OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 4 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

INTELLIGENT SKIN CANCER DIAGNOSIS USING PARTICLE SWARM OPTIMIZATION (PSO) AND TRANSFER LEARNING

Noor Ul Ain*
Department of Computer Science, NFC Institute of Engineering and technology, Multan, Pakistan.
Saroosh Jaffar
Institute of Computer Science and Information Technology, The Women University Multan, Pakistan.
Naeem Aslam
Department of Computer Science, NFC Institute of Engineering and technology, Multan, Pakistan.
Muhammad Sajid
Department of Computer Science, NCBA&E, Multan, Pakistan.
Muhammad Fuzail
Department of Computer Science, NFC Institute of Engineering and technology, Multan, Pakistan.

****Corresponding author: Noor Ul Ain (noorulaincheema@gmail.com)***

Article Info

Abstract

It's true that when our skin spends a lot of time in the sun, abnormal cell growth can occur, and that's what we know as skin cancer. Interestingly, this common cancer can also pop up on areas of skin that don't see much sunlight. The main types of skin cancer are melanoma, squamous cell carcinoma, and basal cell carcinoma. While skin cancer can be serious, even fatal, the outcome really depends on a few things: the specific type of skin cancer, the person's overall health, and how early the cancer is found. Melanoma is often called "the most serious" because it has a greater tendency to spread. It can develop within an existing mole or suddenly appear as a dark spot that looks different from the surrounding skin. On the other hand, basal cell and squamous cell carcinomas are less likely to be life-threatening.

The good news is that artificial intelligence is making huge strides in quickly and accurately identifying many diseases, which is helping people get treatment sooner. One AI technique, called convolutional neural networks, is particularly good at looking at images and providing very precise information.

One study explored using an efficient convolutional neural network model to analyze skin cancer images. They then used the model to extract key features from these images. To get the best features, they tried out different combinations and used optimization algorithms called PSO and GA to select them. Finally, they used an SVM approach to classify each set of chosen features and aimed for the best possible results. This method achieved an accuracy of 89.17%, showing that it could be quite effective.

Another study found that a model called U-Net++ with densenet201 as its backbone performed even better across various measures like accuracy, F1-score, AUC, iou, and dice values, achieving scores of 94.16%, 91.39%, 99.3%, 96.8%, 77.19%, and 75.47% respectively.

Sadly, about 3.5 million people in the United States alone are diagnosed with skin cancer each year. The chances of survival drop significantly as skin cancer progresses. However, finding this type of cancer early can be difficult and expensive. To address this, one study used an automated, threshold-based method to detect, categorize, and segment skin cancer cases. They even used a smart optimization tool called spasa to fine-tune the settings of eight well-known CNN models (like VGG16, VGG19, mobile net, and nasnet) to get the best results.

Keywords:
Skin cancer detection, Transfer learning, Classical machine learning-based approach, Deep learning-based approach.

1. Introduction

Cancer is really a complex group of diseases where the body's cells start growing and spreading uncontrollably. When these abnormal cells invade nearby tissues and form tumors, they can mess with how our organs normally work. Skin cancer, specifically, begins in the cells of our skin, which is actually the largest organ we have. Often, it's caused by too much exposure to ultraviolet (UV) light, whether that's from the sun or artificial sources like tanning beds.

There are different types of skin cancer, and each one has its own characteristics. Melanoma is considered the most serious because it can be life-threatening if it's not caught and treated early. This highly aggressive type of skin cancer starts in the melanocytes, the cells that produce pigment in our skin, and it has the ability to spread quickly to other parts of the body.

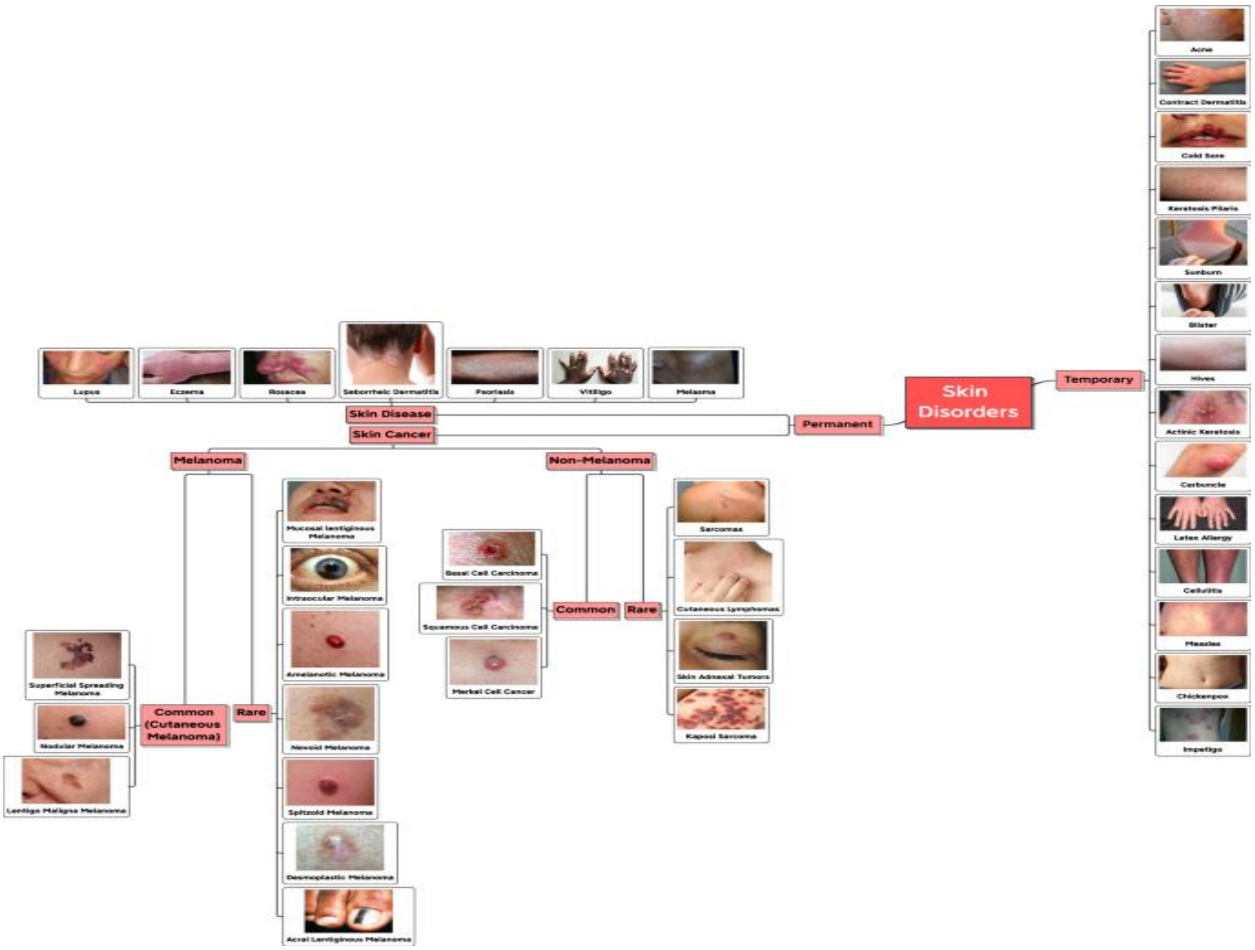


Image 1: Skin disorders taxonomy graphical summary

Dermatologists have traditionally used visual inspection to diagnose skin cancer. They evaluate the presence of lesions, moles, or other abnormalities on the skin. Although this method has merit, it is limited because it mostly relies on the practitioner's subjective expertise. [3]

Furthermore, there's a chance that conventional diagnostic techniques won't always be able to accurately distinguish benign from malignant lesions or pick up on subtle indicators of skin cancer in its early stages. [4]

Now consider the revolutionary way that skin cancer diagnosis has been transformed by the merger of Particle Swarm Optimization (PSO), Convolutional Neural Networks (CNN), and Artificial Intelligence (AI). This novel method increases the accuracy and efficacy of detection by utilizing computational intelligence. [5]

A bio-inspired optimization method called particle swarm optimization (PSO) imitates the group dynamics of living things like fish schools or flocks of birds. PSO is used to optimize the parameters used in image processing and feature extraction in the context of diagnosing skin cancer. This helps in locating pertinent structures and patterns in dermatological images that may be suggestive of cancer. [6]

Convolutional Neural Networks (CNN) are a type of artificial neural network specifically created for image identification applications, complementing PSO. These networks are especially good at spotting indications of skin cancer in dermatological images because they are very good at learning hierarchical representations of features inside images. CNN's inclusion into the diagnostic process improves the system's capacity to identify minute characteristics and patterns that correspond to various forms of skin cancer. [7]

Reusing a pre-trained model on a new task is known as transfer learning in machine learning. A machine uses transfer learning to increase its generalization about another task by using the knowledge it has learned from a prior one. For instance, you may use the knowledge a classifier learns to identify drinks while training it to determine whether an image contains food. [8]

The overarching intelligence directing the combination of CNN and PSO is artificial intelligence (AI). AI gives the system the ability to learn from large datasets using machine learning and transfer learning techniques, thereby increasing the diagnostic accuracy of the system and combining PSO, CNN, and AI results in a powerful diagnostic framework that not only overcomes the drawbacks of conventional techniques but also guarantees flexibility in response to changing diagnostic problems. [9]

A paradigm shift in the diagnosis of skin cancer has been brought about by the combination of PSO, CNN, and AI, transfer learning, which provides a more efficient, accurate, and objective way to discover malignancies. The potential for early detection by this intelligent system to enable prompt intervention and treatment could result in life-saving benefits. The convergence of medical diagnostics and computational intelligence presents an opportunity to transform skin cancer diagnosis for the benefit of worldwide healthcare as we continue to push the boundaries of technological innovation. [10]

2. Related Studies.

Scientists are still actively exploring better ways to segment (precisely outline), detect, and recognize melanoma. To help doctors with diagnosis, many automated techniques and strategies have been developed. Looking back at previous work in this area, we can generally group the approaches into two main categories: machine learning (ML) and deep learning (DL) methods.

2.1 Classical machine learning-based approaches:

Traditional machine learning approaches for identifying skin cancer typically involve several steps: first, the images are prepared (pre-processing); then, important characteristics are identified (feature extraction); sometimes, these characteristics are simplified (reduction); and finally, the image is classified. That feature extraction step is really crucial because the accuracy of the final classification depends on it.

The features that are pulled from the images can be broadly categorized into two types: basic, overall features (low-level or global) and more detailed, localized features (high-level or local).

For example, one researcher, Pugazhenthi [12], used something called a gray-level co-occurrence matrix (GLCM) to pull out texture-related features like contrast, entropy, energy, and inverse difference moment from the segmented skin images. Then, they used decision trees to figure out the skin condition and classify it as eczema, leprosy, or melanoma, achieving a certain level of precision.

Another group, Arivuselvam et al. [13], used a fuzzy clustering method along with GLCM and a Gabor filter to extract features like size, color, and texture from the input images. They calculated these feature values for 1,500 images and then used an SVM classifier to categorize them. Khan [14] took a different approach by first removing noise from the skin lesion images using a Gaussian filter. Then, they segmented the lesion using an improved K-mean clustering technique to create a combined set of unique features. An SVM was then used for classification. They tested their method on the DERMIS dataset, which included 397 skin cancer images, 251 nevus (mole) images, and 146 melanoma images, and they reported an accuracy of 96%.

Astorino [15] proposed a method called multiple-instance learning. They used it on 160 clinical images, dividing them into 80 nevi and 80 melanomas. This approach allowed them to achieve 92.50% accuracy, 97.50% sensitivity (correctly identifying melanomas), and 87.50% specificity (correctly identifying non-melanomas).

Balaji [16] focused on segmenting the skin lesions using a dynamic graph cut technique and then used a Naive Bayes classifier to categorize the skin disorders. They evaluated their method on the ISIC 2017 dataset and reported accuracy rates of 94% for benign (non-cancerous) cases, 91% for melanoma, and 92% for keratosis (another skin condition).

Murugan [17] also worked on segmentation using a watershed method. The resulting segments then went through feature extraction, and they used the ABCD rule (a common method for evaluating moles), GLCM, and the shape of the extracted features to classify the data. They tested several classifiers, including SVM, random forest, and K-nearest neighbor (KNN), and their results showed an accuracy of 89.43%, sensitivity of 91.15%, and specificity of 87.71%.

Finally, İlkin [18] used the SVM algorithm as a classifier, enhanced by a bacterial colony technique, with a Gaussian radial basis function. They trained and tested their model on the ISIC and PH2 datasets and achieved AUC (Area Under the Curve, a measure of classifier performance) values of 98% and 97% for ISIC and PH2, respectively.

2.2 Deep learning-based approaches:

Around the late 1990s, we saw a shift from computer systems that relied entirely on human-designed features to those that learned from the data itself. Instead of manually selecting features, as mentioned before, the computers were then tasked with figuring out how to extract the relevant features from the input to solve the problem. Many deep learning algorithms today are built on this idea of automatically learning to identify important features from the data they are given.

For example, Adegun and Viriri [20] used a special type of neural network called an enhanced encoder-decoder network with interconnected sub-networks to both extract features and learn from them. They tested their approach on two well-known datasets, PH2 and ISBI 2017. For the ISBI 2017 dataset, they reported an accuracy of 95% and a dice coefficient (a measure of overlap) of 92%, while for the PH2 dataset, they achieved 95% accuracy and a 93% dice coefficient.

Albahli [21] used a combination of Yolov4-darknet (for locating the melanoma) and active contour (for precisely outlining it). They evaluated their algorithm using the ISIC datasets from 2016 and 2018 and reported a Jaccard coefficient (another measure of overlap) of 0.989 and a dice score of 1.

Shan [22] developed a segmentation method called FC-DPN, which is based on a fully convolutional network with two pathways. When tested on a modified version of the ISIC 2017 challenge dataset, their method achieved a Jaccard index of 82% and an average dice coefficient of 88%. For the PH2 dataset, they reported a Jaccard index of 83% and an average dice coefficient of 90%.

Junaid [23] presented a CNN-based model for classifying skin cancer. They first gathered a collection of skin cancer images and divided them into four groups. To increase the size of their dataset, they used data augmentation techniques. During testing, their model outperformed Google Net and Mobile Net with an accuracy of 95.89%, showing improvements of 1.12% and 1.76%, respectively.

Alheejawi [24] proposed a deep learning method for segmenting melanoma areas. Using a relatively small dataset of melanoma images, their method achieved a dice coefficient of approximately 85% for segmentation. They emphasized the quick processing time and thorough examination capabilities of their system.

Vani [25] suggested a deep learning approach to predict the presence and type of melanoma. They used pre-processing techniques to improve the quality of the images used for classification. For the melanoma classification itself, they employed self-organizing maps (SOM) and CNN classifiers, claiming their method was accurate and specific.

Li and Jimene [26] introduced a unique testing strategy based on Alex Net and the extreme learning machine network. They also fine-tuned the settings of their method using an enhanced version of the Grasshopper optimization algorithm (GOA). When compared to other state-of-the-art methods, their approach showed high efficiency and achieved 98% accuracy and 93% sensitivity on the PH2 dataset.

Hasan [27] presented a framework for automatically classifying skin lesions. Their proposed solution combined pre-processing with a hybrid convolutional neural network. To create more detailed feature maps of the lesion, they merged information from three separate feature extraction modules. The pre-processing steps included lesion segmentation, data augmentation, and techniques to balance the different classes of images. They tested their algorithm on the ISIC-2016, ISIC-2017, and ISIC-2018 datasets, achieving an AUC (Area Under the Curve, a measure of performance) of 96%, 95%, and 97%, respectively, on these datasets.

Finally, Maniraj and Maran [28] proposed a hybrid deep learning method based on 3D wavelet sub band fusion. Their method involved three main steps: classifying the images into multiple categories, applying a 3D wavelet transform, and using simple median filtering. When tested on the PH2 dataset, it was able to accurately classify normal, benign (non-cancerous), and malignant (cancerous) skin images with an average accuracy of 99.33% and a sensitivity and specificity of over 90%.

Table 1 Related Studies Summary:

Year	Approach	Classification Performance	Dataset	Best Performance

2022	Alex Net + ELM	PH2 dataset	PH2	Accuracy: 98%, Sensitivity: 93%
2022	CNN	ISIC 2016, 2017, 2018 datasets	-	-
2022	Hybrid DL	PH2 dataset	PH2	Avg. Accuracy: 99.33%, Sensitivity/Specificity > 90%
2021	SVM + Fuzzy Clustering	Own dataset	-	Accuracy of 92% and sensitivity of80.11%
2021	CNN	Own dataset	Own dataset	95.98% accuracy
2021	Improved NS-Net DL	Own dataset from Cross Cancer Institute	Own dataset	Dice Coefficient: ~85%
2021	SOM + CNN	Own dataset from ISIC archive	Own dataset from ISIC archive	Accuracy: 90%, Specificity: 99%
2020	CNN (yolov4)	ISIC 2016, ISIC 2018 datasets	ISIC 2016, ISIC 2018	Avg. Accuracy: 93.99%,
2020	FCN and Dual Path Network	ISIC 2017 dataset	ISIC 2017	Dice Coefficient 90.26%
2020	Multiple Instance Learning	Own dataset	Own dataset	Accuracy92.50%, Sensitivity97%, Specificity 87%
2020	Dynamic Graph Cut and Naive Bayes	ISIC 2017 dataset	ISIC 2017	Sensitivity, Specificity, Diagnostic Accuracy 915, 70% and 72%
2019	Decision Tree	Own dataset	Own dataset	Accuracy 87%

2019	SVM, Random Forest, KNN	ISIC 2016 dataset	ISIC 2016	Accuracy, Sensitivity, Specificity 89.41%, 91.2% and 87%
2019	K-means Clustering	DERMIS dataset	DERMIS	Accuracy 96%
2019	Deep Convolutional Encoder-Decoder	PH2, ISBI 2017 datasets	PH2, ISBI 2017	Accuracy, Dice Coefficient 95% and 93%

3. Plan of solution:

Skin cancer detection, classification, and segmentation are critical and challenging tasks in medical imaging applications. Several deep learning architectures are put forth in this study to address the segmentation and classification of skin cancer. The training parameters and hyperparameters are tuned, or optimized, using transfer learning (TL) and Spa SA. Various performance criteria are used for evaluation, and different experiments are carried out. The greatest architectures are recorded, saved, and shared so they can be used later. [29]

4. The proposed system.

The brief overview of the history and foundational elements of the suggested methodology in this section. They are essential to the methods section. It is divided into the subsequent sections:

- Data scaling and augmentation.
- Segmentation.
- Deep learning (DL) classification.
- Transfer learning (TL).
- Parameters optimization.
- Meta-heuristic optimization.
- Performance metrics

4.1 Here we discuss only the Transfer learning (TL) based proposed system.

Transfer learning refers to using a previously learned model for a new task. TL is increasingly being used in DL as deep neural networks that can be trained with tiny amounts of data [30]. In data science, it becomes extremely useful because most real-world problems lack a significant amount of classified data needed to train sophisticated models [31] In TL, the generalization of the second task is improved by utilizing the knowledge gained from the first task [32].. Reducing the requirement that the training and test sets of data be independent and equally distributed can help with generalization. The comprehensive concept is to use the information that the model, which has been trained on a large amount of labeled data, has acquired in a unique task with a tiny quantity of information. Domain adaptation, domain confusion, multitask learning, one-shot learning, and zero-shot learning are the five categories of transfer learning. [33]

Four settings that depend on "what to transfer" in learning are used to categorize the TL process. One, two, three, and four-way techniques are (1) instance transfer, (2) feature-representation transfer [34], and (3) relational knowledge transfer. For DL in TL, a previously trained stored model serves as the foundation. This makes it possible to advance quickly and perform well [35]. There are numerous pre-trained CNN models at one's disposal, including VGG16 [36], ResNet [37], mobile net [38], Xception [39], nasnet [40], and dense net.

The architectures of nasnet [40], VGG, and mobile net are used for image categorization. The present study employs the following architectures: VGG16 and VGG19 [41]; Nas net large and nasnet Mobile; mobile net; mobilenetv2; mobilenetv3small and Mobile Net V3 Large. In every categorization trial, the input image's size is (100*100*3)

4.1.1 Nasnet

Through the notion of NAS, which is based on reinforcement learning, the nasnet Google ML group has brought the idea of an optimal network to life. The architecture consists of a CNN and a controller RNN that need to be taught. Two input picture sizes of 331*331 and 224*224 are used to train the nasnet, yielding the nasnet Large and NASNetMobile designs, respectively. There is a significant expansion in several factors while switching from NASNetMobile to Nas net large. Because Nas net large has 88, 949, and 818 parameters compared to 5, 326, and 716 for NASNetMobile, Nas net large is less trustworthy. [42]

4.1.2 Mobile net

Mobile net was specifically built to efficiently enhance accuracy with the consideration of limited resources for an on-device system. Mobile net incorporates low-latency and low-power models to suit the resource limitations of computing devices. Mobile net utilizes separable filters, which consist of a point- and a depth-wise convolution. To minimize the computational overheads of normal convolution operations, it applies filters of size 1*1. Consequently, the size of the network and computation complexity decrease. The input 24*24*3 size image contains 4.2 million parameters in mobile net. [43]

4.1.3 Mobilenetv2

The architecture of mobilenetv2 is similar to that of the original mobile net, with the exception of removing nonlinearities in thin layers and using inverted residual blocks with bottlenecking features [108]. Compared to the original mobile net, it contains fewer parameters. Mobile nets can handle any input size more than 32 by 32, and they perform better with larger image sizes. There are two types of blocks in mobilenetv2. The first is a residual block the size of one stride. The other, a block the size of two strides, is used for shrinking. Each of the two kinds of blocks has three levels. A depth-wise convolution comes next, followed by another 1 * 1 convolution without an activation function. The first layer is a 1 * 1 convolution with relu6. [44]

4.1.4 Mobilenetv3

The application of automl to identify the best neural network architecture for a specific task is mobilenetv3's main contribution. Precisely, mobilenetv3 combines two automl techniques: nasnet and net adapt. Mnasnet was initially employed by mobilenetv3 to discover a coarse architecture. Using reinforcement learning, Mnasnet selected the optimal configuration from a discrete set. Subsequently, net adapt fine-tunes the architecture by iteratively removing the unused activation channels. Aiming at high and low resource applications, respectively, mobilenetv3 is defined by two models: mobilenetv3large and mobilenetv3small. Mobilenetv3large has 20% less latency and 3.2% higher accuracy on image

classification compared to mobilenetv2. Similarly, mobilenetv3small achieves a 6.6% higher accuracy with comparable latency. [45]

4.1.5 VGG model

Convolution and pooling layers are stacked to generate the VGG. The network depth of VGG16 is 16 layers, and vanishing gradients are not a problem. It consists of three dense layers with two 4096-sized layers, five max-pooling layers, and thirteen convolution layers. All the hidden layers employed a nonlinear relu activation function, while the final layer utilized a SoftMax function. In contrast, there are 19 layers in the VGG19 network depth. There are three thick layers with two 4096-sized layers, five max pool layers, and sixteen convolution layers. Like in VGG16, the relu activation function is used in all hidden layers of VGG19, while the SoftMax function is used in the final layer. [46]

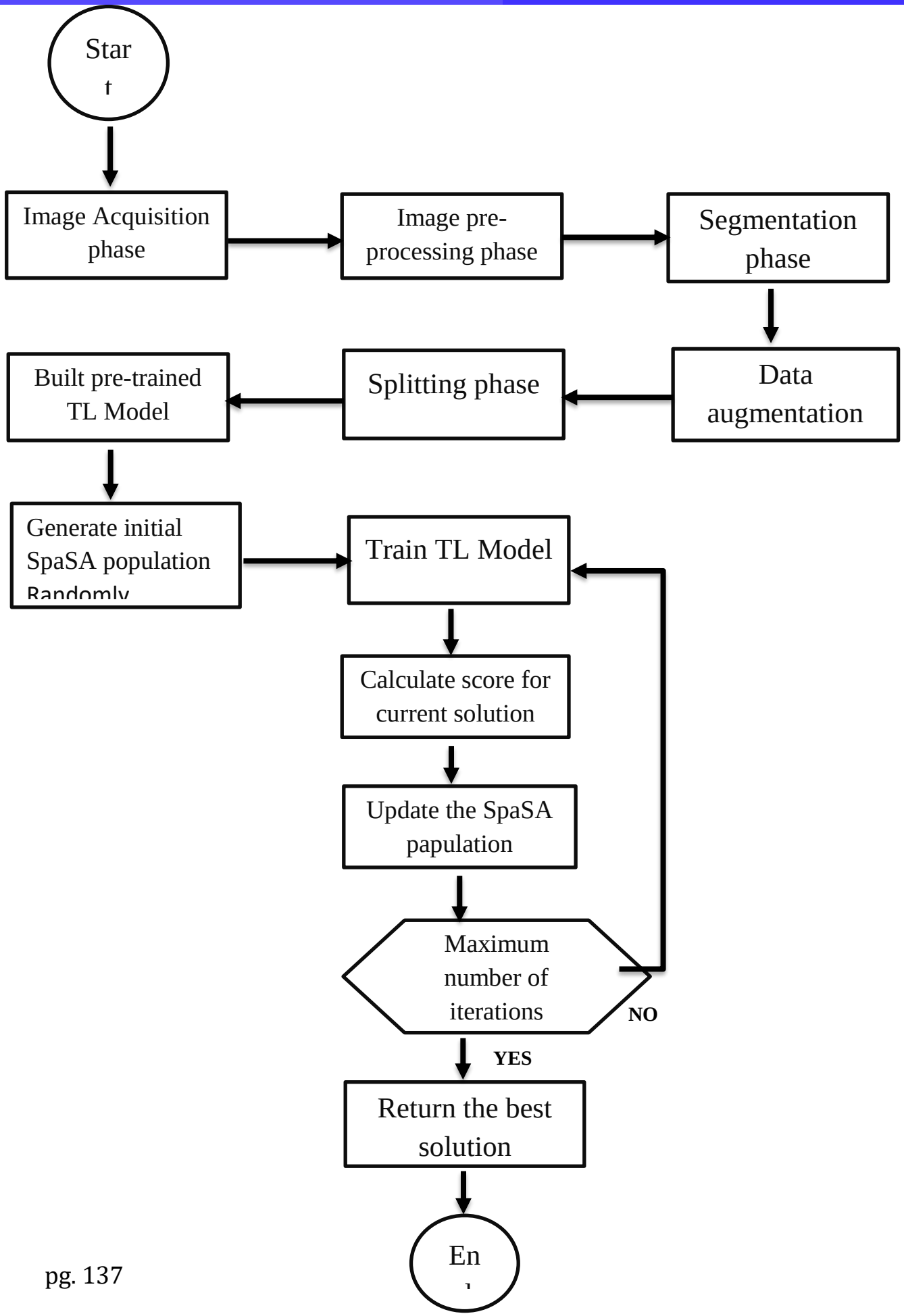
5. Methodology.

The input layer accepts the photos. They undergo pre-processing in the following stage, which involves scaling, balancing, and dataset augmentation. After that, the pre-trained models that have been recommended can be used to classify and segment the photos. The meta-heuristic optimization and transfer learning phase comes to an end at this point. Following completion, a prelude is provided for the figures, statistics, and post-trained models. The phases are discussed in the ensuing subsections.

5.1 The overall pseudocode and flowchart.

Algorithm 1: The Suggested Learning and Hyperparameters Optimization Pesudocode.

Input: $modelName, X, Y$ // The model name (e.g., VGG16) and the dataset (X: images and Y: labels).
Output: X_P, f_{best} // The best overall score and solution (i.e, combination).
SR // The dataset split ratio is defined in Table 10.
T // The maximum number of iterations and set to 10 in the current study (Table 10).
Ps // The maximum number of sparorws and set to 10 in the current study (Table 10).
D // The number of hyperparameters required to be optimized and set to 15 in the current study (Table 10).
$t = 1$ // Initialize the iterations counter.
$trainValX, testX, trainValY, testY = \textbf{SplitDataset}(X, Y, SR)$ // Partition the dataset into training (and validation) and testing portions concerning the split ratio defined in Table 10.
$trainX, validationX, trainY, validationY = \textbf{SplitDataset}(trainValX, trainValY, SR)$ // Partition the dataset into training and validation portions concerning concerning the split ratio defined in Table 10.
$model = \textbf{BuildTLModel}(modelName)$ // Build the pre-trained TL model with the ImageNet pre-trained weights.
$population = \textbf{GenerateInitialPopulationForSpaSA}(Ps, D)$ // Generate the initial population for the SpaSA with the size of $(Ps \times D)$.
// Execute the learning SpaSA hyperparameters optimization process (discussed in Section 5.4) for T iterations.
while $(t \leq T)$ **do**
  // Work during $t \leq T$.
  // First: Calculate the corresponding population fitness scores.
  $fitnessScoresArray = []$ // Initialize a fitness scores array.
  $i = 1$ // Initialize the sparrow' counter.
  while $(i \leq Ps)$ **do**
    // Work during $i \leq Ps$.
    $fitnessScore = \textbf{TrainModelAndCalculateFitnessScore}(model, population[i], trainX, trainY, validationX, validationY, testX, testY)$ // Train the TL model and calculate the fitness score for the current solution.
    $\textbf{PushFitnessScore}(fitnessScoresArray, score)$ // Push the fitness score into the scores array in-place.
    $i = i + 1$ // Increment the sparrow' counter.
  end
  // Second: Sort the population scores.
  $\textbf{SortArray}(population, fitnessScoresArray)$ // Sort the fitness scores array in descending order.
  // Third: Extract the required solutions.
  $X_{best}, X_{worst}, X_P, f_{best}, f_{worst} = \textbf{ExtractSolutions}(population, fitnessScoresArray)$ // Extract the best, worst, and optimial solutions. Also, extract the best and worst scores. They will be injected in the updating process using SpaSA.
  // Fourth: Start the SpaSA updating process using Equation 20, Equation 21, and Equation 21.
  $i = 1$ // Initialize a loop counter.
  while $(i \leq PD)$ **do**
    // Work during $i \leq PD$ using Equation 20.
    if $(R2 < ST)$ **then**
      $population[i] = population[i] \times exp^{(\frac{i}{\alpha \times T})}$
    else
      $population[i] = population[i] + Q \times L$
    end
    $i = i + 1$ // Increment the loop counter.
  end
  $i = 1$ // Initialize a loop counter.
  while $(i \leq (n - PD))$ **do**
    // Work during $i \leq (n - PD)$ using Equation 21.
    if $(i > 0.5 \times n)$ **then**
      $population[i] = Q \times exp^{\left(\frac{X_{worst} - population[i]}{i^2}\right)}$
    else
      $population[i] = X_P + |population[i] - X_P| \times A^+ \times L$
    end
    $i = i + 1$ // Increment the loop counter.
  end
  $i = 1$ // Initialize a loop counter.
  while $(i \leq SD)$ **do**
    // Work during $i \leq SD$ using Equation 22.
    if $(scoresList[i] \neq f_{best})$ **then**
      $population[i] = X_{best} + \beta \times |population[i] - X_{best}|$
    else
      $population[i] = population[i] + K \times \left(\frac{|population[i] - X_{worst}|}{(scoresList[i] - f_{worst}) + \epsilon}\right)$
    end
    $i = i + 1$ // Increment the loop counter.
  end
  $t = t + 1$ // Increment the iterations counter.
end
return X_P, f_{best} // Return the best overall score and solution (i.e, combination).


Star
t
Image Acquisition phase
Image pre-processing phase
Segmentation phase
Data augmentation
Splitting phase
Built pre-trained TL Model
Generate initial SpaSA population Randomly
Train TL Model
Calculate score for current solution
Update the SpaSA papulation
Maximum number of iterations
NO
YES
Return the best solution
En
d

5.2. Dataset acquisition:

Five publicly available datasets are downloaded from Kaggle and used in the current investigation. "ISIC 2019 and 2020 Melanoma dataset" is the name of the first dataset [47], [48], [49], There are 11,449 photos in all. Two classes are distinguished within it: "MEL" and "NEVUS."

It can be downloaded and used from https://www.kaggle.com/datasets/qikangdeng/isic-2019-and-2020-melanoma-dataset

The second one is called "Classification of Melanoma (HAM10000). [50]" It has 10,015 different-sized pictures that make up its composition. Two classes are separated from it: "Melanoma" and "Not Melanoma."

It can be downloaded and used from https://www.kaggle.com/adacslicml/melanoma-classification-ham10k

The third one is named "Image dataset for skin diseases." There are various-sized photos and the total composition. Ten classes are present in it: "Eczema," "Melanocytic Nevi," "Atopic Dermatitis," "Basal Cell Carcinoma," "Benign Keratosis-like Lesions," "Psoriasis images, Lichen Planus, and associated disorders," "Seborrheic Keratoses and other Benign Tumors." "Warts Molluscum and other Viral Diseases" and "Tinea Ringworm Candidiasis and other Fungal Diseases."

It can be downloaded and used from https://www.kaggle.com/ismailpromus/skin-diseases-image-dataset

The fourth one is called “Skin Cancer Segmentation and Classification”. It is composed of images in which images are of different sizes

It can be downloaded and used from https://www.kaggle.com/surajghuwalewala/ham1000-segmentation-and-classification

The fifth one is named “PH2”. It is composed of 200 dermo scopic images of melanocytic lesions. It can be downloaded and used from https://www.fc.up.pt/addi/ph2%20database.html

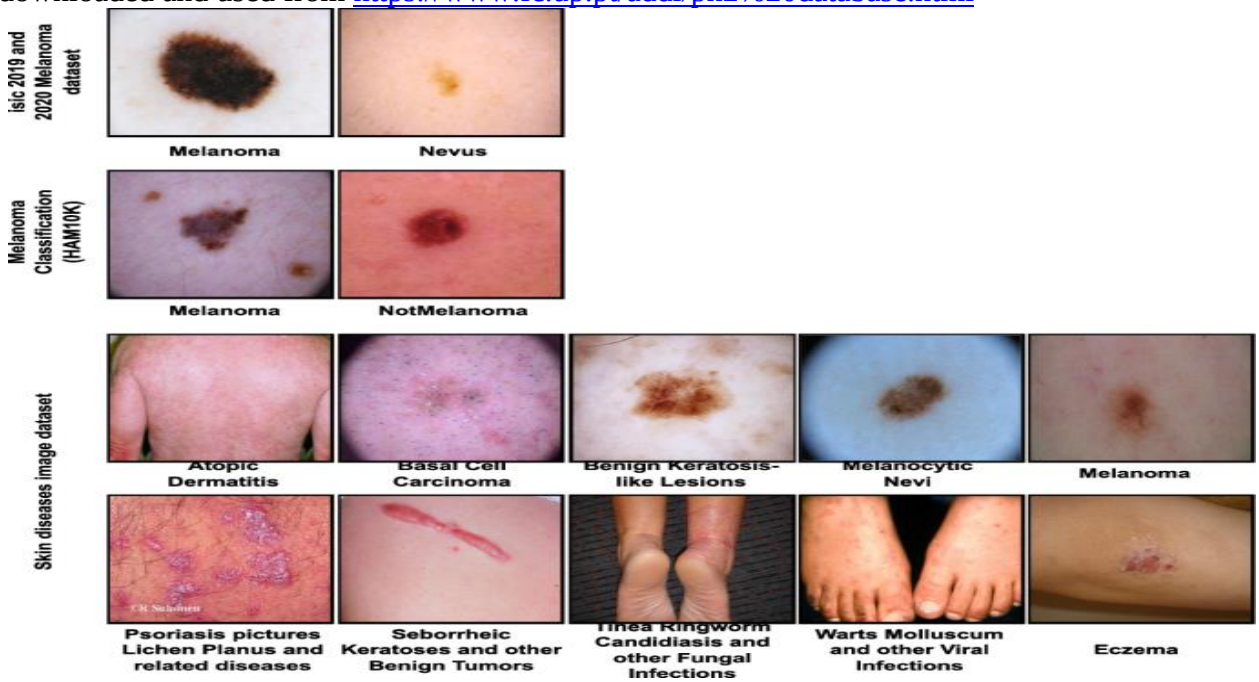


Image 2: Sample from the used dataset.

5.3.Dataset pre-processing:

Data scaling is discussed and the corresponding equations that are used in the current study are for standardization, for normalization, for the min-max scaler, and for the max-absolute scaler where mean and σ is the standard deviation.

Output = input - μ / σ

Output = input / max (input)

Output = input – min (input) / max (input) – min (input)

Output = input / | max (input) |

5.4 Dataset augmentation and balancing:

Since there are unequal numbers of photos in each category, data balancing is used to balance the categories prior to the training procedure.

$$\text{Flipping Matrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}\,\theta & -sin\theta \\ 0 & \text{Sin}\,\theta & \text{Cos}\,\theta \end{bmatrix}$$

$$\text{Rotation matrix} = \begin{bmatrix} \text{Cos}\,\theta & sin\theta & 0 \\ -sin\theta & \text{Cos}\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Shearing Matrix} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Zooming Matrix} = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5.5 Learning and optimization:

It is important to tune some DL training hyperparameters (as shown in Table) to achieve the SOTA performance. The methods of tuning the hyperparameters are grid search, meta-heuristic optimization algorithms, and try-and-error. Because try-and-error does not include the ranges of the hyperparameters, it is a weak method. It is, however, included by the grid search, however, it requires several months to complete the search.

Layer	**Parameters**	**Hyperparameters**
Convolution layer	Kernels' weights	Number of kernels, kernel size, Stride, activation function, and padding
Pooling layer		Filter size, pooling method, Padding, and stride
Fully connected layer	Neurons' Weights	Activation function and

		Number of weights
Others		Optimizer, model architecture, Loss function, learning rate, epochs, Batch size, weight initialization, Dataset splitting, and regularization

Table 1: DL training hyperparameters

5.6 Sparrow search algorithm (spasa):

In order to find the optimal combinations, this method will be utilized to solve the optimization problem. Initially, all sparrow populations and their attributes are randomly selected from the specified ranges. (As shown in table)

Configuration	**Specifications**
Image size	(100×100×3) for classification
Train split ratio	85% to 15%
Shuffle dataset	Yes
Number of epochs	5
Hyperparameters optimizer	Sparrow Search Algorithm (spasa)
Spasa population size	10
Spasa number of iterations	10
Output activation function	SoftMax
Early stopping patience	5
Pre-trained parameters initializers	ImageNet
Pre-trained models	Mobilenhet, mobilenetv2, mobilenetv3small, mobilenetv3large, VGG16, VGG19, NASNetMobile, and nasnetlarge
Loss	Categorical Cross entropy, Categorical Hinge, kldivergence, Poisson, Squared Hinge, and Hinge

Parameters optimizer	Adam, nadam, adagrad, adadelta, adamax, rmsprop, SGD, Ftrl, SGD Nesterov, rmsprop Centered, and Adam amsgrad
Dropout range	[0, 0.6]
Batch size	4 to 48 with a step of 4
Pre-trained model learn ratio	1 to 100 with a step of 1
Scaling techniques	Normalize, Standard, Min Max, and Max Abs
Apply data augmentation	Boolean (Yes or No)
Rotation range	0∘ to 45∘ with a step of 1∘
Width shift range	[0, 0.25]
Height shift range	[0, 0.25]
Shear range	[0, 0.25]
Zoom range	[0, 0.25]
Horizontal flip range	Boolean (Yes or No)
Vertical lip range	Boolean (Yes or No)
Brightness range	[0.5, 2.0]

Table 2: All sparrow population and their attribute.

5.7 Initial population:

First, a random choice of the sparrow population and its relevant parameters is done. Spasa initializes the population by using an arbitrary method. It is not fixed. I is the solution index, j is the dimension index, and xij is the ith sparrow's location in the jth search space. D, or the number of hyperparameters to optimize, will have the value of 15 in this research. It will create a population of (Ps * D), with Ps representing the population size (number of sparrows), and in this case, Ps is being set at 10.

$$Xi, j = lbj + (ubj — lbj) * random (1, D)$$

5.8 Objective function calculation:

For each sparrow, the objective function is invoked to determine the associated score. The problem at hand is a maximization one: the sparrow does better the higher the value. To simplify this step, consider the objective function as a black box, where the score in this case, accuracy will be the output and the solution, the input. What happens inside? The previously defined 15 variables are fetched and appended to the pre-trained CNN model (e.g., VGG16) upon accepting the solution that needs to be tested. These particular variables are first utilized by the model to start learning (i.e., the training and validation

procedure). Then, in order to find the overall performance metrics, the entire dataset is tested. Finally, the accuracy is returned by the objective function.

Accuracy = TP + TN / TP+ TN+ FP+ FN

Specificity = TNR = TN / TN+ FP

Sensitivity = RECALL = TPR = TP / TP +FN

FNR = FN / FN + TP = 1- TRP

FPR = fallout = 1 – TNR = FP / FP + TN

Precision = PPV = TP / TP + FP

Dice = 2 × TP / 2 × TP + FP + FN

JAC = TP / TP + FP + FN = dice / 2 – dice

AUC = 1- FPR + FNR / 2 = 1 – 0.5 × (FP / FP + TN + FN / FN + TP)

5.9 Population sorting:

Following the computation of each sparrow's objective function inside the population set, the sparrows are arranged in descending order based on the values of the objective function.

5.10 Selection:

The current best individual X best and worst individual X worst and their fitness values are picked to be applied in the updating process.

5.11 Population updating

The person with the highest fitness values uses spasa to determine who gets to lead the population movement and gather food throughout the search process. Therefore, it is crucial to update the sparrow location for producers, which may be accomplished with Equation

$$X_{i+j}^{t+1} = \begin{cases} X_{i,j}^{t} \times (-\frac{h}{a \times T}), & If\ R_2 < ST \\ X_{i,j}^{t} + Q \times L, & otherwise \end{cases}$$

In order to improve their nutrition, some of the followers also keep an eye on the discoverers and those with high rates of food predation. The equation is utilized to update the position of the followers.

$$X_{i+j}^{t+1} = \begin{cases} Q \times exp^{(x_{worst}^{t} - x_{i,j}^{t})}, & if\ (i > 0.5 \times n \\ X_{i,j}^{t} + \left|X_{i,j}^{t} - X_{P}^{t+1}\right| \times A^{+} \times L, \end{cases}$$

It is estimated that just 10% to 20% of all sparrows are conscious of the threat. The initial places of the sparrows in the population are generated at random using

$$X_{i+j}^{t+1} = \begin{cases} X_{best}^{t} + \beta \times \left|X_{i,j}^{t} - X_{best}^{t}\right|, & If\ f_i \neq f_g \\ X_{i,j}^{t} + k \times (\frac{X_{i,j}^{t} - X_{worst}^{t}}{(f_i - f_\omega) + c}), & otherwise \end{cases}$$

6. Evaluation and Experiments.

The experiments are separated into two groups: (1) segmentation experiments and (2) experiments related to learning, optimization, and categorization.

6.1 Experiments configurations:

The "Python" programming language is generally used for coding and testing in this current study. Google Colab is the learning and optimization environment with a GPU. The most used Python packages are TensorFlow, Keras, Keras-unit-collection, NumPy, OpenCV, Pandas, and Matplotlib. 15% is for testing and 85% for training and validation. Random dataset shuffling is applied in the learning process. In the RGB color space, the images are resized to (128*128*3) for segmentation and to (100*100*3) for classification. The general settings of the trials are listed in Table 1, the optimization, learning, and classification-specific settings are listed in Table 2, and the segmentation-specific settings are listed in Table 3.

The used experiment's common configurations:

Configuration	**Specifications**
Scripting language	Python
Python packages	TensorFlow, Keras, Keras-Unet-collection, NumPy, OpenCV, Scikit-Learn, SciPy, Pandas, and Matplotlib
Learning and optimization environment	Google Colab (Intel(R) Xeon(R) CPU @ 2.00 ghz, Tesla T4 16 GB GPU with CUDA v.11.2, and 12 GB RAM)

Table 3: Common configuration.

The used optimization, learning, and classification-specific configurations:

Configuration	**Specifications**
Image size	(100×100×3) for classification
Train split ratio	85% to 15%
Shuffle dataset	Yes
Number of epochs	5
Hyperparameters optimizer	Sparrow Search Algorithm (spasa)
Spasa population size	10
Spasa number of iterations	10
Output activation function	SoftMax

Early stopping patience	5
Pre-trained parameters initializers	ImageNet
Pre-trained models	Mobilenet, mobilenetv2, mobilenetv3small, mobilenetv3large, VGG16, VGG19, NASNetMobile, and nasnetlarge
Loss	Categorical Cross entropy, Categorical Hinge, kldivergence, Poisson, Squared Hinge, and Hinge
Parameters optimizer	Adam, nadam, adagrad, adadelta, adamax, rmsprop, SGD, Ftrl, SGD Nesterov, rmsprop Centered, and Adam amsgrad
Dropout range	[0, 0.6]
Batch size	4 to 48 with a step of 4
Pre-trained model learn ratio	1 to 100 with a step of 1
Scaling techniques	Normalize, Standard, Min-Max, and Max Abs
Apply data augmentation	Boolean (Yes or No)
Rotation range	0∘ to 45∘ with a step of 1∘
Width shift range	[0, 0.25]
Height shift range	[0, 0.25]
Shear range	[0, 0.25]
Zoom range	[0, 0.25]
Horizontal flip range	Boolean (Yes or No)
Vertical lip range	Boolean (Yes or No)
Brightness range	[0.5, 2.0]

Table 4: Optimization, learning, and classification-specific configurations

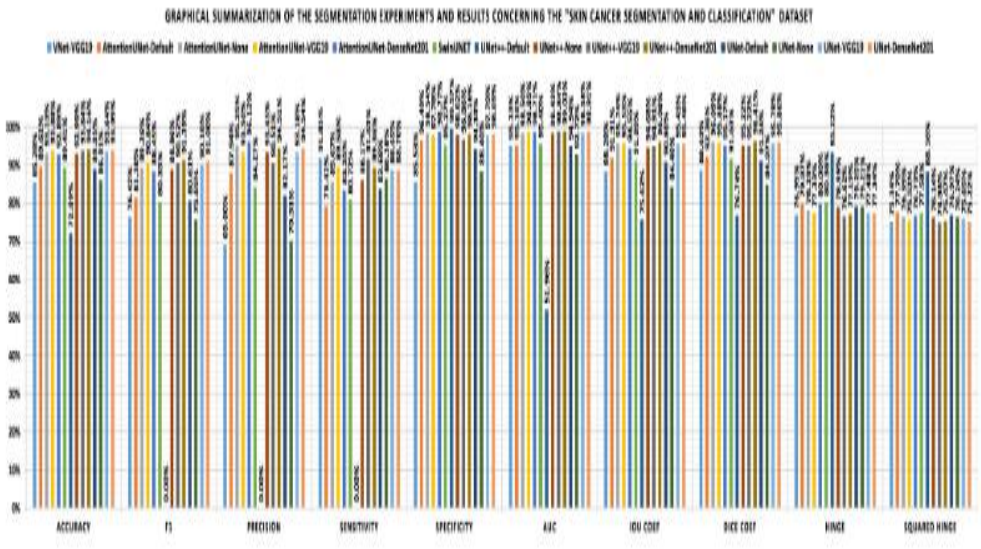
The used segmentation-specific configurations:

Configuration	**Specifications**
Image size	(128×128×3) for segmentation

Train split ratio	85% to 15%
Filters	[64, 128, 256, 512, 1024]
Loss	Binary cross entropy
Parameters optimizer	Adam
Batch size	4
Number of epochs	10
Early stopping patience	5
Output activation function	Sigmoid
Pre-trained parameters initializers	ImageNet
Freeze backbone	Yes
Freeze batch normalization	Yes

Table 5: Segmentation-specific configurations

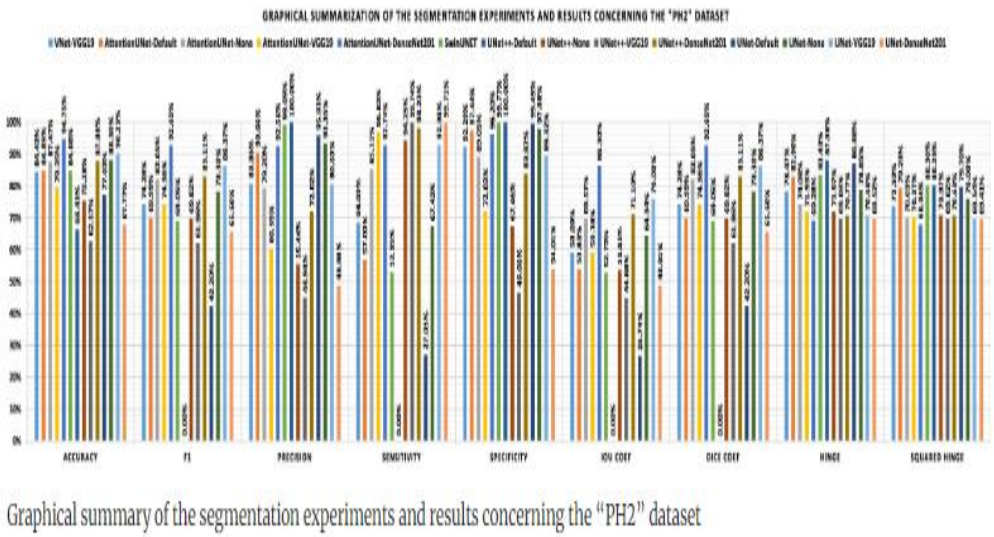
Graphical summary of the segmentation experiments and results concerning the "Skin cancer segmentation and classification" dataset.



Graphical summary of the segmentation experiments and results concerning the "Skin cancer segmentation and classification" dataset

Graph 1: Graphical summary of the segmentation experiments

Graphical summary of the segmentation experiments and results concerning the "PH2" dataset.



Graph 2: Results concerning the "PH2" dataset.

6.2 Learning and Optimization Experiments.

The experiments concerning learning and optimization for the pre-trained TL CNN models (Mobilenet, mobilenetv2, mobilenetv3small, mobilenetv3large, VGG16, VGG19, NASNetMobile, and nasnetlarge) and the spasa meta-heuristic optimizer are introduced and discussed within the following subsection. Five epochs are included in total. Both the population size and the SpaSA iterations number are fixed at ten. The following metrics are captured and reported: mean absolute error, mean iou, mean squared error, mean squared logarithmic error, root mean squared error, TP, TN, FP, FN, recall, sensitivity, specificity, AUC, iou coef., Dice coef., cosine similarity, and logcosh error.

The "ISIC 2019 and 2020 Melanoma Dataset" experiments

The confusion matrix results concerning the "ISIC 2019 and 2020 Melanoma dataset" dataset

Model name	TP	TN	FP	FN
Mobilenet	11,250	11,250	198	198
Mobilenetv2	11,129	11,129	295	295
Mobilenetv3small	10,647	10,647	801	801
Mobilenetv3large	10,994	10,994	446	446
VGG16	10886	10,886	538	538
VGG19	11098	11,098	350	350
NASNetMobile	11,062	11,062	362	362
Nasnetlarge	10,670	10,670	778	778

Table 6: Melanoma Dataset experiments

The "ISIC 2019 and 2020 Melanoma dataset" dataset experiments with the maximized metrics

Model name	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)	Sensitivity (%)	Specificity (%)	AUC (%)	Iou (%)	Dice (%)	Cosine similarity (%)
Mobilenet	98.27	98.27	98.27	98.27	98.27	98.27	99.42	98.00	98.32	98.49
Mobilenetv2	97.42	97.42	97.42	97.42	97.42	97.42	98.19	98.12	98.21	97.56
Mobilenetv3small	93.00	93.00	93.00	93.00	93.00	93.00	97.87	93.37	94.28	94.12
Mobilenetv3large	96.10	96.10	96.10	96.10	96.10	96.10	98.85	96.01	96.58	96.61
VGG16	95.29	95.29	95.29	95.29	95.29	95.29	99.13	94.51	95.49	96.25
VGG19	96.94	96.94	96.94	96.94	96.94	96.94	99.19	96.11	96.78	97.23
NASNetMobile	96.83	96.83	96.83	96.83	96.83	96.83	99.14	96.69	97.16	97.19
Nasnetlarge	93.20	93.20	93.20	93.20	93.20	93.20	98.18	90.67	92.48	94.46

Table 7: Melanoma Dataset experiments with the maximized metrics

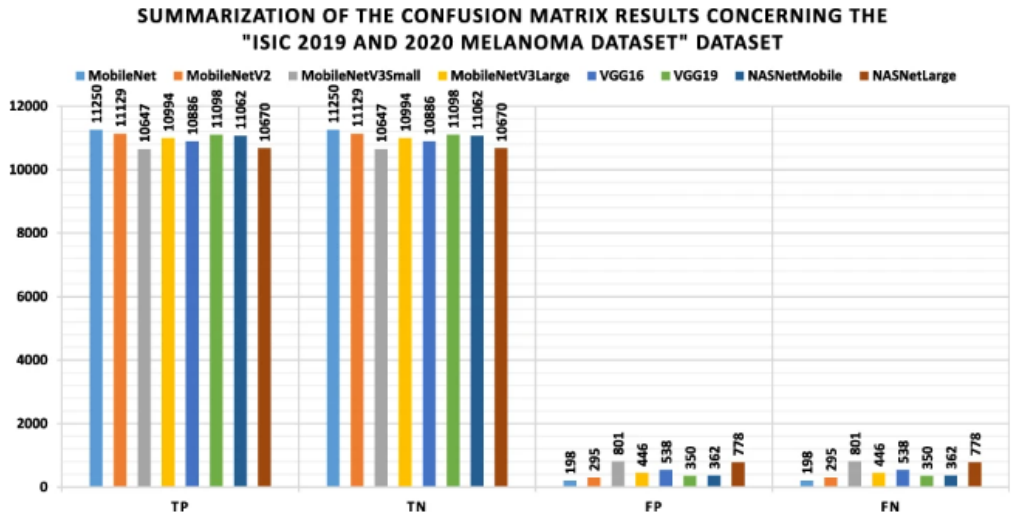
The "ISIC 2019 and 2020 Melanoma dataset" dataset experiments with the minimized metrics

Model name	Logcosh error	Mean absolute error	Mean iou	Mean squared error	Mean squared logarithmic error	Root mean squared error
Mobilenet	0.006	0.025	0.512	0.014	0.007	0.120
Mobilenetv2	0.010	0.027	0.685	0.024	0.011	0.154
Mobilenetv3small	0.024	0.086	0.323	0.054	0.026	0.232

Mobilenetv3large	0.014	0.051	0.478	0.032	0.015	0.178
VGG16	0.016	0.068	0.264	0.034	0.017	0.185
VGG19	0.012	0.048	0.363	0.026	0.013	0.162
NASNetMobile	0.012	0.043	0.289	0.026	0.013	0.162
Nasnetlarge	0.024	0.113	0.250	0.051	0.025	0.225

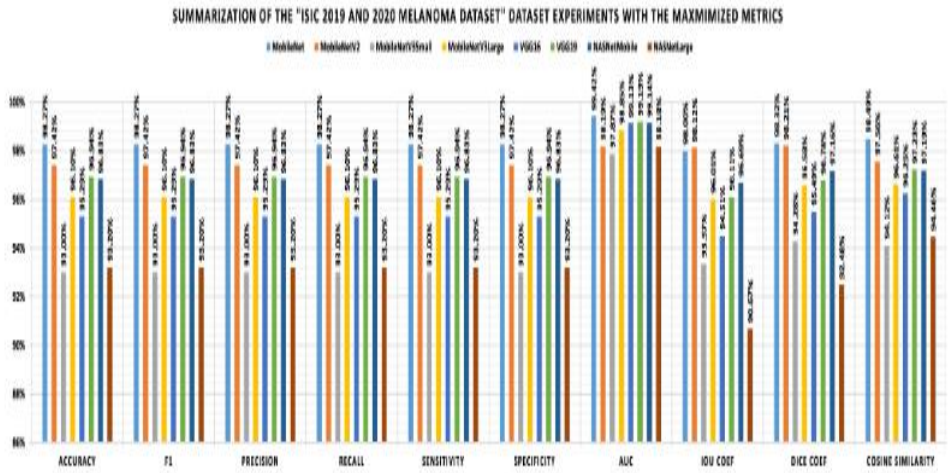
Table 8: Melanoma Dataset experiments with the minimized metrics

The Summary of the confusion matrix result concerning the "ISIC 2019 and 2020 Melanoma dataset" dataset.



Graph 3: Confusion matrix result concerning the "ISIC 2019 and 2020 Melanoma dataset" dataset.

The summary of the "ISIC 2019 and 2020 Melanoma dataset" dataset experiments with the maximized metrics.



Graph 4: "ISIC 2019 and 2020 Melanoma dataset" dataset experiments with the maximized metrics.

The "Melanoma Classification (HAM10K)" experiments.

The confusion matrix results concerning the "Melanoma Classification (HAM10K)" dataset

Model name	TP	TN	FP	FN
Mobilenet	9867	9867	117	117
Mobilenetv2	9745	9745	263	263
Mobilenetv3small	8999	8999	985	985
Mobilenetv3large	9720	9720	264	264
VGG16	9417	9417	591	591
VGG19	9811	9811	189	189
NASNetMobile	9323	9323	677	677
Nasnetlarge	9163	9163	837	837

Table 9: Melanoma Dataset (HAM10K) experiments.

The "Melanoma Classification (HAM10K)" dataset experiments with the maximized metrics

Model name	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)	Sensitivity (%)	Specificity (%)	AUC (%)	Iou (%)	Dice (%)	Cosine similarity (%)
Mobilenet	98.83	98.83	98.83	98.83	98.83	98.83	99.45	98.83	98.97	98.91
Mobilenetv2	97.37	97.37	97.37	97.37	97.37	97.37	98.26	97.95	98.07	97.44
Mobilenetv3small	90.13	90.13	90.13	90.13	90.13	90.13	90.13	93.42	93.42	90.13
Mobilenetv3large	97.36	97.36	97.36	97.36	97.36	97.36	99.21	96.93	97.42	97.62
VGG16	94.09	94.09	94.09	94.09	94.09	94.09	98.44	92.10	93.58	95.06

VGG19	98.11	98.11	98.11	98.11	98.11	98.11	99.52	97.87	98.18	98.24
NASNetMobile	93.23	93.23	93.23	93.23	93.23	93.23	97.59	94.39	95.00	94.29
Nasnetlarge	91.63	91.63	91.63	91.63	91.63	91.63	95.07	91.07	92.30	92.35

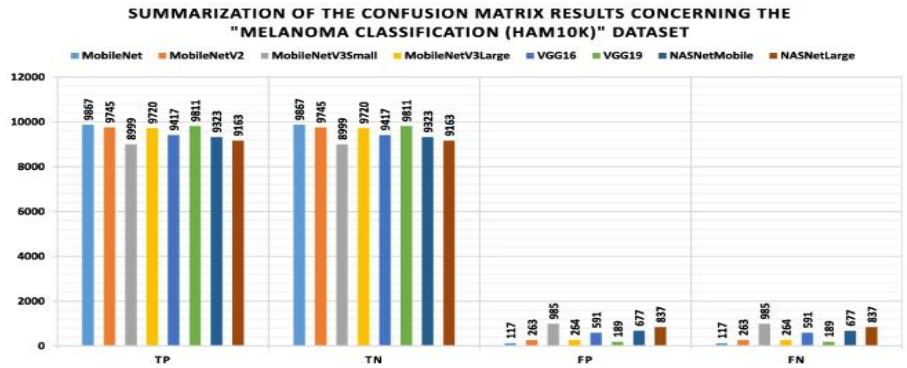
Table 10: Melanoma Dataset (HAM10K) experiments with the maximized metrics

The "Melanoma Classification (HAM10K)" dataset experiments with the minimized metrics.

Model name	Logcosh error	Mean absolute error	Mean iou	Mean squared error	Mean squared logarithmic error	Root mean squared error
Mobilenet	0.005	0.015	0.585	0.010	0.005	0.102
Mobilenetv2	0.011	0.029	0.779	0.025	0.012	0.158
Mobilenetv3small	0.043	0.099	0.820	0.099	0.047	0.314
Mobilenetv3large	0.010	0.039	0.274	0.022	0.011	0.150
VGG16	0.021	0.096	0.251	0.046	0.022	0.213
VGG19	0.007	0.027	0.251	0.017	0.008	0.129
NASNetMobile	0.024	0.075	0.412	0.053	0.026	0.231
Nasnetlarge	0.033	0.115	0.258	0.072	0.035	0.269

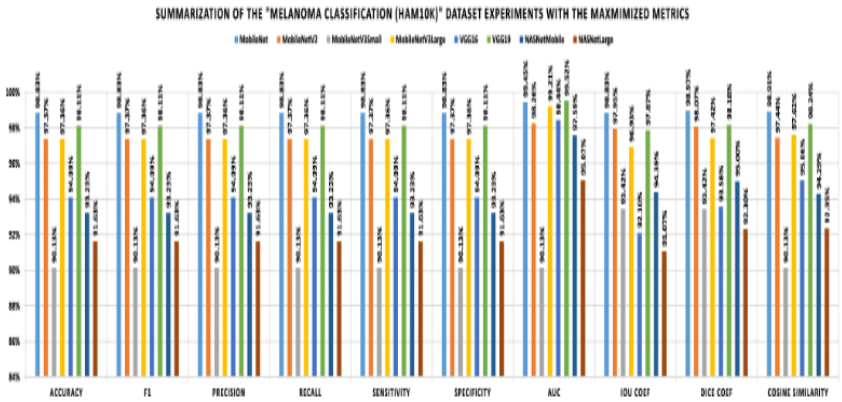
Table 11: Melanoma Dataset (HAM10K) experiments with the minimized metrics

Summary of the confusion matrix results concerning the "Melanoma Classification (HAM10K)" dataset



Graph 5: Confusion matrix results concerning the "Melanoma Classification (HAM10K)" Dataset

Summary of the "Melanoma Classification (HAM10K)" dataset experiments with the maximized metrics



Graph 6 "Melanoma Classification (HAM10K)" dataset experiments with the maximized metrics

The "Skin diseases image dataset" experiment.

The confusion matrix results concerning the "Skin diseases image dataset"

Model name	TP	TN	FP	FN
Mobilenet	21,111	240,729	3351	6009
Mobilenetv2	22,130	241,562	2518	4990
Mobilenetv3small	9657	241,447	2849	17,487
Mobilenetv3large	16,843	240,391	3689	10,277
VGG16	18,261	241,430	2938	8891
VGG19	18,873	242,176	2156	8275
NASNetMobile	20,983	238,828	5396	6153
Nasnetlarge	20,455	237,669	6519	6677

Table 12: Skin diseases data set experiments

The "Skin diseases image dataset" experiments with the maximized metrics

Model name	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)	Sensitivity (%)	Specificity (%)	AUC (%)	Iou (%)	Dice (%)	Cosine similarity (%)
Mobilenet	82.09	81.68	86.22	77.84	77.84	98.63	98.35	81.63	83.96	85.02

Mobilenetv2	85.87	85.33	89.74	81.60	81.60	98.97	99.01	83.28	85.72	88.08
Mobilenetv3small	54.64	45.67	75.16	35.58	35.58	98.83	91.75	55.92	60.32	62.93
Mobilenetv3large	72.11	70.30	82.03	62.11	62.11	98.49	96.80	71.52	74.89	76.87
VGG16	76.19	74.45	86.05	67.25	67.25	98.80	97.70	75.44	78.41	80.29
VGG19	79.30	77.97	89.72	69.52	69.52	99.12	98.28	76.43	79.52	82.88
NASNetMobile	78.23	78.36	79.54	77.33	77.33	97.79	95.91	82.97	84.31	80.78
Nasnetlarge	75.59	75.60	75.84	75.39	75.39	97.33	91.19	82.81	83.42	76.88

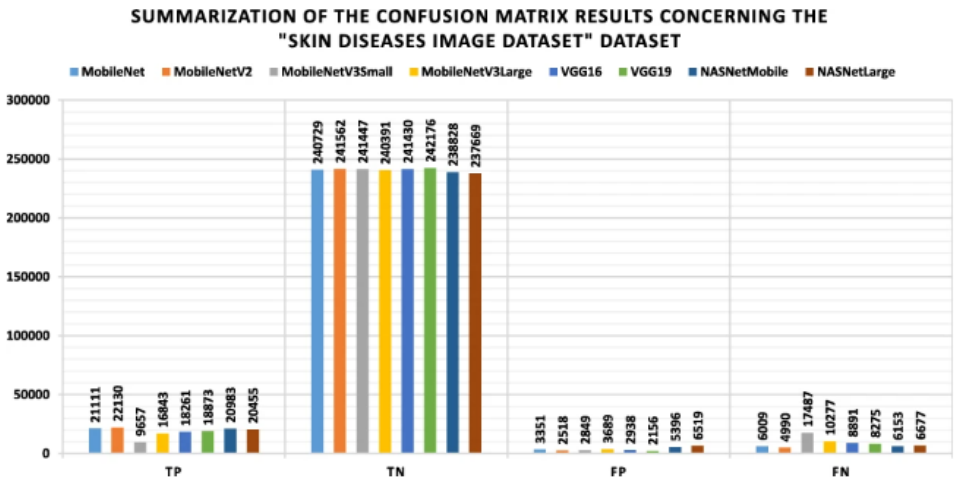
Table 13: Skin diseases experiments with the maximized metrics

The "Skin diseases image dataset" experiments with the minimized metrics

Model name	Logcosh error	Mean absolute error	Mean iou	Mean squared error	Mean squared logarithmic error	Root mean squared error
Mobilenet	0.012	0.048	0.452	0.026	0.013	0.160
Mobilenetv2	0.010	0.043	0.451	0.021	0.010	0.143
Mobilenetv3small	0.027	0.119	0.450	0.057	0.028	0.240
Mobilenetv3large	0.018	0.075	0.450	0.038	0.019	0.195
VGG16	0.015	0.065	0.450	0.032	0.016	0.179
VGG19	0.013	0.061	0.450	0.029	0.014	0.169
NASNetMobile	0.015	0.047	0.476	0.034	0.017	0.185
Nasnetlarge	0.019	0.050	0.585	0.044	0.021	0.210

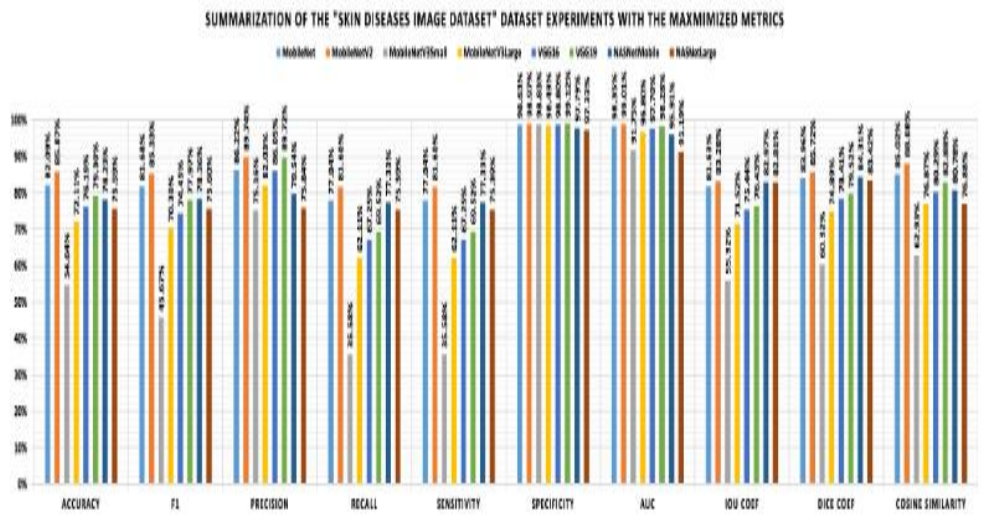
Table 14: Skin diseases experiments with the minimized metrics

The summary of the confusion matrix results concerning the "Skin diseases image dataset".



Graph 7: confusion matrix results concerning the "Skin diseases image dataset".

The summary of the "Skin diseases image dataset" experiments with the maximized metrics.



Graph 8: Skin diseases image dataset" experiments with the maximized metrics

7. Conclusion

Automatically detecting and segmenting skin cancer is a subject of continuous research and needs to be continuously developed. In the current research, we utilized a methodological strategy based on PSO, Transfer Learning, and CNN to automatically classify different skin images into their corresponding categories. CNNs are one of the most advanced techniques in image classification and handling large data.

The study investigated a number of previous works in transfer learning-based skin cancer detection, classification, and segmentation. Various experiments were performed, and the results were presented. For the segmentation task, with the "Skin cancer segmentation and classification" dataset, the U-Net++ with densenet201 as a backbone was the best model according to various metrics like accuracy, F1-score, AUC, iou, and dice values. This architecture attained accuracy scores of 94.16%, 91.39%, 99.3%, 96.8%, 77.19%, and 75.47% based on accuracy, F1-score, AUC, iou, dice, hinge, and squared hinge, respectively.

Also, for the "PH2" dataset, the Attention U-Net with densenet201 as a backbone became the winner, providing scores of 0.137, 94.75%, 92.65%, 92.56%, 92.74%, 96.20%, 86.30%, 92.65%, 96.28%, and 68.4% for loss, accuracy, F1-score, precision, sensitivity, specificity, iou, dice, hinge, and squared hinge respectively.

For tuning the hyper parameters of the pre-trained CNN models to optimal values, the space meta-heuristic optimizer was used throughout the learning, classification, and optimization processes. We utilized different CNN architectures like VGG16, VGG19, Mobilenet, mobilenetv2, mobilenetv3small, mobilenetv3large, nasnetlarge, and NASNetMobile. Different 2-class and 10-class datasets were prepared from various public sources.

The research presented remarkable results in terms of accuracy across various datasets. For the "ISIC 2019 and 2020 Melanoma" dataset, the mobile net pre-trained model had an overall accuracy of 98.27%, whereas for the "Melanoma Classification (HAM10K)" dataset, the mobile net pre-trained model had 98.83% accuracy. For the "Skin diseases image" dataset, the mobilenetv2 pre-trained model had the highest overall accuracy of 85.87%.

Comparative study against 13 of the previous relevant works proved our research to have outperformed many of those previous attempts. There are challenges still, and they are most notably because this field lacks abundant datasets. The future research projects intend to venture into other machine or deep learning methods, increase the performance level of the segmentation of skin cancer phase, and test the system using other available datasets.

Reference:

[1] Khan, Z., Hossain, M. Z., Mayumu, N., Yasmin, F., & Aziz, Y. (2024, November). Boosting the Prediction of Brain Tumor Using Two Stage BiGait Architecture. In 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 411-418). IEEE.

[2] Khan, S. U. R., Raza, A., Shahzad, I., & Ali, G. (2024). Enhancing concrete and pavement crack prediction through hierarchical feature integration with VGG16 and triple classifier ensemble. In 2024 Horizons of Information Technology and Engineering (HITE)(pp. 1-6). IEEE https://doi.org/10.1109/HITE63532.

[3] Khan, S.U.R., Zhao, M. & Li, Y. Detection of MRI brain tumor using residual skip block based modified Mobilenet model. Cluster Comput 28, 248 (2025). https://doi.org/10.1007/s10586-024-04940-3

[4] Khan, U. S., & Khan, S. U. R. (2024). Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT. Multimedia Tools and Applications, 1-21.

[5] Asif, S., Khan, S. U. R., Amjad, K., & Awais, M. (2024). SKINC-NET: an efficient Lightweight Deep Learning Model for Multiclass skin lesion classification in dermoscopic images. Multimedia Tools and Applications, 1-27.

[6] Raza, A., Salahuddin, & Inzamam Shahzad. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of Engineering Sciences, 2(3), 367–396.

[7] Asif, S., Awais, M., & Khan, S. U. R. (2023). IR-CNN: Inception residual network for detecting kidney abnormalities from CT images. Network Modeling Analysis in Health Informatics and Bioinformatics, 12(1), 35.

[8] Khan, M. A., Khan, S. U. R., Haider, S. Z. Q., Khan, S. A., & Bilal, O. (2024). Evolving knowledge representation learning with the dynamic asymmetric embedding model. Evolving Systems, 1-16.

[9] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

[10] Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2024). Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 1-32.

[11] Khan, U. S., Ishfaque, M., Khan, S. U. R., Xu, F., Chen, L., & Lei, Y. (2024). Comparative analysis of twelve transfer learning models for the prediction and crack detection in concrete dams, based on borehole images. Frontiers of Structural and Civil Engineering, 1-17.

[12] Khan, S. U. R., & Asif, S. (2024). Oral cancer detection using feature-level fusion and novel self-attention mechanisms. Biomedical Signal Processing and Control, 95, 106437.

[13] Farooq, M. U., Khan, S. U. R., & Beg, M. O. (2019, November). Melta: A method level energy estimation technique for android development. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

[14] Raza, A.; Meeran, M.T.; Bilhaj, U. Enhancing Breast Cancer Detection through Thermal Imaging and Customized 2D CNN Classifiers. VFAST Trans. Softw. Eng. 2023, 11, 80–92.

[15] Dai, Q., Ishfaque, M., Khan, S. U. R., Luo, Y. L., Lei, Y., Zhang, B., & Zhou, W. (2024). Image classification for sub-surface crack identification in concrete dam based on borehole CCTV images using deep dense hybrid model. Stochastic Environmental Research and Risk Assessment, 1-18.

[16] Waqas, M., Ahmed, S. U., Tahir, M. A., Wu, J., & Qureshi, R. (2024). Exploring Multiple Instance Learning (MIL): A brief survey. Expert Systems with Applications, 123893.

[17] Waqas, M., Tahir, M. A., Al-Maadeed, S., Bouridane, A., & Wu, J. (2024). Simultaneous instance pooling and bag representation selection approach for multiple-instance learning (MIL) using vision transformer. Neural Computing and Applications, 36(12), 6659-6680.

[18] Waqas, M., Tahir, M. A., & Qureshi, R. (2023). Deep Gaussian mixture model based instance relevance estimation for multiple instance learning applications. Applied intelligence, 53(9), 10310-10325.

[19] Waqas, M., Tahir, M. A., & Khan, S. A. (2023). Robust bag classification approach for multi-instance learning via subspace fuzzy clustering. Expert Systems with Applications, 214, 119113.

[20] Khan, S.U.R.; Asif, S.; Bilal, O.; Ali, S. Deep hybrid model for Mpox disease diagnosis from skin lesion images. Int. J. Imaging Syst. Technol. 2024, 34, e23044.

[21] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X.; Zhu, Y. GLNET: Global–local CNN's-based informed model for detection of breast cancer categories from histopathological slides. J. Supercomput. 2023, 80, 7316–7348.

[22] Hekmat, Arash, Zuping Zhang, Saif Ur Rehman Khan, Ifza Shad, and Omair Bilal. "An attention-fused architecture for brain tumor diagnosis." Biomedical Signal Processing and Control 101 (2025): 107221.

[23] Salahuddin, Syed Shahid Abbas, Prince Hamza Shafique, Abdul Manan Razzaq, & Mohsin Ikhlaq. (2024). Enhancing Reliability and Sustainability of Green Communication in Next-Generation Wireless Systems through Energy Harvesting. Journal of Computing & Biomedical Informatics.

[24] Salahuddin, Hussain, M., hamza Shafique, P., & Abbas, S. S. (2024). Intelligent melanoma detection based on pigment network. Kashf Journal of Multidisciplinary Research, 1(10), 1-14.

[25] Salahuddin, Hussain, M., & hamza Shafique, P. (2024). Performance analysis of matched filter-based secondary user detection in cognitive radio networks. Kashf Journal of Multidisciplinary Research, 1(10), 15-26.

[26] Shahzad, Inzamam, Asif Raza, and Muhammad Waqas. "Medical Image Retrieval using Hybrid Features and Advanced Computational Intelligence Techniques." Spectrum of engineering sciences 3, no. 1 (2025): 22-65.

[27] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. Int. J. Imaging Syst. Technol. 2024, 34, e22975.

[28] Raza, A., Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).

[29] M. Wajid, M. K. Abid, A. Asif Raza, M. Haroon, and A. Q. Mudasar, "Flood Prediction System Using IOT & Artificial Neural Network", VFAST trans. Softw. Eng., vol. 12, no. 1, pp. 210–224, Mar. 2024.

[30] Khan, S.U.R.; Raza, A.;Waqas, M.; Zia, M.A.R. Efficient and Accurate Image Classification Via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 10–23.

[31] Farooq, M.U.; Beg, M.O. Bigdata analysis of stack overflow for energy consumption of android framework. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–9.

[32] HUSSAIN, S., Raza, A., MEERAN, M. T., IJAZ, H. M., & JAMALI, S. (2020). Domain Ontology Based Similarity and Analysis in Higher Education. IEEEP New Horizons Journal, 102(1), 11-16.

[33] Khan, Zia, Saif Ur Rehman Khan, Omair Bilal, Asif Raza, and Ghazanfar Ali. "Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization." In 2025 6th International Conference on Advancements in Computational Sciences (ICACS), pp. 1-7. IEEE, 2025.

[34] Shahzad, I., Khan, S. U. R., Waseem, A., Abideen, Z. U., & Liu, J. (2024). Enhancing ASD classification through hybrid attention-based learning of facial features. Signal, Image and Video Processing, 1-14.

[35] Mahmood, F., Abbas, K., Raza, A., Khan,M.A., & Khan, P.W. (2019). Three Dimensional Agricultural Land Modeling using Unmanned Aerial System (UAS). International Journal of Advanced Computer Science and Applications (IJACSA) [p-ISSN : 2158-107X, e-ISSN : 2156-5570], 10(1).

[36] M. Waqas, Z. Khan, S. U. Ahmed and A. Raza, "MIL-Mixer: A Robust Bag Encoding Strategy for Multiple Instance Learning (MIL) using MLP-Mixer," 2023 18th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2023, pp. 22-26.

[37] Meeran, M. T., Raza, A., & Din, M. (2018). Advancement in GSM Network to Access Cloud Services. Pakistan Journal of Engineering, Technology & Science [ISSN: 2224-2333], 7(1).

[38] Khan, S. R., Raza, A., Shahzad, I., & Ijaz, H. M. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).

[39] Asif Raza, Inzamam Shahzad, Ghazanfar Ali, and Muhammad Hanif Soomro. "Use Transfer Learning VGG16, Inception, and Reset50 to Classify IoT Challenge in Security Domain via Dataset Bench Mark." Journal of Innovative Computing and Emerging Technologies 5, no. 1 (2025).

[40] Bilal, Omair, Asif Raza, and Ghazanfar Ali. "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures." VFAST Transactions on Software Engineering 12, no. 1 (2024): 79-92.

[41] Bilal, O., Asif, S., Zhao, M., Khan, S. U. R., & Li, Y. (2025). An amalgamation of deep neural networks optimized with Salp swarm algorithm for cervical cancer detection. Computers and Electrical Engineering, 123, 110106.

[42] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., Li, Y., & Li, X. (2025). Optimized deep learning model for comprehensive medical image analysis across multiple modalities. Neurocomputing, 619, 129182.

[43] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., & Li, Y. (2025). Optimize brain tumor multiclass classification with manta ray foraging and improved residual block techniques. Multimedia Systems, 31(1), 1-27.

[44] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). AI-Driven Diabetic Retinopathy Diagnosis Enhancement through Image Processing and Salp Swarm Algorithm-Optimized Ensemble Network. arXiv preprint arXiv:2503.14209