

Feature Selection Based on Dragonfly Optimization for Psoriasis Classification

N. Bhaswanth¹, Dr. D. VenkataSekhar², Dr. M. PurushothamReddy³

Submitted: 26/01/2024 Revised: 04/03/2024 Accepted: 12/03/2024

Abstract: Due to their efficiency and higher disease detection accuracy than traditional methods, metaheuristic algorithms are prominent in healthcare data analysis. The Dragonfly Algorithm (DFA) uses wrapper feature selection to find illness categorization features. DFA was used to pick features and recognize skin illnesses using CNN, VGG19, and EfficientNet-B2 classifiers. The classifier's accuracy using a given set of features from the training dataset determined the Dragonflies' fitness value in each iteration. The experimental study showed DFA's precision and little loss. Two EfficientNet-B2 and VGG19-based CNN models were created in tandem to analyze performance. DermNet NZ and ISIC 2019 were used to train these models. Disease taxonomy helped the models classify. Both datasets classified all eight skin illnesses with an average accuracy of 88.5% and 0.0003 loss. This shows that Deep Learning can classify a wide range of skin conditions with near-human accuracy and reproducibility. These models can also help clinicians perform large-scale screenings utilizing clinical or dermoscopic images for real-time skin disease diagnosis, improving healthcare practices and patient outcomes. This work advances skin disease diagnostics and shows the medical potential of Deep learning.

Keywords: skin disease, Deep Learning, Dragonfly Optimization Algorithm, feature selection

1. Introduction

The skin is the most sensitive and weak part of the human body, making it more susceptible to various external factors. Among these factors, sunburn caused by ultraviolet (UV) rays from the sun is a significant contributor to skin damage [1]. Additionally, the skin is exposed to various diseases caused by UV rays, fungal or viral infections, and pollution in the surrounding environment [2]. Skin diseases include a wide range of medical conditions that affect the skin's appearance, texture, and functionality. These conditions can vary in severity, from mild issues like acne or rashes to more serious diseases like psoriasis, eczema, or skin cancer. Skin diseases can affect people of all ages, races, and genders, and they can arise due to genetic factors, exposure to irritants or allergens, infections, hormonal imbalances, or autoimmune disorders. Skin diseases can manifest in various ways, such as redness, swelling, itching, dryness, scaling, blistering, or changes in skin color or texture. Early diagnosis and proper treatment of skin diseases are essential for maintaining healthy skin and preventing complications. Feature selection (FS) is a crucial pre-processing strategy that significantly enhances data mining techniques. By removing irrelevant and noisy data from the dataset, FS increases classification accuracy. There are two main FS

techniques, filters and wrappers [3], which employ different assessment criteria. While wrapper techniques use a learning algorithm (e.g., classification) to assess the feature subset, filters rely on data-based methods (e.g., information gain) [4]. The goal of feature selection is to identify an optimal subset of features from a large pool while retaining important information. However, this process can be challenging as it involves a combinatorial explosion of potential subsets, making it computationally expensive. Early diagnosis of skin diseases is vital to prevent permanent scarring, disfigurement, and life-threatening conditions like skin cancer. Additionally, it helps to control the spread of infectious skin diseases such as bacterial or viral infections. Skin disease classification tools that employ machine learning methods have garnered significant attention for their promising results [5]. Combining these methods improves classifier prediction. Data mining, artificial intelligence, and meta-heuristic algorithms can be used to create an automated disease detection system [6]. Different types of sensors can improve data collecting but complicate predicting, decoding, and recognition [7]. Meta-heuristic algorithms like Artificial Immune Systems, Grey Wolf Optimisation, Flower Pollination, Whale Optimisation, Chicken Swarm Optimisation, Lion Optimisation, Grasshopper Optimisation, Crow Search, and Social Spider Optimisation are crucial for feature extraction and selection, especially in disease diagnosis and early detection. [8]. The primary objective of this paper is to develop a skin disease image classifier that exhibits high accuracy in identifying various skin conditions based on images uploaded by users. The classifier will be integrated into a user-friendly website, allowing patients to upload skin

¹ Research scholar, Dept. of Information Technology, Faculty of Engineering and Technology, Annamalai University, Tamil Nadu, India

² Professor, Dept. of Information Technology, Annamalai University, Faculty of Engineering and Technology Tamil Nadu, India. ramavenkatasekar@yahoo.co.in

³ Professor, Dept. of Information Technology, Institute of Aeronautical Engineering, Hyderabad, India purushotham.mps@gmail.com

* Corresponding Author Email: bhaswanth1@gmail.com

m

images and receive accurate diagnoses. In pursuit of this objective, the machine learning algorithm utilized in this paper will be designed to work with a comprehensive dataset of skin images obtained from the esteemed DermNet NZ Image Library and the renowned ISIC 2019 Challenge websites [9][10]. The specific goals of this study encompass:

- i. Develop an advanced skin disease image classifier using the Dragonfly Optimization Algorithm (DFA) for feature selection, addressing issues related to time complexity and random motion.
- ii. Integrate the image classifier into a user-friendly platform, providing patients with convenient and affordable access to dermatological care.
- iii. Enhance the speed and accuracy of skin disease diagnosis for both dermatologists and patients, resulting in better treatment *outcomes*.

2. Background and Related Work

Recently, there has been a growing trend in the healthcare field to employ efficient and reliable metaheuristic algorithms for disease diagnosis. These algorithms have demonstrated superior performance compared to traditional methods, making them a preferred choice for feature selection and constructing classification models with fast training, improved accuracy, and reduced complexity. Among various medical conditions, skin diseases are of particular concern due to the skin's susceptibility to external factors like UV rays, fungal or viral infections, and pollution. Convolutional Neural Networks (CNNs) have emerged as powerful computer models, drawing inspiration from the biological visual cortex. They exhibit remarkable efficiency and accuracy in image classification tasks, and researchers have effectively utilized them for classifying diseases from medical images. An important focus of research efforts has been on selecting the most relevant features to enhance diagnostic accuracy and reduce noise in skin disease data.

In one study [11], a hybrid approach employing Chi-Square, Information Gain, and Principal Component Analysis (PCA) for feature extraction was used to select a representative subset of features from skin disease datasets. Six base learners, including Gaussian Naïve Bayesian, K-Nearest Neighbour, Decision Tree, Support Vector Machine, Random Forest, and Multilayer Perceptron, were employed to evaluate prediction performance. Additionally, Boosting, Bagging, and Stacking ensemble techniques were incorporated to enhance the model's performance further, demonstrating improved results compared to using the entire dataset.

In [12], the authors proposed a novel method that involved the application of six different data mining

classification techniques and the integration of ensemble approaches like Bagging, AdaBoost, and Gradient Boosting to predict skin disease classes. To select the most important features, a feature importance method was employed, resulting in the identification of 15 significant features. A subset of the original dataset was then obtained for comparison with the ensemble approach. The results showed a notable increase in accuracy for dermatological prediction when compared to individual classifiers and even outperformed the feature selection subset method.

In [13], the authors used different feature selection algorithms, including PCA, Information Gain, and Chi-square, on skin disease prediction using data mining techniques investigated. They carried out the classification task using Random Forest, C4.5 Decision Trees, and Functional Tree algorithms. While some of the feature selection algorithms demonstrated improvements in accuracy and sensitivity, the gains were relatively modest, typically around 1%.

In [14], the authors adopted the ensemble machine-learning approach for the early prediction of skin cancer that incorporated an enhanced genetic algorithm technique. A Convolutional Neural Network (CNN) model, specifically ResNet-16, was utilized to extract features from the images automatically. An Enhanced Genetic Algorithm (EGA) was then applied for feature selection. The proposed model achieved higher accuracy and demonstrated effective performance when compared to other techniques, such as the Support Vector Machine (SVM).

In [15], the authors proposed dimensionality reduction and feature selection were integrated with ensemble learning methods and various classification techniques for disease prediction. The approach demonstrated promising results in the prediction of multiple medical diseases, highlighting its potential for versatile applications.

In [16], the authors focused on cluster algorithms for stratifying skin disorders. However, these cluster algorithms had drawbacks, including experimental noise and challenges in interpretation. To overcome these limitations, the study introduced a multi-objective particle swarm approach that integrated hybrid particle swarm and moth flame optimization to optimize cluster distance parameters for K -means clustering. This optimization resulted in improved accuracy and the ability to predict different classes in clustering. Overall, the application of metaheuristic algorithms and ensemble techniques has shown significant potential for skin disease prediction and other medical disease prediction tasks. These methods can revolutionize disease diagnosis and enhance healthcare outcomes by providing accurate and efficient classification models.

3. Proposed Model

Figure 1 depicts the complete methodology and different stages, such as data preprocessing, feature extraction, and selection, which were employed in the study. By applying classification algorithms, we assessed the accuracy and loss of predictions for eight different skin disease classes. A comparative analysis was conducted to determine the most effective prediction approach.

3.1. Dataset analysis

The dataset utilized in this research comprises images sourced from the DermNet NZ Image Library and ISIC 2019 Challenge. This dataset was curated specifically for the study of skin diseases and their classification. The dataset consists of eight distinct classes of skin diseases, namely Acne, Eczema, Actinic keratosis, benign keratosis, Melanoma, Psoriasis, Basal cell carcinoma, and Ringworm. In Table 1, we provide a summary of these datasets, including the total number of images, the categories of diseases, the count of labeled images, and the inclusion of binary segmentation masks.

Table 1. Summary of publicly available skin lesion datasets

Dataset	Disease Category	Image Number	Labeled Images
ISIC 2019	8	33,569	25,331
DermNet NZ	23	19,500	All

3.2 Pre-processing

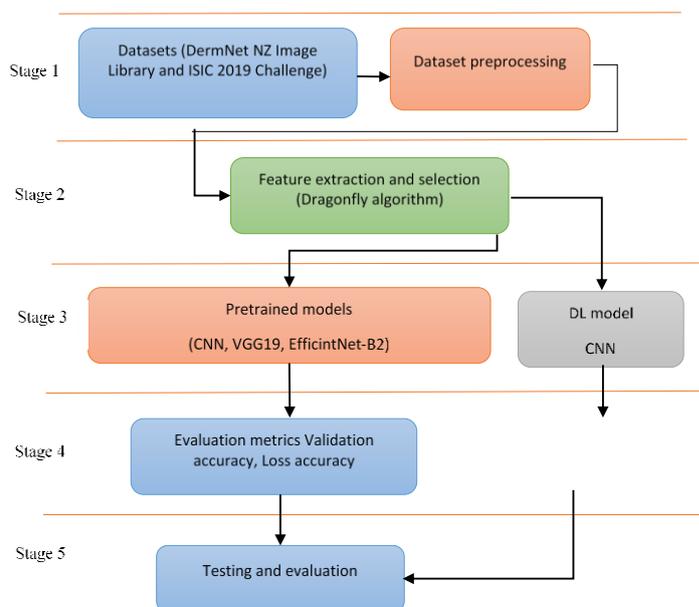


Fig 1. Proposed work methodology

However, to avoid excessive biases, data augmentation is restricted to generate a controlled number of augmented

In this study, the preprocessing stage comprises two phases. In the first phase, data balancing is performed for each class by aggregating the mean and median attribute values. This ensures that the data is evenly distributed among the classes. Subsequently, the balanced data is normalized using Z-score normalization, as described in equation (1). The normalization step is crucial for preventing overfitting issues in the classifier.

$$Z - score = \frac{Score - Mean}{Standard deviation} \quad (1)$$

Z-score normalization is a technique used to standardize multiple parameters onto the same scale. This process involves calculating the Z-score, which is a crucial statistical measure that allows data analysts to assess the probability of a score occurring within a normally distributed dataset. This allows for the comparison of scores from diverse datasets by converting them into the standard normal distribution.

Dataset splitting. The dataset is split into three subsets: training, validation, and testing. The training set receives 80% of the images in an 8:1:1 ratio.

Data Augmentation. Data augmentation is another important step in this study, which involves applying various transformations to the images to increase the size and diversity of the dataset.

images, typically within the range of 200 to 400 images for each class. By augmenting the data, the researchers aim to

overcome bias issues and enhance the size of the dataset, allowing them to apply deep learning techniques more effectively.

3.3 Feature Extraction and Selection

The primary goal of feature extraction and selection in this study is to identify an optimal set of features that have strong discriminative capabilities, particularly for classifying skin lesion images. The researchers prefer utilizing expert knowledge and clinical experience to guide this process, as it can effectively and efficiently lead to the extraction and selection of relevant features.

Dragonfly algorithm

The Dragonfly algorithm mimics dragonfly swarm foraging and migration. Local movement and flight path mutations show the algorithm's exploitation phase as they form subgroups to pursue insects in different places. Dragonflies flying together over large distances resemble the algorithm's global exploration phase. Each dragonfly represents a search space solution and moves according to five behaviors: Separation (S), Alignment (A), Cohesion (C), Attraction to a food source (F), and Distraction from an adversary (E). Dragonflies avoid static barriers and other dragonflies with Separation behavior. Alignment behavior ensures dragonflies match their neighbors' velocities. The Cohesion behavior encourages dragonflies to travel towards the neighborhood center, promoting group cohesion. Dragonflies seek food (optimal solutions) and avoid threats (bad solutions) to survive. The weights associated with these five behaviors are initialized randomly for each dragonfly at the beginning of the algorithm.

They are adaptively adjusted during the iterative process to ensure convergence to the global optima. As the optimization progresses, the radius and Euclidean distance between dragonfly neighbors increase, allowing for better exploration of the search space. The equations governing the five behaviors and the radius are used to steer the movement of the dragonflies and guide the optimization process toward finding the optimal solutions.

$$S_i = - \sum_{j=1}^N (X - X_j) \quad (2)$$

$$A_i = \frac{\sum_{j=1}^N \Delta X_j}{N} \quad (3)$$

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (4)$$

$$F_i = X^+ - X \quad (5)$$

$$E_i = X^- + X \quad (6)$$

$$r = \frac{ub - lb}{4} + 2(ub - lb) \frac{n}{N_{max}} \quad (7)$$

In equations (2) to (6), the symbols S_i , A_i , C_i , F_i , and E_i represent the magnitudes of the separation, alignment, cohesion, attraction to a food source, and distraction from enemy behavior, respectively, for the i th individual in the dragonfly swarm. The variables X represent the position and velocity of the current individual, while N represents the total number of neighboring individuals in the swarm. The symbols X_j and ΔX_j indicate the position and velocity of the j th neighboring individual, and X^+ and X^- denote food source positions and natural enemies, respectively. In equation (8), the variables ub and lb represent the upper and lower limitations of the search space, respectively. The symbols n and N_{max} represent the current iteration and the maximum number of iterations in the Dragonfly Algorithm (DA). During this implementation, the positions of the food source and natural enemies are determined based on the significance associated with the different behaviors. We consider the optimal position of each iteration as the position of the food source, while the worst position corresponds to the natural enemies. This reflects the influence of various behaviors on the movement of the dragonflies.

The DA assumes that the overall behavior of the dragonflies is a combination of five behaviors: separation, alignment, cohesion, attraction to a food source, and distraction from an enemy. These behaviors guide the dragonflies in their movement and decision-making process. When the current individual in the dragonfly swarm has at least one neighboring dragonfly, its velocity and position are updated using specific equations. These equations ensure that the dragonfly aligns its movement with the nearby individuals and adjusts its position accordingly. This dynamic updating process helps the swarm converge towards promising regions in the search space and enhances the algorithm's capability for optimization and exploration.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + \omega \Delta X_t \quad (8)$$

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (9)$$

In the Dragonfly Algorithm, the weights assigned to the five behaviors are represented by s , a , c , f , and e . The inertia weight is denoted by ω , and the current iteration counter is represented by t . When there are no neighboring dragonflies around the current individual, it updates its position using the Levy Flight equation, as described in (10). The

incorporation of this Levy Flight equation enhances the randomness, chaotic behavior, and global search capability of the dragonflies, thereby contributing to their effectiveness in exploring the search space. This random walk strategy further improves the algorithm's ability to search for optimal solutions efficiently.

$$X_{t+1} = X_t + Levy(d) \times X_t \quad (10)$$

In the context of the DA, the variable d represents the dimension of each individual dragonfly in the search space.

$$Levy(x) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{1/\beta}} \quad (11)$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin \pi \beta / 2}{\Gamma(1 + \beta / 2) \times \beta \times 2^{\beta - 1/2}} \right) \quad (12)$$

In the DA, the variables r_1 and r_2 are random numbers generated from a uniform distribution within the range $[0, 1]$. Additionally, β is a constant that typically has a value of 1.5.

3.4 Classification algorithms

VGG-19 Pretrained CNN Model

Figure 2 depicts the VGG-19 model architecture with 19 neural layers [17], and convolutional layers have various filter widths and pooling layers.

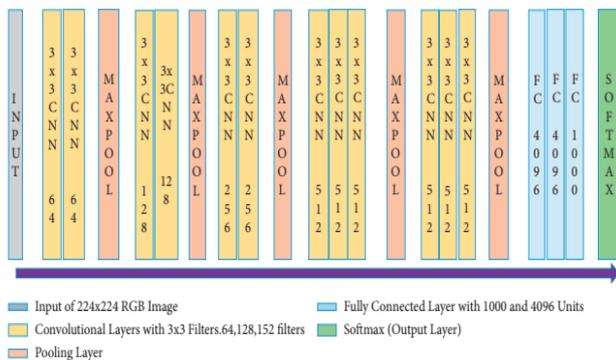


Fig. 2: Architecture of the VGG-19 pretrained CNN model

These layers extract hierarchical information from input images to help the model recognize patterns and representations. The model can classify complicated images because it uses many convolutional and pooling layers to capture patterns. VGG-19 downsamples feature maps using max-pooling layers. Downsampling reduces the computational complexity of the model and allows it to handle huge datasets efficiently. The VGG-19 model ends with 4096-unit fully connected (FC) layers and a softmax layer to classify skin diseases. The model was pre-trained on a huge image dataset to extract relevant features. Pre-

training saves computational time and resources by avoiding model retraining. By using a pre-trained CNN model, the investigators leveraged existing models that are already trained on large image datasets, eliminating the need to train the model from scratch, which saves time and resources.

3.5 Architecture of VGG-19 Pretrained CNN Model.

Figure 2 shows this study's pre-trained VGG-19 CNN model architecture. Layer after layer, convolutional layers with varied filter widths, and pooling layers that reduce layer volume are learned. After pooling and convolutional layers, a 4096-unit FC layer and softmax output layer are produced. The VGG-19 pretrained CNN model has trainable layers to improve accuracy and efficiency. The classification model has four convolutional layers with max-pooling processes. These four convolutional layers collect features from example images to diagnose the five types of psoriasis.

EfficientNet overview.

EfficientNet models scale convolutional neural network layers uniformly using a fixed ratio. This scaling method improves classification accuracy without changing hidden layer functions or surpassing system memory and computing capabilities.

Traditional class activation map generation

We enhanced the EfficientNet classification model to generate Class Activation Maps (CAMs).

Figure 3 illustrates the two main sections of the model's architecture. The initial convolutional layers take data from the input image to generate a multi-channel feature model with a lower spatial dimension. Following fully connected classification layers, feature representations are classified. A Global Average Pooling (GAP) layer connects these two portions by averaging feature channel values to form a vector. Use the final fully connected layer's learnable parameters to weight the feature vector representation to determine each class's classification probability. Creating a Class Activation Map for a class bypasses the GAP layer. The feature extraction layer's output is directly generated by taking the Hadamard product of the fully connected layer's weight and the feature encoding. The output is upsampled to match the original image size, employing bilinear interpolation after normalizing the data to 0–255. This process ensures that the generated Class Activation Map highlights the regions in the image that are most relevant for making predictions about a specific class.

ScoreCAM generation.

This masked input image retains the most pertinent regions corresponding to the activation maps, thus highlighting the areas that significantly contribute to the model's predictions. The innovative approach adopted by ScoreCAM results in the generation of more accurate and informative CAMs, ultimately enhancing the interpretability of the neural

network's decisions. ScoreCAM avoids overestimating the importance of less crucial areas in the image, leading to a more reliable and localized representation of the regions that truly influence the model's predictions. This technique not only enhances the CAM generation process but also provides valuable insights into the neural network's inner workings, making it a valuable tool for researchers and practitioners in computer vision and image classification M^k as:

$$M^k = A^k * I \quad (13)$$

Where *the k'th* neural network layer is represented as, and the input image is denoted as I .

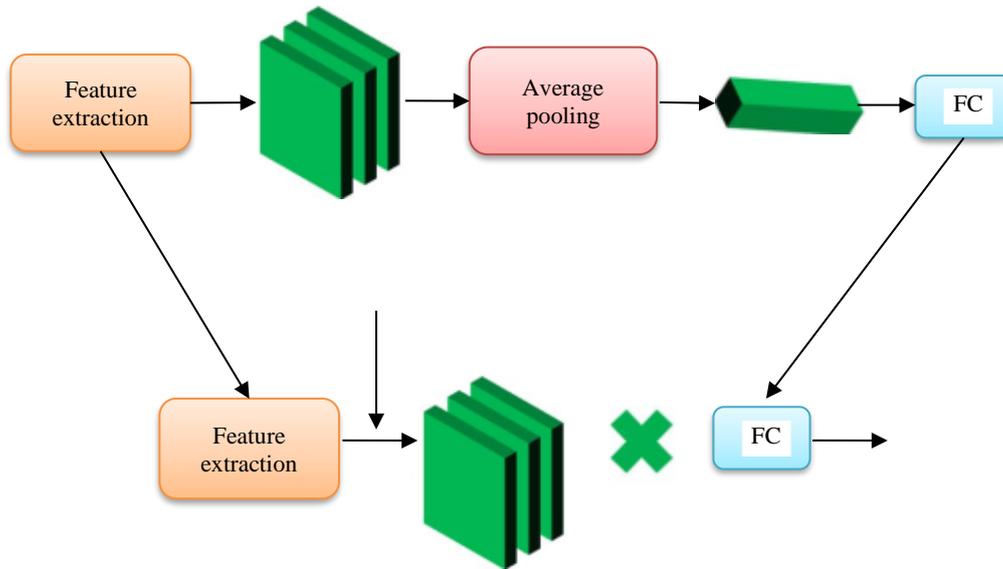


Fig .3: Efficientnet-B2 (FC denotes the fully connected final layer)

After applying the Hadamard multiplication to obtain the masked input, it is fed through the neural network. The final scores S^k are computed by softmax the output, indicating "activation map."

The weighted total of all target class activations generates a single activation map analogous to the traditional CAM technique.

$$A^{tot} = \sum_{k=1}^K S_k A^k \quad (14)$$

Ultimately, a pixel-wise ReLU is employed to process, effectively eliminating all negative values. These negative values indicate features that had no positive impact, allowing the model to focus solely on the relevant and influential features.'

4. Results and Discussion

This study used Matlab R2019a, Windows 10 Professional Edition, an Intel (R) Core (TM) i5 CPU@2.30 GHz processor, 8 GB RAM, and a simulation platform. Python code was running on Google Colab to test the Dragonfly Algorithm and compare it to other approaches.

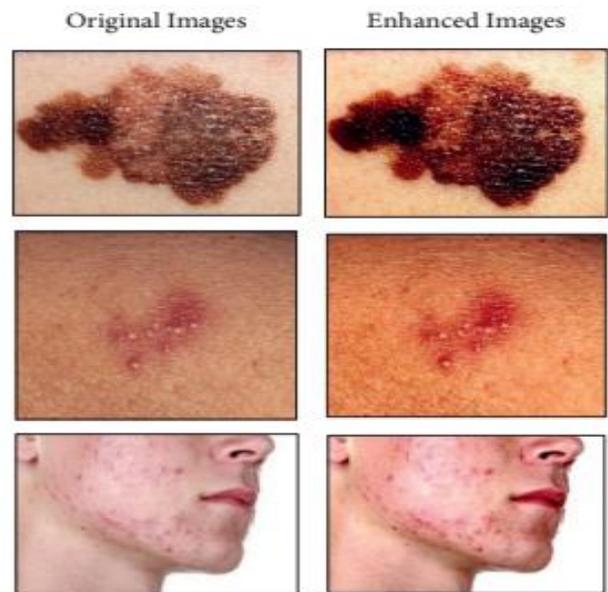
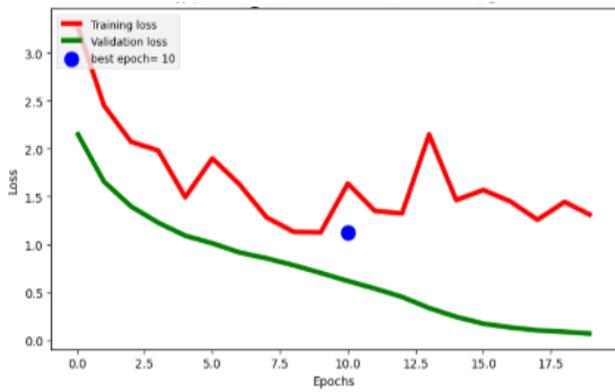
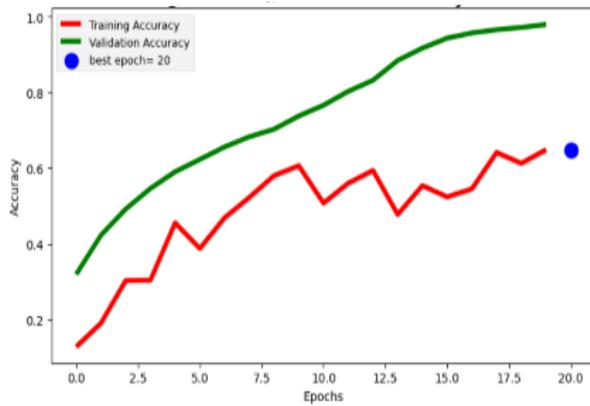


Fig. 4: Examples of preprocessed images.

The study presents a dragonfly-based wrapper feature selection strategy to improve accuracy with a smaller feature set. We tested this feature selection strategy on a set of well-known datasets from the data repository. The Keras API on TensorFlow and Python on Anaconda was used for modeling. Figure 4 shows filter-applied images following image preprocessing.



(a)



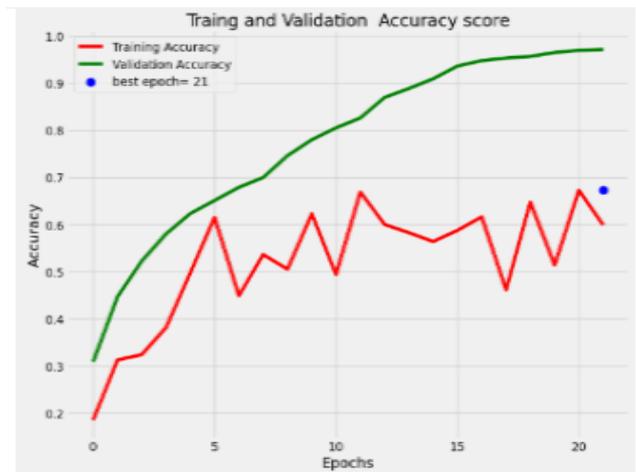
(b)

Fig.5: Model 1 (Baseline CNN Model) before augmentation (a) CNN model loss versus epoch (b) CNN model accuracy score versus epoch

Figure 5 (a) displays the CNN model's loss versus epoch before the augmentation process is applied to the datasets. This graph illustrates the model's performance at each epoch. As epochs rise, validation loss falls while training loss stays low. Comparing the two graphs shows that the training error stabilizes at test epoch 10. Figure 5 (b) illustrates the CNN model's accuracy score versus epoch before applying the augmentation process to the datasets. As the epoch number increases, the validation accuracy shows an upward trend while the training accuracy remains relatively low. By comparing both graphs side by side, it becomes apparent that, at test epoch = 20, the training accuracy score reaches 60%. In Figure 6 (a), the graph depicts the CNN model's loss versus epoch after the datasets undergo the augmentation process. This graph provides insights into the model's performance at each epoch. When analyzing both graphs together, we observe that the training error stabilizes at test epoch = 12, the training error stabilizes.



(a)



(b)

Fig.6: Model 2 (Baseline CNN Model) after augmentation (a) CNN model loss versus epoch (b) CNN model accuracy score versus epoch

Figure 6 (b) illustrates the CNN model's accuracy score versus epoch after applying the augmentation process to the datasets. As the epoch number increases, the validation accuracy shows an upward trend while the training accuracy remains relatively low. By comparing both graphs side by side, it becomes apparent that, at test epoch = 21, the training accuracy score reaches 68.5%.

Figure 7 shows that the model's accuracy and loss values for both training and validation datasets fluctuate until the 15th epoch, demonstrating instability during the initial training. After the 15th epoch, behavior changes, and training and validation values stabilize at the 16th. At the 18th epoch, the VGGNet19 loss function reaches its lowest value. The model's predictions are growing increasingly accurate, reducing the difference between projected and actual values. VGGNet19's best accuracy score is 44.5% at the 16th epoch.

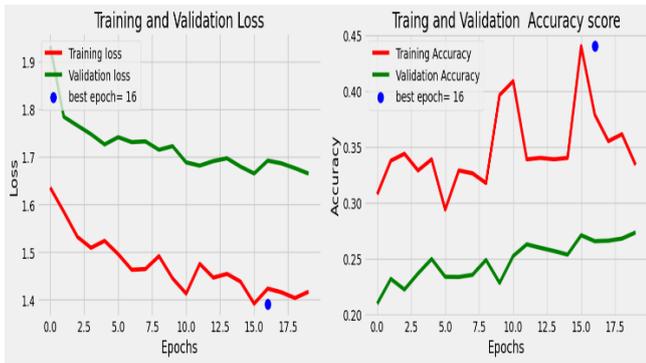


Fig.7: Model 3 VGG19

Figure 8 shows that training has a higher model accuracy value than validation after the 15th period. EfficientNetB2 model loss is modest at 20 epochs, and accuracy is 88.5%. EfficientNetB2 model training ended after 21 accuracy and 20 loss epochs. These epochs stopped training with equal model accuracy and model loss calculations. Model accuracy remained constant after 20 epochs.

Performance of models on reduced after feature selection

The findings reveal that clinical features have a more significant impact on achieving higher accuracy in disease prediction compared to histopathological features. Table 2 presents the accuracy scores obtained from three methods, employing eight different classification algorithms on the reduced dataset after feature selection (FS).

Table 2: Output of accuracy and execution time on the reduced dataset.

Models	Accuracy		Time in sec	
	Without feature selection	With feature selection	Without feature selection	With feature selection
CNN	66.25	71.24	66.32	52.31
VGG19	55.41	57.84	98.74	78.64
EfficientNet-B2	87.51	89.55	23.51	13.24

Additionally, the results reveal that the accuracy achieved after FS surpasses the accuracy obtained without FS. The implementation of the EfficientNet-B2 model, integrated with the Dragonfly optimization algorithm for FS, exhibited the highest accuracy of 87.51% with a relatively short execution time of 13.24 seconds, outperforming the CNN and VGG19 models. Table 3 provides the values of performance indicators, including sensitivity and

specificity, which were computed based on their respective formulas.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (15)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (16)$$

Moreover, the Sensitivity and specificity achieved after feature selection on the reduced dataset are higher compared to the values obtained without feature selection. Also, the EfficientNet-B2 model, utilizing the Dragonfly optimization algorithm for feature selection, demonstrated the highest accuracy of 90.12% and Specificity of 90.24%, outperforming the CNN and VGG19 models.

Table 3: Output of Sensitivity and Specificity on the reduced dataset.

Models	Sensitivity (%)		Specificity (%)	
	Without feature selection	With feature selection	Without feature selection	With feature selection
CNN	65.23	71.24	66.22	67.54
VGG19	58.23	57.84	67.14	75.62
EfficientNet-B2	88.22	90.12	89.24	90.24

5. Conclusion

This paper introduces a novel wrapper feature selection strategy that leverages the Dragonfly optimization technique to create an optimal subset of features for accurate skin cancer classification. Throughout the iterative process, the dragonfly population is classified based on their fitness values, and individual positions are updated accordingly, guiding the algorithm's evolution. We conducted experiments using eight classes of skin diseases to assess the effectiveness of the Dragonfly Algorithm (DFA). In this study, two Convolutional Neural Network (CNN) models were developed based on the EfficientNet-B2 and VGG19 architectures, with data augmentation applied to both labeled and unlabeled datasets. The proposed EfficientNet-B2 classifier model demonstrated promising results, achieving an impressive 88.5% accuracy rate on the test set for identifying various skin diseases. Moreover, the approach can be extended to other skin disorders, and its integration with other DL approaches further enhances its capabilities. Additionally, future research can explore the application of this method to "Psoriasis Area and Severity

Index" (PASI) scoring, expanding its utility in dermatological diagnostics and providing valuable insights into the severity assessment of psoriasis cases. The proposed approach shows promise for advancing the field of skin disease diagnosis and classification, paving the way for more efficient and accurate diagnostic tools in the future.

Conflicts of interest

The authors declare no conflicts of interest

References.

- [1] Alcon, J.F., Ciuhu, C., ten Kate, W., Heinrich, A., Uzunbajakava, N., Krekels, G., Siem, D. and de Haan, G. (2009) Automatic Imaging System with Decision Support for Inspection of Pigmented Skin Lesions and Melanoma Diagnosis. *IEEE Journal of Selected Topics in Signal Processing*, 3, 14-25. <https://doi.org/10.1109/JSTSP.2008.2011156>.
- [2] Marín, C., Alférez, G.H., Córdova, J. and González, V. (2015) Detection of Melanoma through Image Recognition and Artificial Neural Networks. *World Congress on Medical Physics and Biomedical Engineering, Toronto, 7-12 June 2015, Vol. 51*, 832-835. https://doi.org/10.1007/978-3-319-19387-8_204
- [3] Pattem, Sampurnima & Thatavarti, Satish. (2023). Identification of duplicate parts of hyper spectral images based on fuzzy by dimensionality reduction techniques. *Soft Computing*. 1-12. 10.1007/s00500-023-08574-2.
- [4] Shu, Wenhao & Yan, Zhenchao & Yu, Jianhui & Qian, Wenbin. (2022). Information gain-based semi-supervised feature selection for hybrid data. *Applied Intelligence*. 53. 10.1007/s10489-022-03770-3.
- [5] Qiao W, Yang Z (2019) Solving large-scale function optimization problem by using a new metaheuristic algorithm based on quantum dolphin swarm algorithm. *IEEE Access* 7:138972–138989. <https://doi.org/10.1109/ACCESS.2019.2942169>
- [6] Kaul S, Kumar Y (2021) Nature-inspired metaheuristic algorithms for constraint handling: challenges, issues, and research perspective. In: Kulkarni AJ, Mezura-Montes E, Wang Y, Gandomi AH, Krishnasamy G (eds) *Constraint handling in metaheuristics and applications*. Springer, Singapore. https://doi.org/10.1007/978-981-33-6710-4_3
- [7] Bennett R, Logue J, Liu D, Reeder R, Janosko K et al (2020) Kikwit Ebola virus disease progression in the rhesus monkey animal model. *Viruses* 12(7):1–14. <https://doi.org/10.3390/v12070753>
- [8] Hajhashemi V, Hassani Z, Dehmajnoonie I, Borna K (2019) Hybrid algorithms of whale optimization algorithm and k-nearest neighbor to predict the liver disease. *EAI Endorsed Trans Context-Aware Syst Appl* 6(16):156838. <https://doi.org/10.4108/eai.13-7-2018.156838>
- [9] Dermnet. Kaggle. Available online: <https://www.kaggle.com/datasets/shubhamgoel27/dermnet> (accessed on 20 June 2023).
- [10] ISIC Challenge. Available online: <https://challenge.isic-archive.com/landing/2019/> (accessed on 21 June 2023).
- [11] Verma, Anurag & Pal, Saurabh & Tiwari, B. (2020). Skin disease prediction using ensemble methods and a new hybrid feature selection technique. *Iran Journal of Computer Science*. 3. 10.1007/s42044-020-00058-y.
- [12] Verma, Anurag & Pal, Saurabh & Kumar, Surjeet & Tiwari, B. (2019). Comparison of skin disease prediction by feature selection using ensemble data mining techniques. *Informatics in Medicine Unlocked*. 16. 100202. 10.1016/j.imu.2019.100202.
- [13] Abikoye, Oluwakemi & Komolafe, R & Aro, Taye. (2019). Performance evaluation of feature selection algorithms on skin disease prediction. 4. 337-342.
- [14] Ramya, P. & Sathiyabhama, B.. (2023). Skin Cancer Prediction using Enhanced Genetic Algorithm with Extreme Learning Machine. *Journal of Trends in Computer Science and Smart Technology*. 5. 1-13. 10.36548/jtcsst.2023.1.001.
- [15] Sahu, Shreya & Das, Pranesh & Jose a, Binu. (2023). A New Framework for Disease Prediction: Using Dimensionality Reduction and Feature Selection. 10.1007/978-981-19-9858-4_32.
- [16] Baireddy, Ravinder & Nagaraja, R. (2022). A hybrid particle swarm optimization with multi-objective clustering for dermatologic diseases diagnosis. *Journal of Intelligent Systems*. 31. 876-890. 10.1515/jisys-2022-0028.
- [17] Mirjalili, Seyedali. (2015). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*. 27. 10.1007/s00521-015-1920-1.