iterations t_{max}, objective function F_n.
Initialize GGO parameters **a**, **A**, **C**, b, l, c, r_1, r_2, r_3, r_4, r_5, w, w_1, w_2, w_3, w_4, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $t = 1$
Calculate objective function F_n for each agents $\mathbf{X}_i$
Set P = best agent position
Update Solutions in exploration group (n_1) and exploitation group (n_2)
while $t \le t_{max}$ **do**
  for ($i = 1 : i < n_1 + 1$) **do**
    if (t%2 == 0) **then**
      if ($r_3 < 0.5$) **then**
        if ($|A| < 1$) **then**
          Update position of current search agent as $\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A}.|\mathbf{C}.\mathbf{X}^*(t) - \mathbf{X}(t)|$
        else
          Select three random search agents $\mathbf{X}_{Paddle1}$, $\mathbf{X}_{Paddle2}$, and $\mathbf{X}Paddle3$
            Update (**z**) by the exponential form of $\mathbf{z} = \mathbf{1} - \left(\frac{t}{t_{max}}\right)^2$
            Update position of current search agent as
$\mathbf{X}(t+1) = w1 * \mathbf{X}Paddle1 + \mathbf{z} * w2 * (\mathbf{X}Paddle2 - \mathbf{X}Paddle3) + (1 - \mathbf{z}) * w3 * (\mathbf{X} - \mathbf{X}Paddle1)$
      end if
    else
      Update position of current search agent as
$\mathbf{X}(t+1) = w_4 * |\mathbf{X}^*(t) - \mathbf{X}(t)|.e^{bl}.cos(2\pi l) + [2w_1(r_4 + r_5)] * \mathbf{X}^*(t)$
    end if
  else
    Update individual positions as
$\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1 + \mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{Flock1})$
  end if
end for
for ($i = 1 : i < n_2 + 1$) **do**
  if (t%2 == 0) **then**
    Calculate $\mathbf{X}1 = \mathbf{X}Sentry1 - \mathbf{A}1.|\mathbf{C}1.\mathbf{X}Sentry1 - \mathbf{X}|$, $\mathbf{X}2 = \mathbf{X}Sentry2 - \mathbf{A}2.|\mathbf{C}2.\mathbf{X}Sentry2 - \mathbf{X}|$,
    $\mathbf{X}3 = \mathbf{X}Sentry3 - \mathbf{A}3.|\mathbf{C}3.\mathbf{X}Sentry3 - \mathbf{X}|$
    Update individual positions as $\overline{\mathbf{X_i}}\Big|_0^3$
  else
    Update position of current search agent as
$\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1 + \mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{Flock1})$
  end if
end for
Calculate objective function F_n for each $\mathbf{X}_i$
Update parameters
Set $t = t + 1$
Adjust beyond the search space solutions
if (Best F_n is same as previous two iterations) **then**

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

37: **Increase** solutions of exploration group (n_1)
38: **Decrease** solutions of exploitation group (n_2)
39:**end if**
40: **end while**
41: **Return** best agent **P**

Complexity analysis

This section presents a complex study of the GGO algorithm using Algorithm 1. By utilizing the population number denoted as n and the number of iterations as t_{max}, the level of difficulty of each algorithmic step can be determined as

- Initializing of GGO population: O (1).
- Initializing of GGO parameters $\mathbf{a}, \mathbf{A}, \mathbf{C}, b, l, c, r_1, r_2, r_3, r_4, r_5, \mathbf{w}, w_1, w_2, w_3, w_4, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, t = 1: O$ (1).
- Evaluating objective function F_n for each agents $\mathbf{X}_i$: $O(n)$.
- Getting best individual P: $O(n)$.
- Update solutions in exploration group (n_1) and exploitation group (n_2): $O(n)$.
- Update position of search agent in exploration group: $O(t_{\max} \times n)$.
- Select three random search agents $\mathbf{X}_{\text{Paddle 1}}, \mathbf{X}_{\text{Paddle2}}$, and $\mathbf{X}_{\text{Paddle 3}}$: $O(t_{\max} \times n)$.
- Update **z**By an exponential form: $O(t_{\max} \times n)$.
- Update position of search agent based on **z**Nd random agents: $O(t_{\max} \times n)$.
- Update position of search agent in exploration group if $r_3 \geq 0.5 : O(t_{\max} \times n)$.
- Update position of search agent in exploration group if $t\%2 \neq 0 : O(t_{\max} \times n)$.
- Calculate $\mathbf{D}_{\text{Sentry 1}}, \mathbf{D}_{\text{Sentry 2}}$, and $\mathbf{D}_{\text{Sentry 3}}$: $O(t_{\max} \times n)$.
- Calculate $\mathbf{X}_1, \mathbf{X}_2$, and $\mathbf{X}_3$: $O(t_{\max} \times n)$.
- Update position of search agent in the exploitation group: $O(t_{\max} \times n)$.
- Update position of search agent in the exploitation group if $t\%2 \neq 0 : O(t_{\max} \times n)$.
- Evaluating agents' objective function: $O(t_{\max} \times n)$.
- Updating parameters: $O(t_{\max})$.
- Increasing iteration counter: $O(t_{\max})$.
- Adjust beyond the search space solutions: $O(t_{\max})$.
- Test is best F_nIs same as previous two iterations: $O(t_{\max})$
- The GGO algorithm has a $O(t_{\max} \times n)$El of complexity. The algorithm complexity for issues with $d$variable can be calculated as $O(t_{\max} \times n \times d)$.

Table 2 summarizes the configuration parameters used in the algorithms being compared in this paper. Each algorithm runs for 100 iterations with several agents equal to 10. The following algorithms make up the class of algorithms under consideration: the Grey Wolf Optimizer (GWO), the Whale Optimization Algorithm (WOA), and the Genetic Algorithm (GA).

Table 2: Configuration of Compared Algorithms with 100 Iterations and 10 Agents for Each One.

Algorithm	Parameter (s)	Value (s)
GWO	a	2 to 0
	Acceleration constants C_1, C_2	[2,2]
WOA	a	2 to 0

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

	r	[0,1]
GA	Mutation ratio	0.1
	Crossover	0.9
	Selection mechanism	Roulette wheel

4. Results

4.1 Confusion matrix

For further analysis of complete results, it is essential to use the confusion matrix shown in Figure 4. This matrix gives an analysis of several key factors that are useful for evaluation and are outlined as follows: These are accorded as the true positive, False Positive, True Negative, and False Negative and these components are critical in defining the accuracy, precision, recall or efficacy of the model. By breaking the model into such components, we can understand how effectively it is performing, where it might be having problems and thus determine whether our strategy is sound and reliable. The confusion matrix acts as a tool by which we can clearly and straightforwardly analyze the successes and shortcomings of the model in question regarding classification.

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Figure 4: Confusion Matrix

- True Positive (TP): a person has a positive test and has the disease.
- False Positive (FP): a person has a positive test but does not have the disease.
- True Negative (TN): a person has a negative test and does not have the disease.
- False Negative (FN): a person has a negative test and has the disease.

4.2 Model Evaluation

The approaches can be analyzed based on the following parameters: Accuracy, Sensitivity, Specificity, Precision, Negative Predictive Value and F-score. Accuracy gives the total percentage of actual data correctly classified by the given model. At the same time, Sensitivity evaluates the ability of the model to classify positive cases correctly; Specificity evaluates the model's capacity to classify negative cases correctly. Precision focuses on the quality of optimistic predictions, and NPV determines a percentage of indeed

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

negatives compared to all predicted negatives. Table 3 provides these indicators, which compare the evaluated approaches in detail to perform the assessment effectively.

Table 3: Classification Model Evaluation Criteria.

Evaluation Criteria	Value
Accuracy	$\left(\frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}\right)$
Sensitivity (Recall)	$\left(\frac{\text{True Positives}}{\text{True Positives + False Negatives}}\right)$
Specificity	$\left(\frac{\text{True Negatives}}{\text{True Negatives + False Positives}}\right)$
Positive Predictive Value	$\left(\frac{\text{True Positives}}{\text{True Positives + False Positives}}\right)$
Negative Predictive Value	$\left(\frac{\text{True Negatives}}{\text{True Negatives + False Negatives}}\right)$
F-Score	$\left(2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision + Sensitivity}}\right)$

4.3 Feature Extraction Results

The feature extraction process involved in this study was done using the AlexNet deep learning model, which yielded an accuracy of 93.22%. It also shows that the particular model accurately determines latent patterns within any given data set. AlexNet is famous for its deep structure consisting of several convolutional layers, which are particularly effective when dealing with complicated pictures and achievable at diagnosing them and identifying various features. They are essential factors for decision-making and classification of visualization in images. The architecture of AlexNet has been specially developed with large and complex sets of data, thus making it suitable for tasks involving larger dimensionality. The proposed model successfully trains and develops a reputation for achieving greater accuracy; 93.22% indicates the model's stability, and it is also scalable, as it can achieve reasonable rates on unseen data. This level of performance points that AlexNet can perfectly identify and incorporate features that are most helpful within the set dataset, which is a critical capability for solving the image classification problem.

The feature extraction of AlexNet in this method indicates that future optimization and expansion in different fields have the potential to be much more successful. The model's accuracy indicates that deep learning models are well suited to solving complex datasets and should be valuable in diverse fields such as medical image analysis and developing autonomous vehicles. In addition, the robustness of AlexNet in this respect means that it is a candidate model that can be widely utilized in diverse tasks, which need significant accuracy and the extraction of highly complex features. As a performance metric, this points to the evolution of deep learning technology and its suitability for solving various complex problems. Therefore, future perspectives on machine learning and artificial intelligence development entail that AlexNet and other models are significant tools for achieving enhanced results in various fields.

4.4 Feature Selection Results

In this section, we offer an account of the feature selection as conducted in the preceding sections of the paper. It was done so by employing the binary Greylag Goose Optimization (bGGO) feature selection algorithm. It is efficient in terms of reducing the computation required when identifying the best features from the dataset. The bGGO algorithm emulates the greylag geese's migration behavior while implementing the flock's selection process by comradeship and rivalry joinery. In this study, after identifying the most significant features, the algorithm was rerun several times to confirm the features' reliability. The results of the bGGO were assessed in terms of average and the best fitness values achieved during the optimization mode. The average fitness summarizing the bGGO moves was 0.60068. Applying the selected features is generally a good measure to determine the variability in the set dataset. The best fitness value as the combination of the optimization level in any single run has been computed as 0.50248.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

This value characterizes the maximum value inherent in the algorithm in terms of recognition of the most crucial features. These chosen features were then used to group the entered images into particular classes. The classification was performed using deep learning classifiers, which were effective even for intricate patterns and higher dimensions. Using the combination of the selected features with the deep learning classifiers, the algorithm enabled the increase of the rate of classified images accurately and effectively. In conclusion, our study demonstrates the effectiveness of the Binary Greylag Goose Optimization (bGGO) algorithm in feature selection. When applied to the analysis and classification of images, this algorithm significantly improved the performance of deep learning classifiers. Our findings underscore the potential of the bGGO algorithm in addressing complex machine-learning problems, further enhancing its capabilities.

4.5 Classification Results

The classifiers used in this study were convolutional neural networks (CNNs). Table 4 illustrates that the classification results of both the proposed and comparative approaches are influenced by the optimization of the CNN model's parameters. The GGO-CNN model achieved an accuracy of 0.9919, demonstrating its superiority over other state-of-the-art classifier models constructed using the CNN technique. The WWPA-CNN-based model produced the second-best classification results with an accuracy of 0.9802, followed by the DTO-CNN-based model with a score of 0.9708, the GWO-CNN-based model with a score of 0.9639, the WOA-CNN-based model with a score of 0.9578, the GA-CNN-based model with a score of 0.951, and finally, the standard CNN with an accuracy of 0.9425, which yielded the least accurate results.

Table 4: Classification Result of Different Optimization Algorithms Based On CNN

Models	Accuracy	Sensitivity (TRP)	Specificity (TNP)	P-value (PPV)	Nvalue (NPV)	For
CNN	0.942529	0.94	0.944681	0.935323	0.948718	0.937656
GA+CNN	0.951064	0.949367	0.95279	0.95339	0.948718	0.951374
WOA+CNN	0.957895	0.961538	0.954357	0.95339	0.962343	0.957447
GWO+CNN	0.963983	0.961538	0.966387	0.965665	0.962343	0.963597
DTO+CNN	0.970874	0.972222	0.969582	0.968379	0.973282	0.970297
WWPA+CNN	0.980216	0.98	0.980392	0.976096	0.983607	0.978044
GGO+CNN	0.99197	0.989523	0.994488	0.994617	0.989273	0.992063

Table 5 summarizes the comparisons between the proposed solution in this paper and related works highlighted by reference [28]. The following comparison highlights how this proposed approach can perform better accuracy than the previous one. Notably, the sensitivity we obtained as part of the methodology employed in this study was recorded at an impressive 0.99197 and an asterisk symbol representing a p-value of statistical significance. There is a substantial increase in the proposed solution's performance, proving that the concept is advantageous rather than inferior to the previously proposed methods. By systematically analyzing the measures identified in this research and comparing them with different results, we provide valuable information supporting the solidity and credibility of the proposed strategy and opening up a new path to future developments.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

Table 5: Comparison Between Proposed Methodology and Other Similar Work

4.6 Statistical analysis

We conducted an additional experiment to understand the proposed methodology's statistical characteristics better. This study concentrates on the statistical difference between the proposed methodology and other methods for optimizing convolutional neural networks. We compare the GGO results with those of GA, WOA, GWO, DTO, and WWPA to showcase the superiority of the proposed CNNS optimization. The statistical analysis was performed using GGO compared to other optimizing methods in terms of the statistical analysis in Table 6.

Table 6: Statical Analysis Result Obtained By GGO-CNN

	CNN	GA+CNN	WOA+CNN	GWO+CNN	DTO+CNN	WWPA+CNN	GGO+CNN
Number of values	6	6	6	6	6	6	6
Minimum	0.9405	0.9501	0.9539	0.9604	0.9709	0.9802	0.991
25% Percentile	0.942	0.9508	0.9569	0.9631	0.9709	0.9802	0.9917
Median	0.9425	0.9511	0.9579	0.964	0.9709	0.9802	0.992
75% Percentile	0.9438	0.9518	0.9584	0.9656	0.9736	0.9845	0.992
Maximum	0.9475	0.9541	0.9599	0.9694	0.9751	0.9852	0.992
Range	0.007	0.004	0.006	0.009	0.004213	0.005	0.001
Mean	0.943	0.9514	0.9576	0.9644	0.9719	0.9817	0.9918
Std. Deviation	0.002345	0.001366	0.001966	0.002881	0.001776	0.002345	0.000403
Std. Error of Mean	0.000957	0.000558	0.000803	0.001176	0.000725	0.000957	0.000165
Geometric mean	0.943	0.9514	0.9576	0.9644	0.9719	0.9817	0.9918
Geometric SD factor	1.002	1.001	1.002	1.003	1.002	1.002	1
Sum	5.658	5.708	5.745	5.786	5.832	5.89	5.951

Hence, significant tests of the proposed methodology were applied to check the assumed hypothesis. We applied the ANOVA test, a significant test, to determine whether there was a statistically significant difference

Ref	Accuracy	Sensitivity	p Value
[28]	.9883	0.8571	0.7595
Our work	.99197	0.989523	0.994617

between the proposed GGO algorithm and the existing algorithms. The measured values of the ANOVA test are shown in Table 7, where p-value ($p > .005$) means statistically significant. We also performed a Wilcoxon signed rank test for the proposed methodology, recording the results in Table 8 with a value of p less than .005.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

Table 7: ANOVA Test for Performance of the GGO-CNN.

ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.01057	6	0.001762	F (6, 35) = 435.1	P<0.0001
Residual (within columns)	0.000142	35	4.05E-06		
Total	0.01072	41			

Table 8: The Wilcoxon Signed Test for Performance of the GGO-CNN, Compared To Other Competing Optimization Methods

	CNN	GA+CNN	WOA+CNN	GWO+CNN	DTO+CNN	WWPA+CNN	GGO+CNN
Theoretical median	0	0	0	0	0	0	0
Actual median	0.9425	0.9511	0.9579	0.964	0.9709	0.9802	0.992
Number of values	6	6	6	6	6	6	6
Wilcoxon Signed Rank Test							
The sum of signed ranks (W)	21	21	21	21	21	21	21
The sum of positive ranks	21	21	21	21	21	21	21
The sum of hostile ranks	0	0	0	0	0	0	0
P value (two-tailed)	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313	0.0313
Is it exact or an estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	*	*	*	*	*	*	*
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How significant is the discrepancy?							
Discrepancy	0.9425	0.9511	0.9579	0.964	0.9709	0.9802	0.992

Figure 5 shows the accuracy plot and histogram plot of the results achieved using GGO+CNN. The accuracy plot indicates that the proposed algorithm got the proposed algorithm got the best accuracy in comparison to the other six optimization algorithms. Figure 6 indicates the accuracy of the histogram plot in the range of 0.940 to 0.994. It also shows that the proposed optimization algorithm works consistently and correctly in sorting the images into categories of monkeypox cases.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

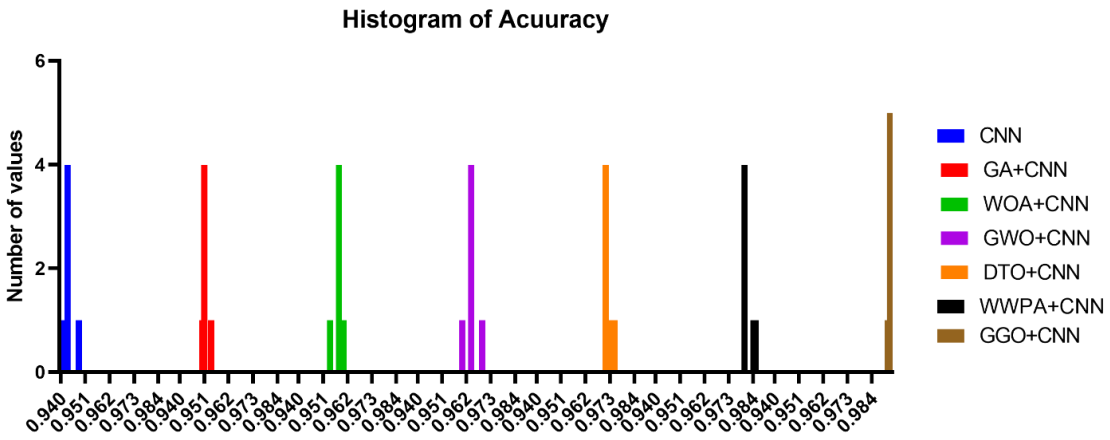

Figure 5: Accuracy Histogram of Introduced Optimization Algorithm Compared with Other Algorithms Based On CNN

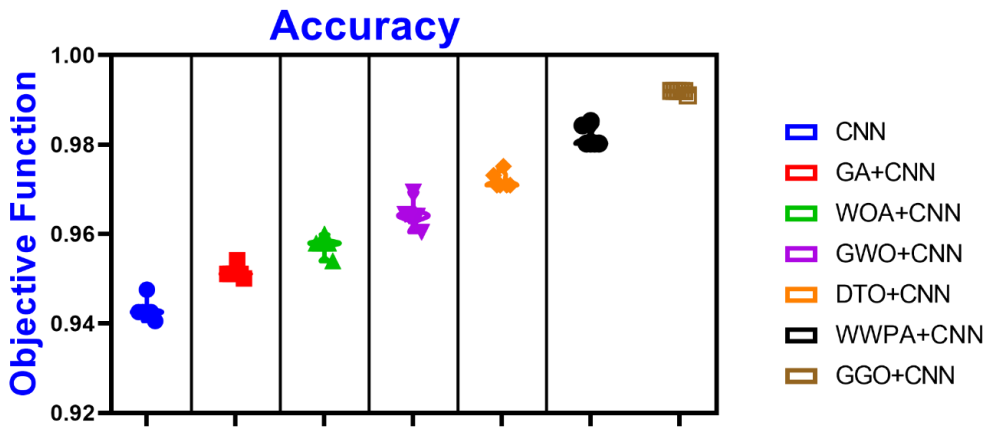

Figure 6: Accuracy Plot of Optimization Algorithm Compared with Other Algorithms Based On CNN

Figure 7 provides a graphic representation of the statistical analysis of the classification results, including four plots. The residual, homoscedasticity, QQ, and heatmap plots exist to visualize the results better. These graphs present a set of results that indicate steady performance of the GGO with residual errors ranging between -0.04 to +0.04 with homoscedasticity values ranging from 0 and + 0. The null hypothesis was rejected at $p<0.04$.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

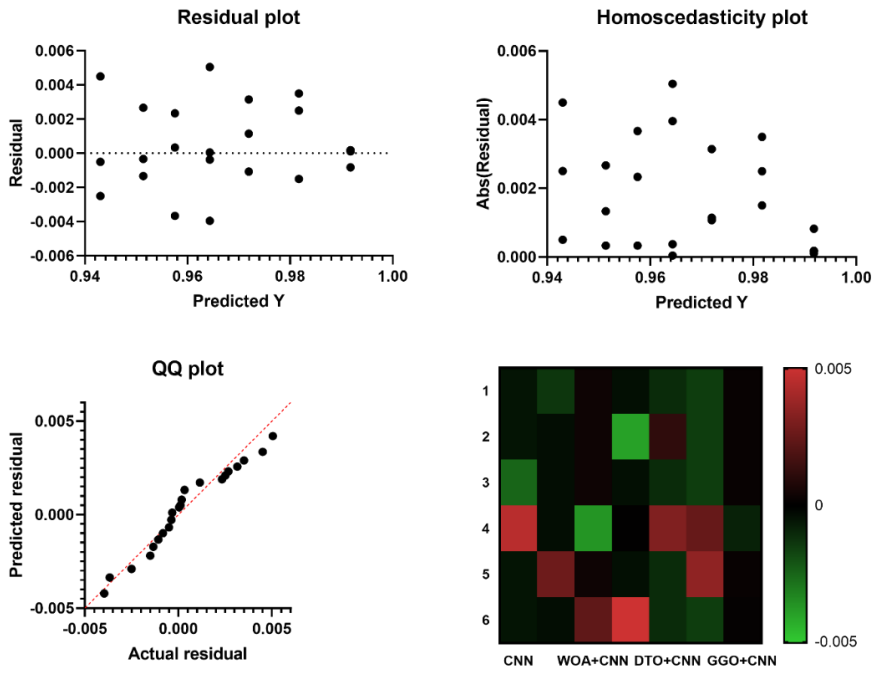

Figure 7: The Residual Plot, Homoscedasticity Plot, QQ Plot, And Heat Maps QQ Plots of the GGO-CNN And Other Optimization-Based Models.

Based on the classification result in Table 4, we represent the result in different plots to fully understand the results, not just some numbers. Figure 8 and Figure 9 indicate the regression plots, which, as the name indicates, create a regression line between two parameters and aid in visualizing their linear correlations.

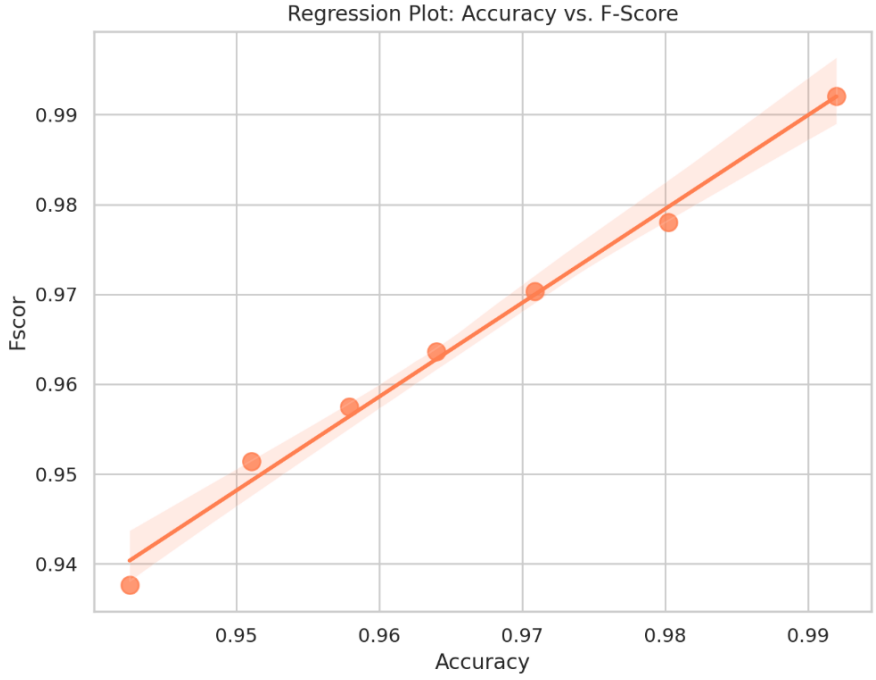

Figure 8: The Regression Plot of Accuracy And F-score

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

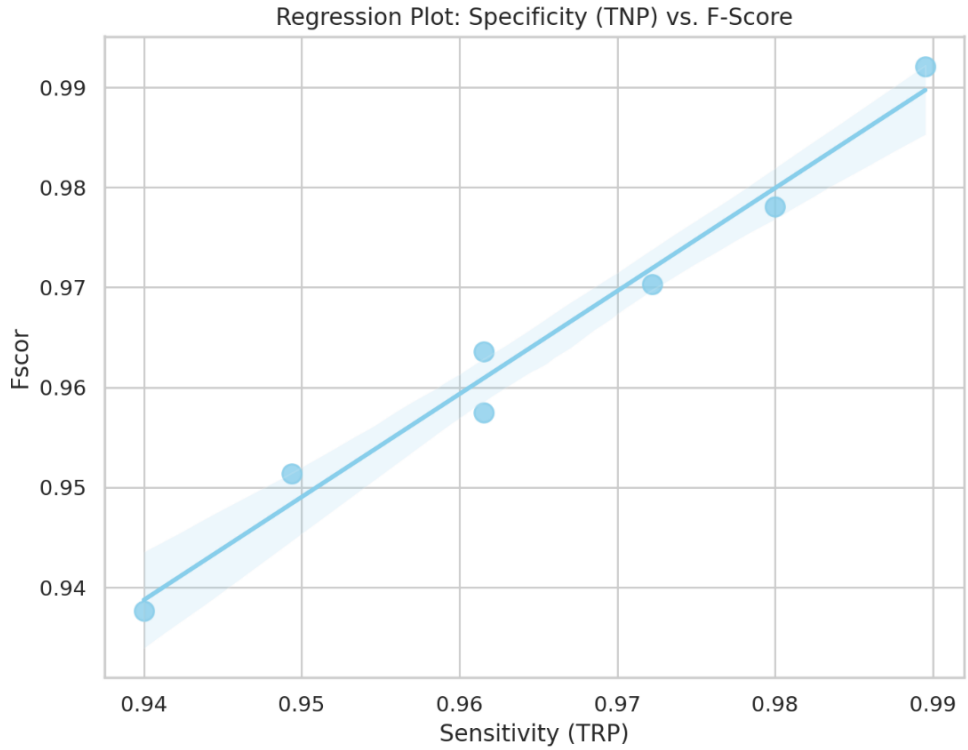

Figure 9: The Regression Plot of Sensitivity And F-score

Figure 10 also shows the sensitivity, defined as true equals positive rate, which gives information about the model's capacity to recognize positive instances. It is an essential metric for applications where it is more important to identify positive cases correctly and have relative flexibility on false negatives.

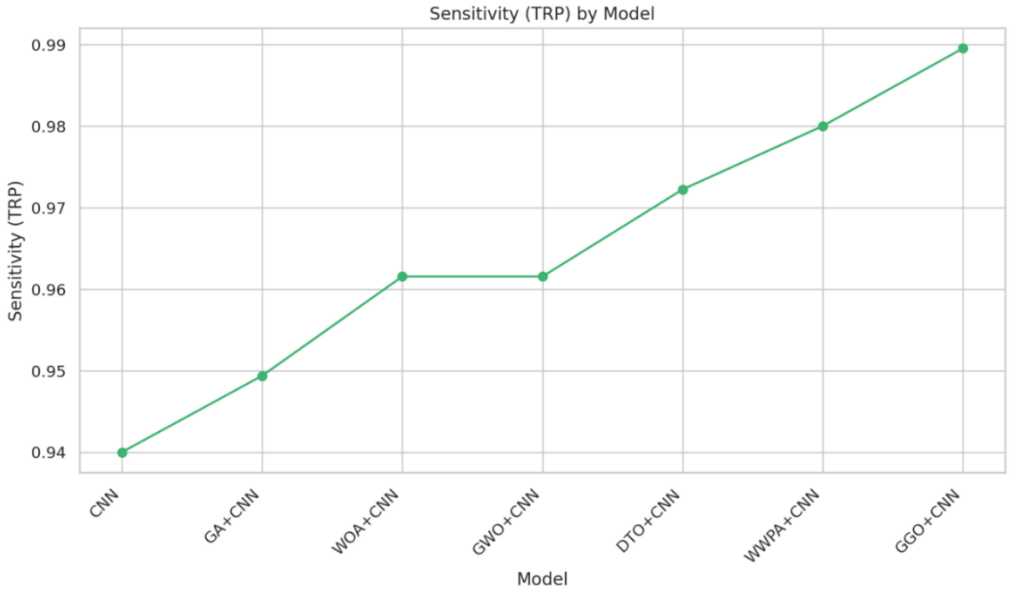

Figure 10: The Plot of Sensitivity of Different Approaches Based On CNN

Figure 11 outlines the N value; it is another measurement standard, though the preciseness of the definition in a given study may vary. The N value could be the number of iterations, nodes, or other related measures influencing the CNN performance and computation speed. By comparing these parameters, the following paper

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

seeks to correctly identify the effects of the various optimizations on the CNN and determine which algorithm is optimal for the given application. It is crucial for future research studies and practice solutions in optimizing and training neural networks.

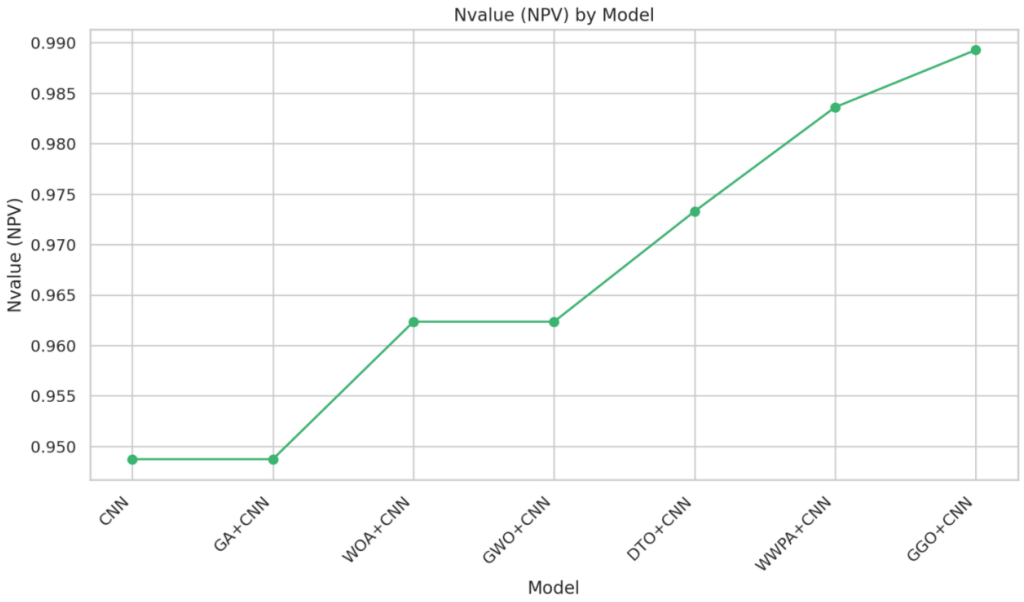

Figure 11: The Plot Of N-Value of Different Approaches Based On CNN

There are different representations of accuracy. Figure 12 shows the Kernal Density Estimation (KDE) plot, representing the accuracy values distribution. It is valid for a deep understanding of the spread, which may be helpful for evaluation. Figure 13 also shows a representation of accuracy in a Box Plot shape for different optimization models.

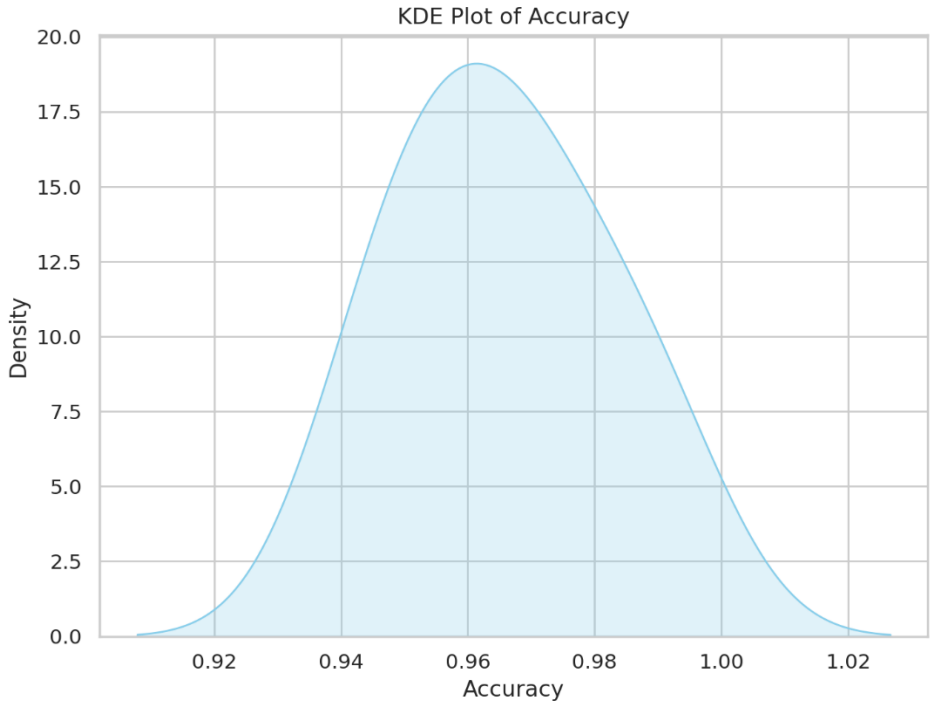

Figure 12: KDE Plot of Accuracy

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

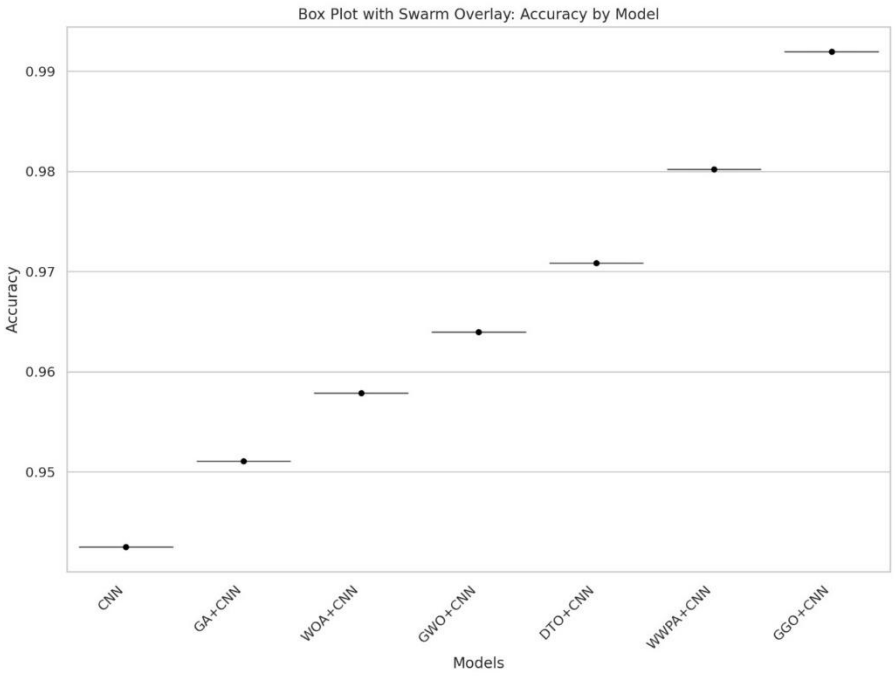

Figure 13: Accuracy Box Plot Swarm Overlay

5. conclusion

In this study, we applied a novel optimization method using GGO based on CNN for monkeypox classification. Monkeypox is a viral disease caused by orthopoxvirus that causes skin lesions all over the body. The processed methodology passed through some processes: firstly, it extracted the most common features of the MSID dataset by applying different DL pre-trained models, notably on AlexNet. Then, optimize extracted features to reduce the number of features by applying Greylag Goose Optimization (bGGO) in binary form to improve accuracy and remove redundant features. The classification process was applied to GGO, and different optimization algorithms and classification metrics were calculated; it reached an accuracy of .9919 and a sensitivity of .9895 for the proposed methodology. On the other hand, statistical analysis was conducted to prove the validity of the proposed methodology by applying the Wilcoxon signed test and analysis of variance (ANOVA) test. The results proved the hypothesis correct. We also introduced some plots that indicated the statical results (regression, QQ, Residual, homoscedasticity) plot.

References

[1] Altun, M., Gürüler, H., Özkaraca, O., Khan, F., Khan, J., & Lee, Y. (2023). Monkeypox detection using CNN with transfer learning. *Sensors*, *23*(4), 1783.**https://doi.org/10.3390/s23041783**

[2] Alharbi, A. H., Towfek, S. K., Abdelhamid, A. A., Ibrahim, A., Eid, M. M., Khafaga, D. S., ... & Saber, M. (2023). Diagnosis of monkeypox disease using transfer learning and binary advanced dipper throated optimization algorithm. *Biomimetics*, *8*(3), 313. **https://doi.org/10.3390/biomimetics8030313**

[3] Muhammed Kalo Hamdan, A., & Ekmekci, D. (2024). Prediction of monkeypox infection from clinical symptoms with adaptive artificial bee colony-based artificial neural network. *Neural Computing and Applications*, 1-16. https://doi.org/10.1007/s00521-024-09782-z

[4] Savaş, S. (2024). Enhancing Disease Classification with Deep Learning: a Two-Stage Optimization Approach for Monkeypox and Similar Skin Lesion Diseases. *Journal of Imaging Informatics in Medicine*, 1-23. https://doi.org/10.1007/s10278-023-00941-7

[5] Asif, S., Zhao, M., Li, Y., Tang, F., Ur Rehman Khan, S., & Zhu, Y. (2024). AI-Based Approaches for the Diagnosis of Mpox: Challenges and Future Prospects. *Archives of Computational Methods in Engineering*, 1-33. https://doi.org/10.1007/s11831-024-10091-w

[6] Abdelhamid, A. A., El-Kenawy, E. S. M., Khodadadi, N., Mirjalili, S., Khafaga, D. S., Alharbi, A. H., ... & Saber, M. (2022). Classification of monkeypox images based on transfer learning and the Al-Biruni

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

Earth Radius Optimization algorithm. *Mathematics*, *10*(19), 3614. **https://doi.org/10.3390/math10193614**

[7] Jaradat, A. S., Al Mamlook, R. E., Almakayeel, N., Alharbe, N., Almuflih, A. S., Nasayreh, A., ... & Bzizi, H. (2023). Automated monkeypox skin lesion detection using deep learning and transfer learning techniques. *International Journal of Environmental Research and Public Health*, *20*(5), 4422. **https://doi.org/10.3390/ijerph20054422**

[8] Saleh, A. I., & Hussien, S. A. (2024). Monkeypox diagnosis based on Dynamic Recursive Gray wolf (DRGW) optimization. *Biomedical Signal Processing and Control*, *87*, 105483. https://doi.org/10.1016/j.bspc.2023.105483

[9] Surati, S., Trivedi, H., Shrimali, B., Bhatt, C., & Travieso-González, C. M. (2023). An Enhanced Diagnosis of Monkeypox Disease Using Deep Learning and a Novel Attention Model Senet on Diversified Dataset. *Multimodal Technologies and Interaction*, *7*(8), 75. https://doi.org/10.3390/mti7080075

[10] Lakshmi, M., & Das, R. (2023). Classification of monkeypox images using LIME-enabled investigation of deep convolutional neural network. *Diagnostics*, *13*(9), 1639. https://doi.org/10.3390/diagnostics13091639

[11] Asif, S., Zhao, M., Li, Y., Tang, F., & Zhu, Y. (2024). CGO-Ensemble: Chaos Game Optimization Algorithm-Based Fusion of Deep Neural Networks for Accurate Mpox Detection. *Neural Networks*, 106183. https://doi.org/10.1016/j.neunet.2024.106183

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

[13] Bala, D., Hossain, M. S., Hossain, M. A., Abdullah, M. I., Rahman, M. M., Manavalan, B., ... & Huang, Z. (2023). MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification. *Neural Networks*, *161*, 757-775.

[14] Campana, M. G., Colussi, M., Delmastro, F., Mascetti, S., & Pagani, E. (2024). A Transfer Learning and Explainable Solution to Detect mpox from Smartphones images. *Pervasive and Mobile Computing*, *98*, 101874. https://doi.org/10.1016/j.pmcj.2023.101874

[15] Campana, M. G., Colussi, M., Delmastro, F., Mascetti, S., & Pagani, E. (2023). Mpox Close Skin Images. https://doi.org/10.5281/zenodo.7948350

[16] Ahsan, M. M., Alam, T. E., Haque, M. A., Ali, M. S., Rifat, R. H., Nafi, A. A. N., ... & Islam, M. K. (2024). Enhancing Monkeypox diagnosis and explanation through modified transfer learning, vision transformers, and federated learning. *Informatics in Medicine Unlocked*, *45*, 101449. https://doi.org/10.1016/j.imu.2024.101449

[17] Ahsan, M. M., Uddin, M. R., & Luna, S. A. (2022). Monkeypox image data collection. *arXiv preprint arXiv:2206.01774*.

[18] Ahsan, M. M., Uddin, M. R., Ali, M. S., Islam, M. K., Farjana, M., Sakib, A. N., ... & Luna, S. A. (2023). Deep transfer learning approaches for Monkeypox disease diagnosis. *Expert Systems with Applications*, *216*, 119483. https://doi.org/10.1016/J. ESWA.2022.119483.

[19] Demir, F. B., Baygin, M., Tuncer, I., Barua, P. D., Dogan, S., Tuncer, T., ... & Acharya, U. R. (2024). MNPDenseNet: Automated Monkeypox Detection Using Multiple Nested Patch Division and Pretrained DenseNet201. *Multimedia Tools and Applications*, 1-23. https://doi.org/10.1007/s11042-024-18416-4

[20] Bala, D., & Hossain, M. S. (2022). Monkeypox skin images dataset (msid). *Mendeley Data*, *6*, 2023.

[21] Kundu, D., Rahman, M. M., Rahman, A., Das, D., Siddiqi, U. R., Alam, M. G. R., ... & Ali, Z. (2024). Federated Deep Learning for Monkeypox Disease Detection on GAN-Augmented Dataset. *IEEE Access*. doi={10.1109/ACCESS.2024.3370838}}

[22] M. Ahsan. "MONKEYPOX IMAGE DATA COLLECTION, https://github.com/mahsan2/Monkeypoxdataset-2022." (Accessed 01.09.2022).

[23] Thorat, R., & Gupta, A. (2024). Transfer learning-enabled skin disease classification: the case of monkeypox detection. *Multimedia Tools and Applications*, 1-19. https://doi.org/10.1007/s11042-024-18750-7

[24] Yadav, S., & Qidwai, T. (2024). Machine learning-based monkeypox virus image prognosis with feature selection and advanced statistical loss function. *Medicine in Microecology*, *19*, 100098. https://doi.org/10.1016/j.medmic.2024.100098

[25] Kaggle. Monkeypox skin lesion dataset. Available: https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset; 2022.

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024

[26] Raha, A. D., Gain, M., Debnath, R., Adhikary, A., Qiao, Y., Hassan, M. M., ... & Islam, S. M. S. (2024). Attention to Monkeypox: An Interpretable Monkeypox Detection Technique Using Attention Mechanism. *IEEE Access*. doi={10.1109/ACCESS.2024.3385099}}

[27] Dermnet. Accessed: Sep. 2023. [Online]. Available: http://www.dermnet.com/

[28] Khafaga, D. S., Ibrahim, A., El-Kenawy, E. S. M., Abdelhamid, A. A., Karim, F. K., Mirjalili, S., ... & Ghoneim, M. E. (2022). An Al-Biruni earth radius optimization-based deep convolutional neural network for classifying monkeypox disease. *Diagnostics* **https://doi.org/10.3390/diagnostics12112892**

[29] Monkeypox Skin Images Dataset (MSID). Available online: https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset (accessed on 2 October 2022

[30] Bala, D., Hossain, M. S., Hossain, M. A., Abdullah, M. I., Rahman, M. M., Manavalan, B., ... & Huang, Z. (2023). MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification. *Neural Networks*, *161*, 757-775. https://doi.org/10.1016/j.neunet.2023.02.022.

[31] Lakshmi, M., & Das, R. (2023). Classification of monkeypox images using LIME-enabled investigation of deep convolutional neural network. *Diagnostics*, *13*(9), 1639. **https://doi.org/10.3390/diagnostics13091639**

[32] Ahsan, M. M., Uddin, M. R., Ali, M. S., Islam, M. K., Farjana, M., Sakib, A. N., ... & Luna, S. A. (2023). Deep transfer learning approaches for Monkeypox disease diagnosis. *Expert Systems with Applications*, *216*, 119483. https://doi.org/10.1016/j.eswa.2022.119483

[33] El-kenawy, E. S. M., Khodadadi, N., Mirjalili, S., Abdelhamid, A. A., Eid, M. M., & Ibrahim, A. (2024). Greylag goose optimization: Nature-inspired optimization algorithm. *Expert Systems with Applications*, *238*, 122147. https://doi.org/10.1016/j.eswa.2023.122147

Doi: https://doi.org/10.54216/FPA.160206
Received: December 16, 2023 Revised: February 18, 2024 Accepted: May 27, 2024