International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.13, No.1 (Apr-23)

http://dx.doi.org/10.12785/ijcds/130152

Performance Improvement of Convolutional Neural Network Architectures for Skin Disease Detection

Hasan Maher Ahmed[1] and Manar Younis Kashmola[2]

[1]*Software Department,College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*
[2]*Computer Science Department, College of IT on leave from Mosul University, Ninevah University, Mosul, Iraq*

Received 26 Dec. 2021, Revised 22 Dec. 2022, Accepted 6 Feb. 2023, Published 16 Apr. 2023

Abstract: Convolutional neural networks are one of the most important techniques used in classification processes, specifically digital image classification. In this paper, work has been done to apply a set of different deep CNN architectures in classifying skin disease images. The performance of those deep networks or training methods was also improved which improved the image classification result. The main objective of this research paper is to improve the performance of convolutional neural network architectures for the detection of skin diseases, as the data set of images of skin diseases was adopted from the International Collaboration for Skin Imaging (ISIC) 2020, where the number of images that were used 5224 digital images of five skin diseases included 1327 Nevus pictures, 1098 pictures of basal cell carcinomas, 1099 pictures of pigmented benign keratoses, 1046 pictures of seborrheic keratoses, and 654 pictures of squamous cell carcinomas. The performance of AlexNet, ZfNet, VGG16, and VGG19 deep networks has been improved by generating new seed weights for each network based on Artificial Bee Algorithm, Bat Algorithm, Gray Wolf Optimization, Bacterial Foraging Optimization, and Particle Swarm Optimization. After obtaining the results from the improved architectures, it was found that the performance accuracy increased significantly, and the architectures gave clear stability in training the deep network data set.

Keywords: Deep Learning, Convolutional Neural Network, Swarm Optimization, Skin Diseases, Image Classification

1. INTRODUCTION

One of the most recent advances in machine learning and artificial intelligence research is deep learning. It's also one of today's most popular scientific research topics. Deep learning techniques have ushered in a new era of computer vision and machine learning[1]. New deep learning technologies emerge regularly, often circumventing even the most advanced machine learning algorithms and existing deep learning techniques. The world has made some significant advancements in this subject in recent years. Deep learning's rapid evolution makes it tough to keep up with, especially for novice researchers[2].
Multilayer networks have shown success due to their ability to exploit the synthetic structure of natural data. The set of elements in one layer forms a new element in the next layer with hierarchical combinations. If we simulate this hierarchy as a set of layers and leave the network with the task of extracting and learning the appropriate properties for it, we have created what is known as a deep learning model. Hence, it can be said that deep learning networks are hierarchical networks[3]. Deep learning architectures have been the reason for unprecedented progress in computer vision applications, from recognizing and accurately locating elements in images, to determining spatial properties of the element to be recognized in the image.
Characterization extraction involves transferring them to higher dimensions, as they can usually be linearly separated in those higher dimensions due to the increased number of levels of separation possible in higher dimensions. CNNs were designed for image recognition tasks. The main design aim of CNNs was to develop a network in which neurons in the early levels extracted local visual characteristics, which were then combined by neurons in later layers to form higher-order features. Many different CNN architectures have been created over the years to meet real-world challenges[4].
CNNs are commonly used in computer vision applications and visual scene analysis and are considered a solution to many artificial intelligence problems such as image and video processing, image analysis, video and image recognition and classification, Medical color video and image analysis, language processing, and natural dialects as in Figure 1. CNNs use a type of deep neural network that can recognize and classify certain features in images and are usually good in networks with more than one layer that are used to extract properties from input data (e.g. digital images or digital videos) while the task of the layer The fully connected is to produce the desired output.[5]
The term "convolution" in CNN refers to the mathematical function of convolution in which a filter (also known as

E-mail address: hasanmaher@uomosul.edu.iq, Manar.kashmola@uomosul.edu.iq

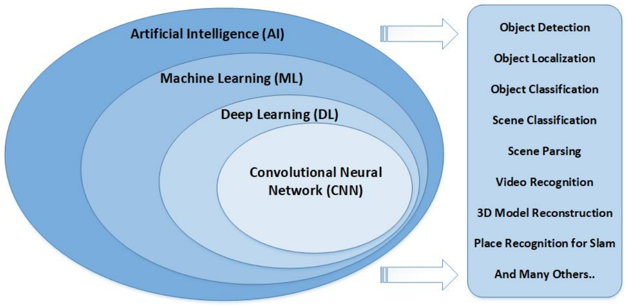

Figure 1. The main areas of use of convolutional neural networks

a kernel) is applied in a convolution layer whose job is to extract properties from the input data and several filters can be used to extract different attributes. The filter is tiny enough to scan the entire image and extract the features using proper arithmetic between the filter values and the pixels. The filter settings are reset during the periodic training phase, and when the network has been trained for a particular number of epochs (epochs imply all training samples have been entered at the same time), these filters start looking for different characteristics in the image. Simple and evident features, such as edges in various directions, are extracted using the first hidden layers. The complexity of the attributes that must be recognized and extracted grows as we go deeper into the network's hidden levels[6].

There are two aspects to the CNN architecture: a fully connected layer that uses the output from the convolution process and predicts an image class based on the features extracted in earlier stages, and a convolution tool (convolution is an efficient feature extraction method) that separates and selects the various features of the image for analysis in a process known as feature extraction[7].

Recently, Convolutional Neural Networks (CNNs) have become very important and their outstanding performance in the field of image classification has attracted wide attention. In deep learning, there are many different deep network structures. A convolutional neural network is a machine learning algorithm that can provide the best accuracy in many use cases. However, in many cases, the accuracy of the neural network we build may not be satisfactory, or it may not allow us to get ahead in the classification process. Therefore, we are always looking for better ways to improve the performance of the model[8].

Convolutional neural networks (CNNs) are neural networks that have at least one convolutional layer. After entering an image, a succession of convolutional layers, non-linear activation layers, pooling layers, and fully connected layers, the matching class labels can be produced in a conventional CNN architecture. The entire image is passed to the network to train the weights of each network layer. When the input is a simple image, the network recognition effect will be better, but when the input becomes a more complex and mutable image, the network recognition effect will be more difficult. Adding more layers to know the abstract features of the input image may help, but this will increase the number of neurons, greatly increase the computing resources required for training, and take up a lot of memory[9].

The CNN target recognition process is to first find relevant low-level features such as edges, line segments, and curves and then use multiple convolutional layers to build more abstract high-level features. The convolutional layer's learning procedure involves exchanging the weights of several convolution kernels (or feature detectors) to learn the local information for each image, which is then used to generate an abstract feature map. Bypass kernel sharing reduces the number of parameters needed to train the network greatly. Since trained detectors can frequently be used to detect abstract features in images through convolutional layers, convolutional neural networks are more suitable for complex image recognition tasks[1].

The main benefit of filters and receptive fields is to identify important features in the input images. The feature map or the activation map determines the locations of the features in a given image, which is a matrix resulting from multiplying each of the filter matrix and all matrices of receptive domains in an element-wise product manner, i.e. multiplying the similar elements in terms of location in the two matrices so that The first element of the filter matrix is multiplied by the first element of the receptive domain matrix, the second element of the filter matrix by the second element of the receptive domain matrix, and so on until the last element in each of the two matrices is reached, then summing all the products and storing them in one value (the output is one value Only in the feature map and to complete the map you have to go through all the receiving fields)[10].

Filters in Convolutional Neural Network layers extract feature maps by detecting low-level features like edges and curves to forecast if an image belongs to one of the classification types. To recognize higher-level characteristics such as objects in an image, convolutional neural networks must send the output via another convolutional layer, where the first layer's output (corrected feature maps) becomes the input of the second convolutional layer. The inputs of the first layer are only the original image, while the inputs of the second layer will be the feature maps that result from the first layer, so the output will be activations representing higher-level features. These features types can be a combination of curve and straight edge or a combination of several straight edges and during Network Navigation and Browse More Convolutional Layers Feature maps representing more and more complex features will be obtained, and at the end of the network][11].

2. RELATED WORK

Some of the recent research work conducted to diagnose and classify various skin diseases through image processing, data mining, and deep learning techniques are presented. WU et al. presented a study of different CNN algorithms for classifying facial skin diseases based on clinical images. Studies using five major disease classification algorithms were presented in the dataset. Studies were then conducted using an independent dataset of the same types of diseases but from other parts of the body[12].

Mohammed and colleagues also presented a study in which they developed a skin identification method that uses a scale to assess the distances between pixels and skin tones to distinguish between the skin and non-dermal pixels in color still photos. The YCbCr color scheme, which isolates the overall image energy in the luminous range, resulted in superior skin pixel recognition than regular red-green-blue images, according to the researchers. Backgrounds or wood items in a poorly categorized RGB color space. Then, to discriminate between cutaneous and non-cutaneous pixels, a histogram-based image segmentation plot was applied. The demand for compact skin model representation should spur the development of skin detection parametric models[13].

Jagdish et al. also presented a paper in which the advanced study of the detection of skin diseases using image processing methods was considered. Some diseases have a genetic or some situational cause. A dermatological study was previously tested using fuzzy sets using KNN and SVM machine learning methods with wavelet analysis with 50 sample images. The results represent that the K-Nearest Neighbor classification algorithm performs well compared to the Support Vector Machine (SVM) classification technology. The algorithm also identifies the type of skin disease using classification methods[14].

Srinivasu et al. recommended using MobileNet V2 deep learning and long-term memory to conduct a computerized process study of dermatological classification (LSTM). The gray-level occurrence matrix has been used to assess disease progression and compare performance with other recent models developed by the Visual Geometry Group (VGG) for large-scale image recognition, such as Micro Neural Networks (FTNN), Convolutional Neural Networks (CNN). In addition, the HAM10000 dataset was employed[15].

Yao et al. also proposed a new data augmentation strategy to classify a single skin lesion model based on a small, unbalanced data set where several DCNNs were trained on the input data set from which compatible models are selected. DropOut and DropBlock have also been added to maintain stability, and a strategy has been introduced to work with RandAugment to preserve components of the input data and address data loss issues. The problem of classification difficulty was also overcome by applying the multi-weighted focal loss functions and combining them with the proposed RandAugment, as the results gave a classification accuracy similar to that of the multiple aggregation models in the ISIC 2018 Challenge Test data set[16].

3. PROPOSED METHODOLOGY

A. *Deep Convolutional Neural Network*

In several computer visual recognition applications, convolutional neural networks (CNN) have been used., as they have been characterized by good performance and high quality. Since image classification processes are important in visual object recognition and semantic segmentation, standard CNN architectures must be developed from the image classification network architecture. Optimizations (improving the educational and representative qualities of deep networks) can be done by reformulating the connections between network layers[17].

Convolutional operations in convolution layers in CNN architectures allow the processing of inputs of variable size depending on the kernel to collect and retrieve distinct input features. The main properties of the convolution layer are sparse interactions and weight sharing. Lower-level features such as corners, edges, and endpoints are collected in the first layer as in Figure 2. The upper layer then collects and extracts more sophisticated and higher-level characteristics by handling lower-level qualities[13].

The operations of communication between inputs and outputs are formed when the matrix is multiplied in traditional neural networks, as each input unit is associated with a corresponding output unit, and thus this correlation will increase the amount of computation when the input image contains a large number of pixels and this will affect the storage requirements[16].

One of the most important features of convolution layers in deep networks is the weight-sharing operations, as they are adopted to accommodate inputs of variable sizes, by relying on convolution kernels to limit the number of parameters. It can be noted that sharing weights means using the same weights and deviations in all units in the class. The first layer of LeNet-5 is a convolution layer consisting of six convolutions, each with its own weights and values. so the first layer will contain a total of (6355 + 6) = approximately 156 parameters as in Figure 3. One of the most important advantages of weight sharing is that it reduces network training parameters to a higher extent than the entire linked network structure, it significantly decreases the increase in network processing caused by a high number of parameters and so increases network operating efficiency[18].

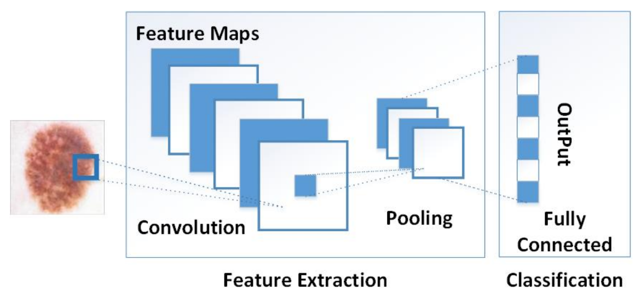

Figure 2. Feature Extraction in Convolutional Neural Networks

Sub pooling (also called aggregation) consists of two models namely aggregation rate and maximum aggregation. Subsampling is a special convolution operation, so convolution and subtraction are a good way to greatly simplify the complexity of the model, reduce both the model parameters, the spatial extent of the data, and the number of parameters in the network, and reduce the number of computational resources expended. The feature map formed after the convolution operation is usually the input data for the aggregation layer[14].

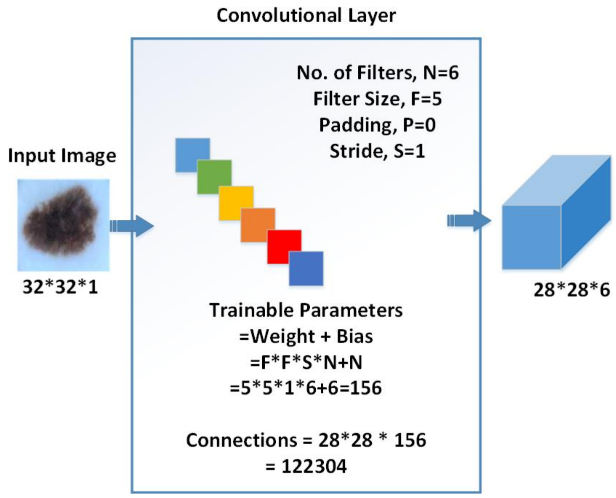

Figure 3. Arithmetic operations in the convolutional layer

The retrieval of the feature map (such as digital image data) is then done after a chain of operations from the convolution and aggregation layers and then transforms all the resulting neurons into a fully linked layer which binds to the softmax layer to classify the output and improve the overall performance of the CNN depending on the local information and the stratigraphic differentiation in both Wrapping and grouping layers[19].

The AlexNet design demonstrates improved performance. Scattering can activate neurons selectively or in a general fashion, since ReLU may induce both nonlinearity and scattering into the network. To artificially augment the dataset, AlexNet employs label-preserving alterations as in Figure 4. Image translations, horizontal reflections, and RGB channel intensities in training images are all examples of data augmentation[15]. To lower network model parameters and avoid overfitting, neurons can be deleted from the network with a given probability. The matching edge of the pooling kernel is less than the pooling step size[19].

Researchers turned their attention to CNN after AlexNet

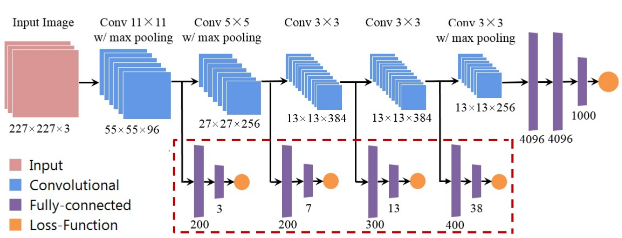

Figure 4. AlexNet Architecture

won the ImageNet image categorization competition. ZFNet was introduced, which used deconvolution to improve performance. A shallow network detects an image's edge, color, and texture features, but a high-level network recognizes the image's abstract features, as shown by feature maps as in Figure 5. There are hierarchies between features. The feature's stability and discriminatory ability both improve as the level rises. According to various occlusion studies, the model is strongly tied to local factors in classification[20].

In large-scale image identification tasks, the shallow neural

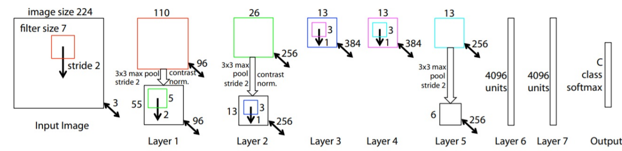

Figure 5. ZfNet Architecture

network approach has some limitations. To test the ability of the VGG deeper network model to fully analyze networks of increasing depth using a system with relatively small (3x3) convolutional filters., demonstrating that expanding the depth to 16 to 19 weight layers results in a considerable improvement. Unlike AlexNet and ZFNet, VGG develops its network using a small (3x3) convolution kernel. It is important to maintain the computational cost of the convolutional neural network architecture for each layer by reducing the size of the feature map depending on the pooling layer. The receptive field of two (3x3) convolutions is the same as a (5x5) convolution, while the receptive field of three (3x3) convolutions is the same as a (7x7) convolution. The network used three (3x3) convolutions rather than a (7x7) convolution for two reasons: The decision function is more selective because it has three ReLU layers rather than one; second, the number of parameters can be lowered as in Figure 6[20].

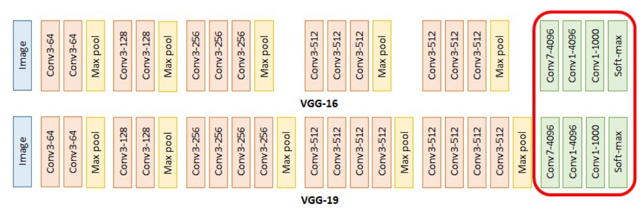

Figure 6. VGG16 and VGG19 Architectures

B. Dataset

The data came from the ISIC 2020 archives, which stands for the International Skin Imaging Collaboration. ISIC develops suggested standards for dealing with skin imaging methodologies, tools, and formulations, with a focus on security and interoperability (ie, the ability to share images across technology and clinical systems). In addition, to test and evaluate suggested standards, ISIC has built and expanded the Open Source Community Proceedings (ISIC Archive) for skin pictures. Various images of skin infections can be found in the ISIC archive as in Figure 7. The data set was constructed for five skin diseases, including 1,098 images of basal cell carcinomas, 1,327 images of nevus, 1,099 images of benign pigmented keratoses, 1046 images of seborrheic keratoses, and 654 squamous cell carcinomas, as in figure 7[21].

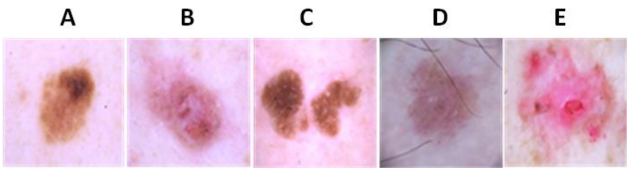

Figure 7. Samples of Training Images: A: Nevus, B: Basal Cell Carcinoma, C: Pigmented Benign Keratosis, D: Seborrheic Keratosis, E: Squamous Cell Carcinoma

C. Data Augmentation

Machine learning applications, particularly in the deep learning domain, are rapidly diversifying and expanding. Techniques for data augmentation could be useful in combating the issues that the artificial intelligence sector faces [10]. To improve the performance and results of machine learning models, the data can be increased by creating additional and various cases for training data sets because a large and sufficient data set leads to the formation of a better and more accurate machine learning model. For machine learning models, data gathering and labeling can be time-consuming and costly[22]. Data augmentation strategies can be used to lower operational expenses by modifying datasets as in Figure 8.

Data augmentation is one of the important stages of creating a good data model to obtain high-accuracy classifications[11]. At the same time, if data augmentation decreases classifiability, it means that the models will not be able to produce accurate predictions for real-world inputs, and data augmentation processes and procedures produce Variations that real-world models can match-make machine learning models more robust[1].

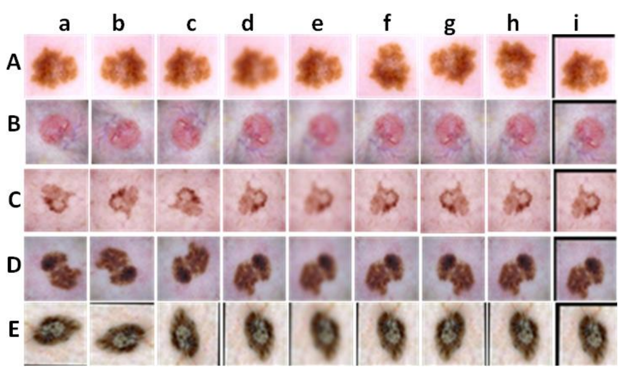

Figure 8. Some Samples of Data Augmentation: A: Nevus, B: Basal Cell Carcinoma, C: Pigmented Benign Keratosis, D: Seborrheic Keratosis, E: Squamous Cell Carcinoma, a: Original Image, b: Flipping, c: Cropping, d: 2-D Gaussian Filtering, e: Resizing, f: Rotation by 90, g: Rotation by 180 h: Rotation by 270, i: Translation

D. Deep Augmentation

Machine learning applications, particularly in the field of deep learning, are rapidly diversifying and expanding. Approaches to data augmentation can be a useful tool for coping with the problems that artificial intelligence encounters. By supplying additional and different cases for training data sets[23][24].

Data gathering and categorization can be time-consuming and costly processes for machine learning models. Data augmentation strategies can be used to reduce operational expenses when data sets change. In this paper, an architecture for a generative adversarial network (GAN) was designed and built[19]. The proposed architecture consists of a generating network and a discriminant network. The generative network generates new data through many transformations while the role of the discriminative network is to distinguish between the real data and the fabricated data produced by the generative network[25]. The proposed architecture for producing 128 x 128 images are shown in Figure 9. Machine learning models can be made more powerful by using data augmentation approaches to create alternative shapes that the model would encounter in the real world[22].

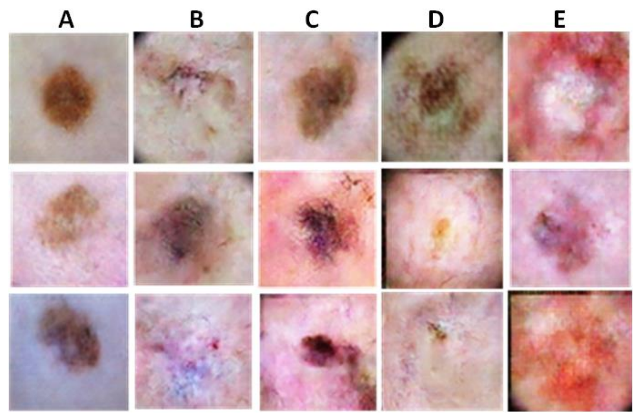

Figure 9. Samples of Deep Augmentation images: A: Nevus, B: Basal Cell Carcinoma, C: Pigmented Benign Keratosis, D: Seborrheic Keratosis, E: Squamous Cell Carcinoma

E. Weight Optimization

Convolutional neural networks have in the receptive fields the filter, as this network trains itself iteratively to choose random values for the filter called weights or parameters, and passes them on all the input images so that the convolutional network can finally extract simple repetitive patterns in the images through which it is determined Certain features of the target group such as geometric shapes, edges, and curves, etc., and finally after modifying these weights several times are fixed and set mainly as values for filters based on the importance and priority of the detected feature[26]. The output value of each neuron in the neural network is calculated from the output of the previous layer and the vector weights and biases define the basic functions of the function (usually the real numbers)[27]. Learning is the process of repeatedly changing these biases and weights. Filters are made up of weights and bias vectors that represent different aspects of the input (for example, a certain shape). A distinguishing feature of CNNs is that several neurons can share the same filter[20].

A CNN architecture is made up of a series of discrete layers that use a differentiable function to turn an input size into an output size (such as holding class degrees). There are a few different sorts of layers that are regularly utilized[28]. Weight decay is a straightforward additive regularizer that simply adds error at each node proportionate to the sum of weights or the weight vector's squared magnitude[26]. The weight vectors are sparse during optimization due to the sum of weight regularization. In other words, regularized neurons use a small percentage of their most important inputs and are almost impervious to noise. The most prevalent type of regularization is squared magnitude regularization. It is possible to achieve this by directly squaring the parameters. Peak weight vectors are punished harshly, while scattered weight vectors are encouraged according to the intuitive interpretation, so the network will be instructed to use all its inputs and for all layers briefly instead of using some of them repeatedly, because weights and inputs have multiple interactions[11].
CNNs are fully connected, multi-layered perception processors. Their extensive connectedness renders them sensitive to data overload, which can result in weight loss, disconnection, and other issues. CNNs use the data's hierarchical structure to aggregate increasingly complex patterns utilizing smaller, simpler patterns stored in their filters[29].
Each neuron in the output neuron is calculated by a function defined on the introduction of the receptive solutions in the previous solutions. They are voted on by these processes[14]. Learning progresses in a convolutional neural network by repeatedly performing these analyzes and weights as in Figure 10. The size of the feature map changes with the change in the depth of the convolutional neural network, so the feature map decreases with the direction of the depth of the network, and the layers close to the input will have fewer filters. Each layer must have arithmetic balancing by preserving the output of feature values, and the input information must be preserved and activations stored (feature maps * pixels) with no decrement between layers. The number of feature maps is very important in influencing capacity and depends on the complexity of the tasks and the number of examples available[30].
A convolutional neural network consists of multiple hidden

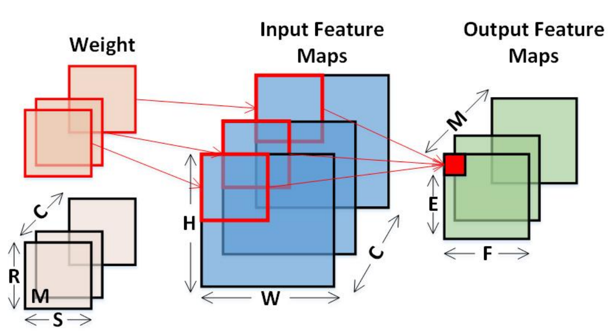

Figure 10. The interaction of weights with the features of the entered images

layers and also contains the RELU activation function and then additional convolutions such as flattening layers. The feature map is created when a convolution kernel is passed along a layer's input array, which subsequently contributes to the next layer's input[31].
The output is sent to the next layer of each convolutional neuron after which the data is analyzed in the receptive domain[**?**]. Although features are learned and the data is categorized, larger inputs such as high-resolution images make this structure impractical. Even with a shallow structure, the massive input volume of the images requires a large number of neurons (each pixel reflects an important input feature)[27]. The importance of the convolution process is to reduce the number of parameters in the convolutional neural network and this will lead to the formation of a deeper network. As a 5 x 5 tile area generates the same weights for 25 parameters to be used for learning. It is important to use regular weights to generate fewer parameters which will prevent gradients from disappearing and amplify the gradient difficulties that occur during mesh backpropagation[32]. The most important characteristic of convolutional neural networks is that they are fully compatible with data with similar structures (e.g. images) because the spatial relationships between discrete properties are dealt with concentrically during convolution and/or aggregation[33].
Swarm intelligence refers to the collective behavior of decentralized, self-organized systems, whether natural or artificial (SI). Artificial intelligence research makes use of this concept. SI systems are frequently made up of a population of simple agents that interact on a local level with one another and their surroundings. Nature, especially biological systems, is a constant source of inspiration for me. There is no central formula for determining the behavior of individuals, but random local interactions determine the intelligent global behavior of individuals. Intelligent optimization methods are related to natural systems such as flocks and colonies of terrestrial and marine animals[12].
Swarm concepts can be applied to robots, becoming swarm robots, while swarm intelligence is applied to various algorithms. In the area of forecasting difficulties, swarm prediction has been applied. In synthetic collective intelligence, genetically engineered organisms are being studied for approaches similar to those proposed for swarm robots. Particle Swarm Optimization A PSO is a population whose behavior depends mainly on statistical example techniques. An assembly in turn is a set of elements that represents a set of solutions and each element is a particle[34]. This model is inspired by the social behavior of flocks of birds or groups of fish as they move from one place to another[35]. The goal of this algorithm is to obtain the optimal and best solution and result - by simulating the behaviors of birds in search of the best food - and therefore any system that depends on this algorithm will initially be formed from a random pool of random solutions and within this pool is searched for the optimal solution through Generations update[36].
The Bacterial Foraging Optimization was inspired by Escherichia coli's social foraging behavior (next E.coli).

Chemotaxis is a process in which bacteria travel towards a nutrition gradient while avoiding unpleasant environments. In a friendly environment, germs will typically travel a greater distance[36].
Chemotaxis is a technique for imitating the movement of an E.coli cell by swimming and tumbling via flagella. Biologically, E.coli bacteria can migrate in two ways. It can swim in one direction for an extended period or tumble, and it can transition between the two modes of functioning at any time during its life. Reproduction: The bacteria with the lowest objective function values eventually die, or bacteria with higher values are placed in the same position with the process of splitting into two parts to maintain the stability of the size of the swarm. Some changes in the environment of bacteria occur slowly or quickly, which may change to a change in temperature, which leads to the loss of a group of bacteria or the spread of a group of bacteria in a new place. These processes are repeated in a BFO where certain bacteria are randomly eliminated with a very small chance and new substitutions are randomly introduced through the search area.
Gray wolves in nature have a leadership structure and a hunting process that the Gray Wolf Optimization algorithm resembles. Wolves live in packs and are gregarious animals[37]. There are four tiers of wolf packs: alpha, beta, delta, and omega. A wolf swarm is a strict social hierarchical swarm divided into alpha, beta, delta, and omega. The alpha group leads the herd and their task is to make the appropriate decisions. Beta helps the alpha make choices so that if the alpha dies, the best individual in the beta becomes the alpha, and the rest of the herd works to provide feedback to the alpha[38].
Despite having to kneel to alphas and betas, delta wolves control the omega. In the end, Omega wolves must obey all other wolves. They take up the job of caregivers on occasion. The social order of wolves has linked to the solution fit in a mathematical model. The alpha solution is the one that is the most suitable. The second and third best answers are beta and delta, respectively. Omega is assumed for the rest of the possible solutions. The hunting (optimization) is led by alpha, beta, and delta wolves, with omega wolves trailing after.
The bat algorithm works on the principle of echolocation ability to navigate. The bat makes a loud noise and listens for echoes from sound reflections from the environment. The main use of echo is to estimate the distance between food and an obstacle[39]. While the loudness can be changed in several imposter ways, the number is hypothesized between a large positive number and a small constant number. Let's utilize the following approximations in addition to these simplified principles: The wavelength segment [min, max] corresponds to the frequency f from the section [fmin, fmax]. For example, a wavelength section [0.7 mm, 17 mm] corresponds to a frequency segment [20 kHz, 500 KHz]. Any wavelength can be used for this taste. Furthermore, using wavelengths is unnecessary; The frequency can be adjusted while the wavelength is fixed[40].
Artificial bee algorithms are optimized algorithms adopted in artificial intelligence techniques and the main task of a swarm of bees is to find flowers densely in a wide field and the search is initially for flowers from random locations with random speed vectors. In the Artificial Bee Algorithm model, there are three categories of bees in the colony: employed, bystanders, and scouts. Scouts seek at random, hired bees to collect food that has already been identified, Monitors choose food sources based on observations of worker bees. Dances are used by bees to communicate with one another. A bee watches the dances of other bees before starting to collect food. Bees use a dance to communicate where food might be found[41].
Bees, both working and non-working, search the area around their colony for plentiful food sources. A working bee collects and transmits information about a food source to passers-by[42]. Scout bees cannot improve solutions even if multiple experiments are done and do not use the solutions again. The population's number of solutions is represented by the number of food sources[42]. The amount of nectar that has been identified indicates the extent of the number of food sources discovered and thus determines the quality and fitness of possible solutions and thus possible solutions to the problem of improvement can be reached, Figure 11 shows the general scheme of hybridization of algorithms.

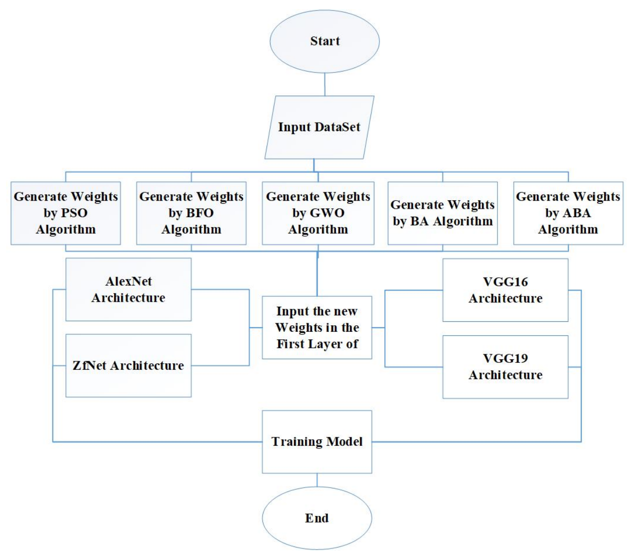

Figure 11. General scheme of hybridization of algorithms

4. RESULTS AND DISCUSSION

The processes of classifying digital images based on artificial intelligence differ from one method to another in terms of classification accuracy. This section presents the digital image datasets used for this approach shown in table I, the performance results of the applied algorithms before and after optimizations on them, and the results of all the datasets used for classifying the five dermatological diseases.

TABLE I. Digital image data sets used for classification

Skin Disease	No. of Training Images	Data Aug. Images	Deep Aug. Images	No. of Testing Images
Nevus	1327	11943	1000	2580
BCC	1098	9882	1000	2170
PBK	1099	9891	1000	2170
SK	1046	9414	1000	2080
SCC	654	5886	1000	1370
Total	5224	47016	5000	10370

First, the AlexNet algorithm was applied to the data set, and Figure 12 shows the accuracy obtained from applying this algorithm to the digital image data for the classification of the five dermatological diseases, and the highest accuracy was 76.44% for 150 epochs.

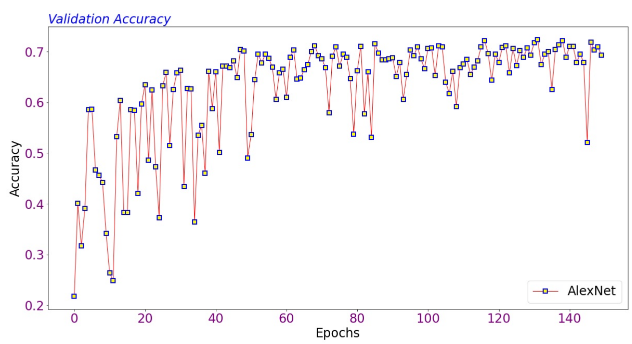

Figure 12. Accuracy of algorithm classification before optimization using AlexNet architecture

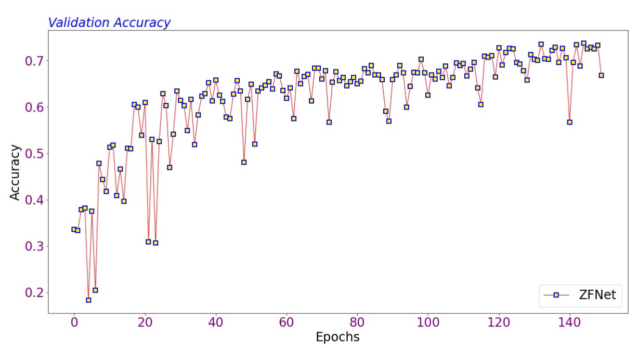

Figure 13. Accuracy of algorithm classification before optimization using ZfNet architecture

Then the ZFNet algorithm was applied to the same data set, where the highest accuracy of 73.75% was obtained, as shown in Figure 13.
Then the VGG16 algorithm was applied to the same data set, where the highest accuracy of 59.58% was obtained, as shown in Figure 14. Then the VGG19 algorithm was applied to the same data set, where the highest accuracy of 68.49% was obtained, as shown in Figure 15.

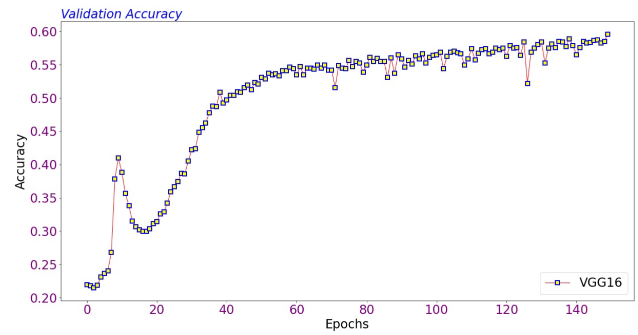

Figure 14. Accuracy of algorithm classification before optimization using VGG16 architecture

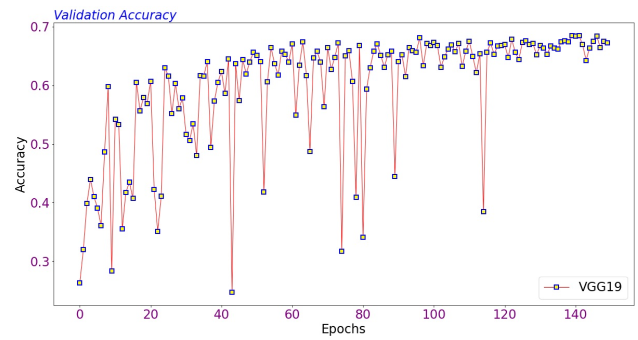

Figure 15. Accuracy of algorithm classification before optimization using VGG16 architecture

AlexNet weights were generated using intelligent swarm algorithms. First, new weights for the convolutional neural network were created using the swarm bat's approach, in which the bat utilizes echolocation to measure distance and ”knows” To determine echolocation and show the difference between food and an obstacle. Algorithm 1 shows the bat swarm algorithm.

Algorithm 1: Bat Swarm Algorithm
Begin
f(y), y= (y1, ..., yd) T
bat population y_j $(j = 1, 2, ..., n)$**,** x_j
while max number $>$ **count**
Adjust the frequency and update the velocities and locations/solutions to generate fresh solutions
IF random $> R_j$
Choose a solution from the list of the best options
Create a local solution based on the best option that has been chosen
end if
make a new solution
if random $< A_j$ **AND** $f(y_j) <$ f(xx)**THEN**
Accept, and Increasing Rj while lowering Ai
end if
determine who is currently the best xx
end while
Results and visualization after post-processing
END

where: R_0: amplitude of the impulse, x_0: the sound volume, fmin: frequency of the minimum wave, fmax: frequency of the maximum wave, and alpha: constant for changing the sound volume
To get to the food, bats set some parameters: random speed (vi), direction (xi), frequency (fmin), and wavelength. Based on the victim's proximity, they can change the wavelength (or frequency) of the generated sound impulse as well as the amount of emission r [0, 1].
The most important advantage of convolutional neural network algorithms is that they generate weights for all network inputs, and the values of the weights change from layer to layer. To improve the performance and classification accuracy of convolutional neural networks. New weights were also generated based on the bee swarm algorithm. The artificial bee algorithm model has three types of bees in the colony: worker bees, spectators, and scouts. Scouts search at random, worker bees collect already discovered food, and spectators watch worker bee dances and select food sources based on the dances. Non-working bees are spectators and scouts. The food sources in the community represent the number of solutions. Near their hive, bees hunt for abundant food sources. The worker bee collects and conveys information about the food supply to passers-by. Several experiments are applied by worker bees. If the solutions are improved, the improvement processes continue, otherwise the worker bees are transferred to scouts and all previous solutions are canceled. Algorithm 2 shows the Artificial Bee Algorithm.

Algorithm 2: Artificial Bee Algorithm
Begin
make a population start
For the first iteration, find the best agent currently available
determine how many scouts, bystanders, and employed bees there are
change the global best (G) to the current best (C)
Loop iteration = 0
assess each agent's fitness
sort fitness in ascending order to find the most qualified agents
agents from A to C should be chosen from the list of finest agents
create fresh bees that will swarm to the most suitable solution
Evaluate C agent
If G > C
G = C
End If
End Loop
Where only standard arguments are accepted by the aba method
End For
End

The bacteria swarm method, which simulates the movement of the Escherichia coli cell through swimming and landing through the flagella, was also used to generate new weight values. E. coli bacteria travel in different ways where they can travel in the same direction for a long time or they may get stuck. Automatically, the bacteria are of two types, as the first type is the least healthy bacteria and eventually will die, and the second type is the healthiest bacteria where they multiply by division and are planted in the same place and in this way the size of the swarm will be maintained. Several factors affect the survival or decay of bacteria, and a group of bacteria may live gradually or directly. The high temperature has a significant effect in killing a group of bacteria present in a high concentration area, or it is also possible to kill all bacteria in the area or spread them to a new place. In BFO, some bacteria are randomly eliminated, which are infrequent, in the assays while new substitutions are randomly generated across the search space to repeat this process. Algorithm 3 shows the Bacterial Foraging Optimization.

Algorithm 3: Bacterial Foraging Optimization
Begin
create a population of bacteria for foraging optimization at random
calculate each agent's fitness
change the best agent for the entire world to the best agent for the entire world
loop the total number of iterations
loop how many chemotactic steps are there?
loop every search engine
change the agent's direction at random
Calculate the moved agent's fitness
loop swimming duration
If current fitness level is higher than your prior level
the agent should be moved in the same direction
else
the agent should be moved in a random direction
End If
End Loop
End Loop
determine each agent's fitness
calculate and sort the total of all chemotactic loops' fitness functions
allow only half of the population to live and divide them according to their health
If it isn't the most recent iteration
Loop every search engine
replace agent with a new randomly generated one if possible
End Loop
End If
End

As the mathematical model of the wolves' social hierarchy transferred to the suitable answer, new weights were generated based on the gray wolf's swarm algorithm. Alpha is the best option. The second and third best answers, respectively, are beta and delta. Omega is meant to be the rest of the possible solutions. Algorithm 4 shows the Gray Wolf Optimization.

Algorithm 4: Gray Wolf Optimization
Begin
create a gray wolf population at random
find the best agents (A, B, C) in the first, second, and third places
global best (G) agent = 1st best agent
calculate fitness
while max > i iteration
loop search agent
Update the current search agent's position
end loop
update α, β and δ
calculate the search agents' fitness
Update 1st, 2nd, and 3rd best agent
i = i+1
end while
End

As the factors are particles in the parameter space of the optimization job, new weights values were generated using the Particle Swarm Optimization technique. Every particle has a position and velocity vector in every repetition. The value of the objective function for each position of the particle is determined, and the particle adjusts its position and velocity according to particular rules based on that value. Algorithm 5 shows the Particle Swarm Optimization.

Algorithm 5: Particle Swarm Optimization
Begin
agents should be started
find the most recent finest
global best (G) = current best (C)
While iteration > 0
particle velocity should be calculated
change the velocity of the particles
particle positions should be updated
choose new agents based on the selection approach
If the C is better than the G
change the C to the G
End If
End While
End

After applying the algorithms to generate the values of weights, higher accuracy results were obtained with stability in the performance of the work of the network, as shown in Figure 16, Figure 17, Figure 18, Figure 19, and Figure 20.

table II shows the highest classification accuracy achieved after hybridization between convolutional neural network architectures and swarms algorithms. table III, table IV, table V, table VI, table VII and table VIII shows the results of applying the confusion matrix to all architectures.

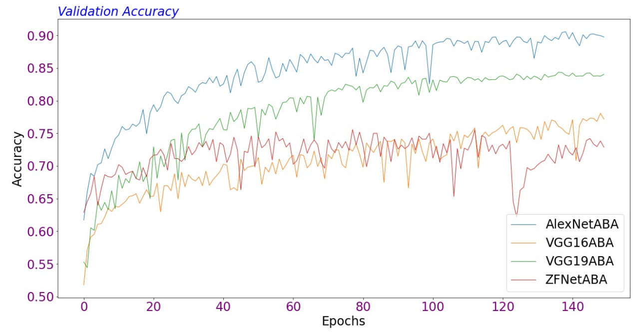

Figure 16. CNNs architectures classification accuracy after hybridization with the ABA algorithm

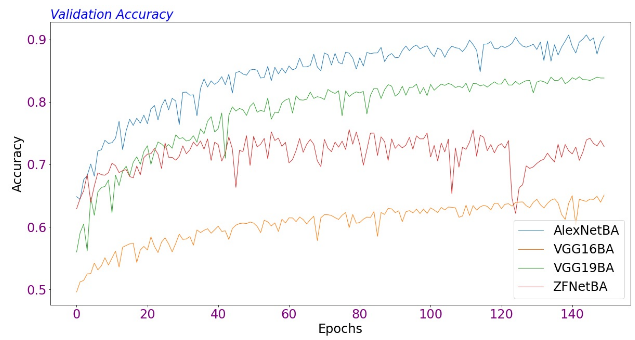

Figure 17. CNNs architectures classification accuracy after hybridization with the BA algorithm

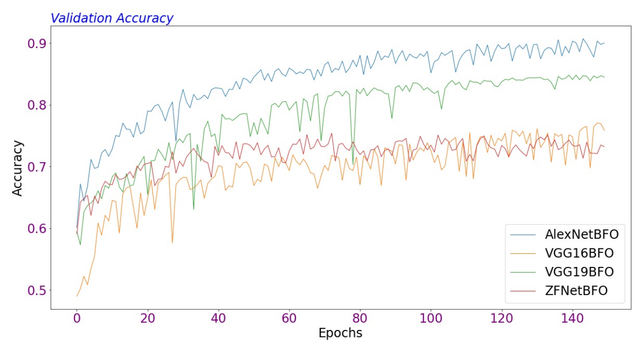

Figure 18. CNNs architectures classification accuracy after hybridization with the BFO algorithm

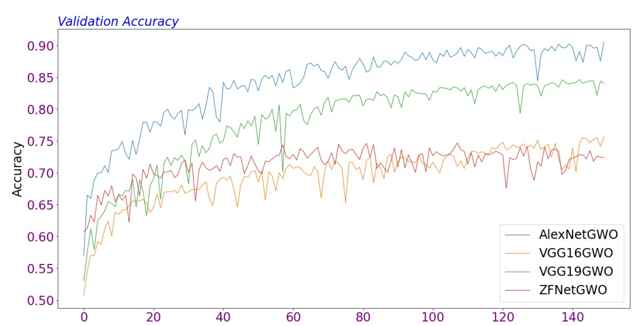

Figure 19. CNNs architectures classification accuracy after hybridization with the GWO algorithm

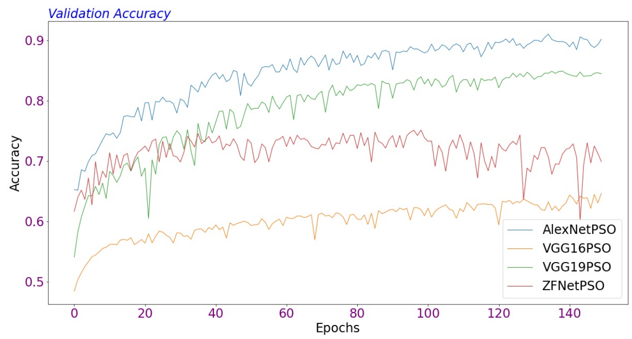

Figure 20. CNNs architectures classification accuracy after hybridization with the PSO algorithm

TABLE II. Highest classification accuracy for hybrid architectures

Net	Org.	PSO	BFO	GWO	BA	ABA
AlexNet	76.44	91.02	90.69	90.60	90.48	90.76
ZfNet	73.75	74.30	74.76	74.13	76.33	74.20
VGG16	59.58	64.63	76.92	75.68	65.10	78.06
VGG19	68.49	84.89	84.79	84.66	83.98	84.31

TABLE III. Correct and incorrect classification results from applying the confusion matrix Before Enhancing

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	1853	1516	1487	1429	859
	F	586	479	470	452	272
ZfNet	T	1780	1456	1428	1373	825
	F	659	539	529	508	306
VGG16	T	1439	1177	1154	1109	667
	F	1000	818	803	772	464
VGG19	T	1658	1356	1330	1279	769
	F	1025	838	822	791	476

TABLE IV. Correct and incorrect classification results from applying the confusion matrix After Enhancing by ABA algorithm

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	2195	1795	1761	1692	1017
	F	244	200	196	189	114
ZfNet	T	1804	1476	1448	1391	836
	F	635	519	509	490	295
VGG1	T	1902	1556	1526	1467	882
	F	537	439	431	414	249
VGG19	T	2048	1675	1643	1580	950
	F	391	320	314	301	181

TABLE V. Correct and incorrect classification results from applying the confusion matrix After Enhancing by BA algorithm

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	2195	1795	1761	1692	1017
	F	244	200	196	189	114
ZfNet	T	1853	1516	1487	1429	859
	F	586	479	470	452	272
VGG16	T	1585	1296	1272	1222	735
	F	854	699	685	659	396
VGG19	T	2024	1655	1624	1561	938
	F	415	340	333	320	193

TABLE VI. Correct and incorrect classification results from applying the confusion matrix After Enhancing by BFO algorithm

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	2195	1795	1761	1692	1017
	F	244	200	196	189	114
ZfNet	T	1804	1476	1448	1391	836
	F	635	519	509	490	295
VGG16	T	1853	1516	1487	1429	859
	F	586	479	470	452	272
VGG19	T	2048	1675	1643	1580	950
	F	391	320	314	301	181

TABLE VII. Correct and incorrect classification results from applying the confusion matrix After Enhancing by GWO algorithm

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	2195	1795	1761	1692	1017
	F	244	200	196	189	114
ZfNet	T	1804	1476	1448	1391	836
	F	635	519	509	490	295
VGG16	T	1829	1496	1467	1410	848
	F	610	499	490	471	283
VGG19	T	2048	1675	1643	1580	950
	F	391	320	314	301	181

TABLE VIII. Correct and incorrect classification results from applying the confusion matrix After Enhancing by PSO algorithm

CNNs Arch.		Nevus 2439	BCC 1995	PBK 1957	SK 1881	SCC 1131
AlexNet	T	2219	1815	1780	1711	1029
	F	220	180	177	170	102
ZfNet	T	1804	1476	1448	1391	836
	F	635	519	509	490	295
VGG16	T	1560	1276	1252	1203	723
	F	879	719	705	678	408
VGG19	T	2048	1675	1643	1580	950
	F	391	320	314	301	181

5. Conclusions and Future Work

Deep learning architectures (Convolutional Neural Networks) are long chains of engineering functions that are applied one by one after which these processes are organized into units called layers (deep learning models are usually stacks of layers or, more generally, layer graphs). The parameters of these layers are determined by weights and are the parameters learned during training. The knowledge of the model is stored in its weights and the learning process consists in finding good values for these weights. The proposed convolutional neural network architectures in this paper fulfill the requirements for accuracy in classification operations, especially in the field of machine perception, where useful information was extracted from the input digital images (data set) and their characteristics were extracted through the compatibility between classification architectures (LeNet, ZfNet, VGG16, and VGG19) and swarm algorithms to generate weights. After applying the proposed hybrid architectures, the results obtained showed the stability of the training processes for the input data and all architectures, and that the results of the training processes before the hybridization process were fluctuating in giving accuracy results. AlexNet architecture gave the highest classification accuracy when hybridized with Particle Swarm Optimization algorithm, while ZfNet architecture gave the highest classification accuracy when hybridized with Bat Algorithm, VGG16 architecture gave the highest classification accuracy when hybridized with Artificial Bee Algorithm, and VGG19 architecture gave the highest classification accuracy when hybridized with the Particle Swarm Optimization algorithm. Data augmentation processes (traditional augmentation and deep augmentation) have increased classification accuracy and effectively contributed to faster access to solutions.

ACKNOWLEDGMENT

The researchers would like to thank the Department of Computer Science and Software Department, College of Computer Science and Mathematics, University of Mosul for all the support.

References

[1] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Systems*, vol. 215, p. 106771, 2021.

[2] S. Rahman, L. Wang, C. Sun, and L. Zhou, "Deep learning based hep-2 image classification: A comprehensive review," *Medical Image Analysis*, vol. 65, p. 101764, 2020.

[3] H. M. Ahmed and M. Y. Kashmola, "A proposed architecture for convolutional neural networks to detect skin cancers," *IAES International Journal of Artificial Intelligence (IJ-AI)*, 2022.

[4] X. Yang and M. Sun, "A survey on deep learning in crop planting," in *IOP Conference Series: Materials Science and Engineering*, vol. 490, no. 6. IOP Publishing, 2019, p. 062053.

[5] I. Idrissi, M. Mostafa Azizi, and O. Moussaoui, "A lightweight optimized deep learning-based host-intrusion detection system deployed on the edge for iot," *International Journal of Computing and Digital System*, 2021.

[6] S. Begampure and P. Jadhav, "Intelligent video analytics for human action detection: a deep learning approach with transfer learning," *International Journal of Computing and Digital System*, 2021.

[7] R. Nair, V. Jain, and P. S. Nair, "Enhancing the performance measure of sentiment analysis through deep learning approach," *International Journal of Computing and Digital System*, 2021.

[8] J.-e. Liu and F.-P. An, "Image classification algorithm based on deep learning-kernel function," *Scientific programming*, vol. 2020, 2020.

[9] S. Zou, W. Chen, and H. Chen, "Image classification model based on deep learning in internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[10] Ò. Lorente, I. Riera, and A. Rana, "Image classification with classic and deep learning techniques," *arXiv preprint arXiv:2105.04895*, 2021.

[11] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, 2021.

[12] Z. Wu, S. Zhao, Y. Peng, X. He, X. Zhao, K. Huang, X. Wu, W. Fan, F. Li, M. Chen *et al.*, "Studies on different cnn algorithms for face skin disease classification based on clinical images," *IEEE Access*, vol. 7, pp. 66 505–66 511, 2019.

[13] S. G. Mohammed, A. H. Majeed, E. K. Ali Aldujaili, and S. S. Hassan, "Image segmentation for skin detection," *Journal of Southwest Jiaotong University*, vol. 55, no. 1, 2020.

[14] M. Jagdish, S. P. G. Guamangate, M. A. G. López, J. A. De La Cruz-Vargas, and M. E. R. Camacho, "Advance study of skin diseases detection using image processing methods," *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal— NVEO*, pp. 997–1007, 2022.

[15] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, vol. 21, no. 8, p. 2852, 2021.

[16] P. Yao, S. Shen, M. Xu, P. Liu, F. Zhang, J. Xing, P. Shao, B. Kaffenberger, and R. X. Xu, "Single model deep learning on imbalanced small datasets for skin lesion classification," *IEEE Transactions on Medical Imaging*, 2021.

[17] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS research*, vol. 43, no. 4, pp. 244–252, 2019.

[18] H. Wu and Z. Zhou, "Using convolution neural network for defective image classification of industrial components," *Mobile Information Systems*, vol. 2021, 2021.

[19] S. T. Krishna and H. K. Kalluri, "Deep learning and transfer learning approaches for image classification," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S4, pp. 427–432, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] V. Rotemberg, N. Kurtansky, B. Betz-Stablein, L. Caffery, E. Chousakos, N. Codella, M. Combalia, S. Dusza, P. Guitera, D. Gutman *et al.*, “Publisher correction: Author correction: A patient-centric dataset of images and metadata for identifying melanomas using clinical context,” *Scientific Data*, vol. 8, no. 1, pp. 1–1, 2021.

[22] L. Nanni, M. Paci, S. Brahnam, and A. Lumini, “Comparison of different image data augmentation approaches,” *Journal of Imaging*, vol. 7, no. 12, p. 254, 2021.

[23] Z. Fabian, R. Heckel, and M. Soltanolkotabi, “Data augmentation for deep learning based accelerated mri reconstruction with limited data,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 3057–3067.

[24] C. Shorten, T. M. Khoshgoftaar, and B. Furht, “Text data augmentation for deep learning,” *Journal of big Data*, vol. 8, no. 1, pp. 1–34, 2021.

[25] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[26] K. Horak and R. Sablatnig, “Deep learning concepts and datasets for image recognition: overview 2019,” in *Eleventh international conference on digital image processing (ICDIP 2019)*, vol. 11179. SPIE, 2019, pp. 484–491.

[27] S. Malek, “Deep neural network models for image classification and regression,” Ph.D. dissertation, University of Trento, 2018.

[28] Z. Gao, “Object-based image classification and retrieval with deep feature representations,” 2018.

[29] O. A. Malik, “Deep autoencoder for identification of abnormal gait patterns based on multimodal biosignals,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 1–8, 2021.

[30] M. Meddad, C. Moujahdi, M. Mikram, and M. Rziza, “A hybrid face identification system using a compressed cnn in a big data environment for embedded devices,” *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 689–701, 2020.

[31] K. Alrajhi and M. A. ELAffendi, “Automatic arabic part-of-speech tagging: Deep learning neural lstm versus word2vec,” *International Journal of Computing and Digital Systems*, vol. 8, no. 03, pp. 307–315, 2019.

[32] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan, “An automatic arabic sign language recognition system based on deep cnn: an assistive system for the deaf and hard of hearing,” *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 715–724, 2020.

[33] H. M. Ahmed and M. Y. Kashmola, “Generating digital images of skin diseases based on deep learning,” in *2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM)*. IEEE, 2021, pp. 179–184.

[34] A. Jaradat, E. Taqieddin, and M. Mowafi, “A high-capacity image steganography method using chaotic particle swarm optimization,” *Security and Communication Networks*, vol. 2021, 2021.

[35] G. Rossides, B. Metcalfe, and A. Hunter, “Particle swarm optimization—an adaptation for the control of robotic swarms,” *Robotics*, vol. 10, no. 2, p. 58, 2021.

[36] C. Guo, H. Tang, B. Niu, and C. B. P. Lee, “A survey of bacterial foraging optimization,” *Neurocomputing*, vol. 452, pp. 728–746, 2021.

[37] A. Kumar, S. Singh, A. Kumar *et al.*, “Grey wolf optimizer and other metaheuristic optimization techniques with image processing as their applications: a review,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1136, no. 1. IOP Publishing, 2021, p. 012053.

[38] Y. Wang, T. Wang, S. Dong, and C. Yao, “An improved grey-wolf optimization algorithm based on circle map,” in *Journal of Physics: Conference Series*, vol. 1682, no. 1. IOP Publishing, 2020, p. 012020.

[39] S. U. Umar and T. A. Rashid, “Critical analysis: bat algorithm-based investigation and application on several domains,” *World Journal of Engineering*, 2021.

[40] A. Alihodzic and M. Tuba, “Improved bat algorithm applied to multilevel image thresholding,” *The Scientific World Journal*, vol. 2014, 2014.

[41] R. A. Saleh and R. Akay, “Artificial bee colony algorithm with directed scout,” *Soft Computing*, vol. 25, no. 21, pp. 13 567–13 593, 2021.

[42] D. Chen, S. Zhang, Y. Yang, J. Liu, and C. Zhang, “Optimization of character image matching based on artificial bee colony algorithm,” in *Journal of Physics: Conference Series*, vol. 2035, no. 1. IOP Publishing, 2021, p. 012034.

Hasan Maher Ahmed He is a faculty member at the Software Department, University of Mosul, IRAQ. His M.Sc. degree was in Computer Science in the field of Image Processing and Artificial Intelligent from the University of Mosul/ IRAQ in 2009. His current area of research deals with Image Processing, Artificial Intelligent, Deep Learning, and Multimedia.

Manar Younis Kashmola She is a professor in Computer Networks. The Dean of the College of Information Technology, Ninevah University, and a faculty member at the Department of Computer Science, University of Mosul, IRAQ. She is also a Chairman of the Board of Directors of the Advisory Office at the University of Mosul She obtained her Ph. D. degree in Computer Science in the field of Computer Networks from the University of Mosul / IRAQ in 2004. Her current area of research Deals with Mobile Computing, IoT, wireless communication, and Distributed Systems.