International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Classification of Vitiligo using Transfer Learning with New Activation Function Retan

Mrs. Neha Agrawal[1], Dr. Sagaya Aurelia[2]

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

Abstract: In neural networks, activation function is mostly utilised to derive non-linear variations. Based on the input that is provided to a neuron, the activation function determines that neuron's output. Numerous activation functions, including sigmoid, relu, tanh, and others, are utilised in neural networks. To improve the model's accuracy in the proposed study, a novel activation function called retan is employed. When retan was used instead of another activation function, model accuracy rose by about 4 to 5 percent. With the aid of transfer leaning, the proposed activation function was utilised to categorise the vitiligo image. The model with retan activation function offers 92.59 percent validation accuracy and 90.80 percent training accuracy.

Keywords: *Activation Function, CNN, Deep Learning, Transfer Learning, Vitiligo*

1. Introduction

1.1. Motivation:

A frequent acquired skin ailment known as vitiligo causes distinct white spots to appear on the body and is caused by the loss of melanocytes in the epidermis. There are several theories on how vitiligo manifests, but the root cause is still unidentified. It is a specific autoimmune disorder. A pigmentary disorder with an unknown origin, White macules that appear on the skin are a defining feature of vitiligo, brought on by the selective death of melanocytes. Although it affects anywhere from less than 0.1 percent and more than 8 percent of the world's population, the illness only affects roughly one percent of persons in the United States and Europe. The condition is not life-threatening or contagious. It can be stressful or make you feel bad about yourself.

Fig 1. Images for skin disease Vitiligo: Adopted from [13]

In order to diagnose cancer, a variety of variables including shape, size, colour, and texture are extracted using traditional computer vision techniques as a classifier. Artificial intelligence (AI) has now a capability to deal with these issues. The most reputable deep-learning architectures, including recurrent neural networks (RNN), deep neural networks (DNN), and convolutional neural networks (CNN), are used in the medical industry to identify cancer cells. The classification of skin cancer is another area in which these models are effective. Additionally, CNN, a DNN in particular, has already produced outstanding results in this area. The most widely used machine learning algorithm for feature learning and object categorization is the CNN model. The usage of huge data sets in these disciplines allows for more accurate outcomes, which is transfer learning's other advantage.

The output activation function is often either the softmax function for more than two potential output classes or the sigmoid function when there are only two potential output classes. Unfortunately, no testing environment or general test settings have been developed with the intention of evaluating the feasibility of any potential output activation functions except the sigmoid function.

We want to test and assess different output activation functions in the same network in this manuscript. Here, our goal is to determine the ideal combination of output activation function.

1.2. Contribution

The following is a description of our paper's primary contribution:

- We suggest a deep convolutional neural network (DCNN) model with a novel activation function that more precisely categorises Vitiligo even in patients with the earliest stages.
- Our proposed DCNN model outperforms other deep learning (DL) models with a big dataset in terms of accuracy.

[1] *Sr. Consultant, Project Lead, Ceptes Software Pvt. Ltd. Bengaluru (Orcid ID- 0000-0003-0445-0597)*
[2] *Assistant Professor, Department of Computer Science, Christ Deemed to be University, Bengaluru*
** Corresponding Author Email: neha.agrawal@res.christuniversity.in*

• On the same dataset, various activation functions are also assessed in order to compare our suggested DCNN model, and eventually, our proposed model provides the best classification accuracy.

1.3. Organization

The remainder of this paper is illustrated as follows: Introduction to various activation functions and transfer learning are represented in Section 2; related work is represented in Section 3. Data collection and methodology are discussed in Section 4 along with our suggested DCNN model in Section 5. Results and discussion are presented in Section 6, and the conclusion and recommendations for further study are outlined in Section 7.

1.4. Introduction to activation function & Transfer Learning

In recent years, the neural network has emerged as a very popular deep learning method. There are nodes in the neural network that accept input and produce output. Weights are multiplied by the input of each node, and using an activation function, the weights are converted into output. The activation function captures input node nonlinearity. People can obtain the outputs from the inputs through the numerous neurons that constantly work in connection with the weights and bias for an artificial neural network (ANN). The neurons really have no idea of the boundaries of the value because, in the absence of activation functions, the outputs can be anything on the range [+ Inf Inf,]. And no matter what, people always model ANNs as having an input layer, a hidden layer, and an output layer. Whatever the case, there are always several hidden layers. As a result, each layer's output is a linear function of the top layer if activation functions are absent. The outputs of an ANN are a linear combination of the inputs, regardless of how many layers it contains. By computing weighted sum and then adding bias to it, the activation function determines whether or not a neuron should be active. This introduces non-linearity into the neuron's output and allows the outputs to range from [0,1] to [1,1]. The various categories of activation functions include:

I. Sigmoid: A logistic function, a non-linear activation function, is a sigmoid activation function. Its output is between 0 and 1. The output of every neuron will be positive. Its contour resembles a S shape. Although it is monotonic and has a very steep output, small input changes would result in large output changes when the input was close to 0. However, as the input approached either end, the output tended to 0 or 1, responding very little to the input, leading to what are known as "vanishing gradients," which are gradients that are very small or even close to 0. In this scenario, the network won't continue to learn because there won't be much signal going through the neuron to its weights and back to its data repeatedly. Therefore, extra care should be taken to avoid saturation when initialising the weights of sigmoid neurons. Fortunately, there are solutions to this issue, and sigmoid is still widely used in classification problems, particularly in binary classification output layers where the result is either 0 or 1. Since the value for the sigmoid function only ranges from 0 to 1, the result can be easily predicted to be 1 if the value is greater than 0.5 and 0 otherwise.

$$\text{def } sigmoid\ activation\ function(x)$$
$$return\ 1/(1 + numpy.\exp{(-x)}) \quad (1)$$

II. Relu: Rectified Linear Units were created with big positive values in mind, minimising saturation. ReLu has gained a lot of popularity in recent years and is typically used in ANNs' hidden layers, particularly in nearly all Convolutional Neural Networks (CNN) or Deep Learning models (DP). While the linear component (>0, sometimes referred to as a piecewise linear function) makes the patterns in your data easy to interpret, the nonlinearity enables you to identify patterns in your data. ReLu has gained a lot of popularity in recent years and is typically used in ANNs' hidden layers, particularly in nearly all Convolutional Neural Networks (CNN) or Deep Learning models (DP).

$$\text{def } relu\ active\ function(x)$$
$$return\ nupy.array([0, x]).\max{()} \quad (2)$$

III. Tanh: Sigmod functions and tangent hyperbolic functions (tanh) resemble each other visually. Tanh is symmetric in 0 and has a value between -1 and 1. When it comes to the sigmoid, they saturate for a very high number of points at (0, 0), but they are quite sensitive there (positive and negative). They perform better than the sigmoid because of their symmetry. Tanh function is also nonlinear, but unlike Sigmoid, it produces zero-centered results and confines real-valued numbers to the range [1,1], so there is no need to be concerned about activations going off the rails. Despite the fact that its gradient is stronger than Sigmoid's, it nevertheless suffers from the issue of "vanishing gradients". The range [1,1] concentrates the data, improves learning, and causes the mean of the outputs to be zero or extremely close to zero and thus it is usually used in hidden layers of ANNs.

$$\text{def} \tanh\ activation\ function(x):$$
$$return\ 2 * sigmoid\ activation\ function(2 * x) - 1 \quad (3)$$

IV. Leakey Relu: This activation function, in contrast to ReLU, yields a linear slope when a=0.01, enabling neurons to continue to function even when gradient flow is present. It is an improvement to the ReLu function for the "Dying ReLu" problem since its output does not equal 0 for negative inputs. We can easily obtain its derivative

def $leaky\ relu\ activation\ functionx)$;

$return\ 0.01 * x\ if\ x < 0\ else\ x$ (4)

A new activation function called "Retan" was put forth in our suggested study. It consists of relu and tanh together. The tanh function is applied to the input first, and the relu function is then applied to the output. The output is between 0 and 1. As previously discussed, the output of tanh produces a range of -1 to 1, which will be changed to a range of 0 to 1 with the aid of relu. Negative values can also be mapped with the aid of the tanh function, however this suffers from vanishing gradient. To solve the vanishing gradient issue, the relu function was used once more.

def $Retan(x)$:

$return\ relu(\tanh(x))$ (5)

Table 1: Advantage and Disadvantage of different Activation Function

SN	Name of Activation Function	Advantage	Disadvantage
1	Sigmoid	gives smooth gradient	Vanishing gradient problem
2	Relu	computationally efficient	Negative values becomes zero
3	Tanh	Helps in centring data and learning will becomes faster	Vanishing gradient problem
4	Retan	Negative values can also be mapped, efficiency of relu helps in learning the data faster	Efficiency has to be proved for different data set

A machine learning technique called transfer learning uses a model developed for one task as the foundation for a model on another.

Because building neural network models for computer vision and natural language processing challenges takes large computing and time resources, as well as significant skill jumps, it is a typical deep learning strategy to employ pre-trained models as the starting point. The Develop Model Approach and the Pre-Trained Model Approach are two popular methods.

Develop Model Approach

1. Select a source task first. In order to demonstrate your understanding of the ideas learnt while mapping input to output data, you must select a related predictive modelling problem with a big amount of data.
2. Create a source model. The next step is to create a proficient model for this initial assignment. To confirm that some feature learning has taken place, the model must be superior to a naive model.
3. The Model of Reuse Then, a model for the second work of interest can be created using the model fit on the source job. This can include using the entire model or just some parts of it, depending on the modelling technique used.
4. Fine-tune the model. It's possible that the model will need to be modified.

Pre-trained Model Approach

1. First choose a source model. One of the available models is eventually selected as the pre-trained source model. Models from numerous research organisations that are based on enormous and challenging datasets could be added to the pool of candidate models.
2. Reuse Model: You can build a model for a different job that interests you using the model that you previously trained. Depending on the modelling technique employed, this can include using the complete model or simply a portion of it.
3. Adjust the model: Depending on the data for the input-output pair that is available for the relevant work, the model may need to be adjusted or improved.

The convolutional neural network architecture of the Inception family has been improved with Inception-v3. Image processing and object detection features are included in the Inception V3 convolutional neural network.

The convolutional neural network architecture of the Inception series has advanced to the Inception V3. With object detection and image processing, it is an improved version of Inception V1.

As suggested by the name, a Google team developed it and it was originally made available in 2014 as GoogLeNet.

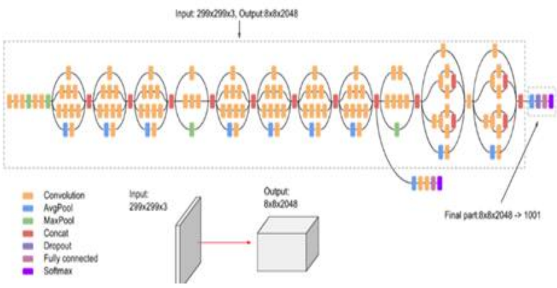


Fig 2. Inception V3 Model Adopted from [14]

2. Related Work

The author suggested using image processing to identify skin conditions. This technique uses a digital image of the affected skin area to identify the kind of disease through image analysis. [1]. The author proposed an improved ReLu Activate function. He claimed that by using the modified activation function SignReLu, the convergence rate is faster, the gradient vanishing problem is successfully solved, and the accuracy of neural network identification is noticeably increased. [2]. By employing a conventional multilayer feedforward network with a locally bounded continuous activation function, the author showed that any continuous function can be estimated to any degree of precision, but only if and only if the network's activation function is not a polynomial [3]. The author showed that estimators based on deep neural networks with ReLU activation function and properly constructed network architecture approach the minimax rates of convergence [4]. This article suggests a universal activation function (UAF) that performs nearly optimally in situations involving categorization, reinforcement learning (RL), and quantification. By adjusting the UAF's parameters, the gradient descent algorithms can evolve the UAF to a suitable activation function for each given problem [5]. This paper contains a quick overview of the many activation functions used in the field of deep learning in addition to analysing the importance of activation functions and improving the functionality of artificial neural networks [6]. In this research, the author explored the effects on neural networks as well as the benefits and drawbacks of numerous activation functions. Additionally, research findings on the MNIST dataset were highlighted in order to compare the effectiveness of various activation functions [7]. The author presented a novel piecewise activation function in accordance to the design philosophy of the activation function in the CNN model. In this study, five commonly used activation functions have been analysed and compared [8]. The majority of the AFs utilised in DL were collated by the author, who also discusses current trends in their use and application in real-world deep learning deployments in comparison to SOTA research findings. This compilation assists in selecting the most acceptable and relevant AF for a certain application that is prepared for deployment [9]. Based on the significance of an activation function, nine new activation functions built from combinations of traditional functions like ReLU and sigmoid were given in this research. Additionally, a study on the impacts of activation functions on a CNN's performance was given [10]. This study had two goals: first, it explored a variety of hyper-parameter configurations to see how they affected DNN performance and whether they could be used as a starting point for tuning DNNs. Second, it compared the performance of DNNs to Naive Bayes, k-nearest neighbour, random forest, and support vector machines, which are widely used in the field of cheminformatics [11]. A parametric funnel rectified exponential unit was suggested by the author (PFREU). PFREU encourages competition between spatial conditions, nonlinear mapping, and linear mapping. They tested their methodology using four datasets of varying sizes, and the experimental outcomes is demonstrated its superiority [12].

3. Data Description:

In order to create our dataset, we gathered photos from the web database Dermnet that are particular to skin conditions as well as from the ayurvedic clinic Viswam Ayurveda. 150 photos of vitiligo and 120 images of other diseases are included in the database. The clinic provided 79 photographs, and a total of 71 Vitiligo images were downloaded. A total of 120 photos of different skin conditions, such as actinic cheilitis-sq-cell-lip, acne pitted scars, and atypical mycobacterium, were retrieved. In the suggested work, 270 images altogether were used.

4. Methodology

This section describes our system's technique for the detection, extraction, and classification of photographs of the vitiligo disorder. The technique will be very helpful in detecting Vitiligo. The pre-processing, model building, and classification modules can be used to break down the entire architecture into smaller units. Figure 3 displays the system's block diagram.

Fig. 3. Block Diagram of Proposed System

All codes were written in python. A different library like Keras, matplotlib were used.

The following steps can be used to discuss the methodology:

Pre-processing: High performance systems for detecting skin diseases must overcome several major challenges. Examples include building a database and standardising image dimensions. Drives with images loaded and training folders with images and class material loaded. The procedure for image scaling is explained in the section that follows.

Image Resizing: An input image is either made larger or smaller to address the issue of various image sizes in the database. The same amount of features will be obtained from all images by standardising the image size. Additionally, shrinking the image speeds up the system by reducing processing time. In order to optimise and assess the suggested DCNN model, In our research, we split the dataset into three sets for training, validation, and testing. 250 photos make up the training dataset, 54 images the validation dataset, and 66 images the test dataset. The terms "Non-Vitiligo" and "Vitiligo" were used as classifications.

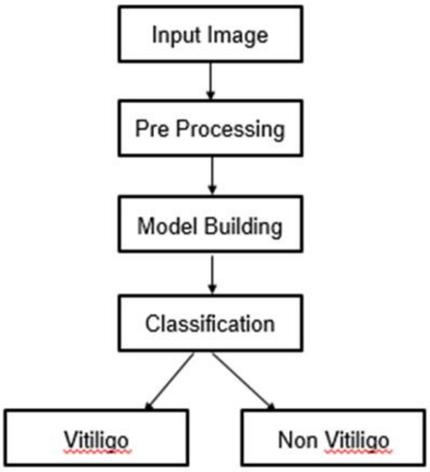


Fig. 4. Resized images of Vitiligo

Data Augmentation:

Data augmentation is the technique of adding slightly modified copies of current data without actually acquiring new data from training data already there. The size of the training dataset can be intentionally increased by data warping or oversampling, or it can help the model avoid over-fitting by addressing the fundamental cause of the issue. To avoid this over-fitting, we added more data using a variety of augmentation settings, such as rotation, random cropping, and mirroring. Table 2 provides the parameters we employed for our dataset's data augmentation.

Table 2: Data Augmentation Parameters

SN	Parameter	Parameter Value	Action
1	Rotation Range	40	picture rotation from -40 to 40 to produce input
2	Width Shift Range	0.2	Image is moved by 0.2 pixels in the horizontal direction.
3	Height Shift Range	0.2	Image is moved by 0.2 pixels in the vertical direction.
4	Shear Range	0.2	Increase the image's slant angle by 0.2 degrees.
5	Zoom Range	0.2	zoom in or out by 0.2 from the middle
6	Horizontal Flip	TRUE	Randomly reverses the horizontal direction of the images.

Model Construction: Model was constructed with the combination of different layers. Some layers for the model is customized. Model construction is defined in following steps:

The Inception V3 model with pre-trained weights was downloaded. An input layer, convolutional layer, batch normalisation, and activation layer make up the V3 inception model. The top six layers were left out. The V3 model's layers were all frozen, and weights were stored.

A conv2D layer, a batch normalisation layer, an activation layer with a retan activation function, and additional layers are combined to create the innovative Conv2D BN Layer. When very deep neural networks are trained using the batch normalisation technique, the contributions to a layer for each mini-batch are normalised. As a result, the learning process is stabilised and a smaller number of training epochs are required to build deep neural networks.

Once more, the Inception layer was added. Structure of Inception layer is given in Figure 4. It consists of three layes. First layer is the parallel layer of 4 layers Branch 1x1, Branch 3x3, Branch 5x5, Branch pool layer. Branch 1x1 is the con2D_BN Layer with 1x1 filters and retan activation function. Similarly, Branch 3x3 is the con2D_BN Layer with 3x3 filters and Branch 5x5 is the con2D_BN Layer with 5x5 filters. Combining two layers, the branch pool layer is composed of a max pooling layer with a 3 pool size and a con2D BN layer with a 1x1 filter. The variety in input for successive layers is increased by concatenating feature-maps learned by several layers, which also boosts efficiency. The gradients reach the bottom layers more effectively and cut down on training error since the lower layers have a shorter path to the top layers.

Second layer in inception layer is again parallel layer of 3 layers Branch 3x3, Branch 5x5 and Branch pool. Third layer consist of only a single layer

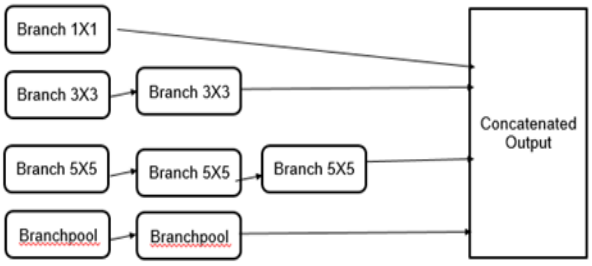


Fig. 5. Block Diagram of Inception Layer

The structure of the model is shown below in the Figure 5. Model includes inception V3 model, 2 Conv2d_BN model, two inception layer, one global average pooling layer followed by dense layer.

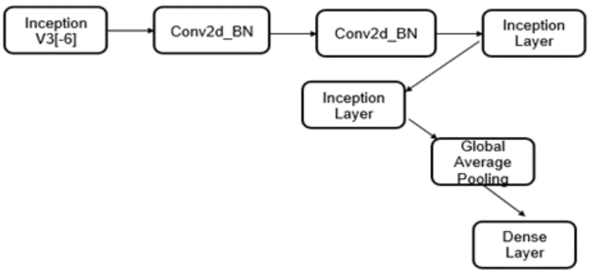


Fig. 6. Block Diagram of Model Constructed

Model Training and Validation: In our suggested DCNN model, numerous parameters are employed to fine-tune it. The main ones we employed in our experiment, together with the values we chose for each, are listed below. This design is contrasted with a number of various configuration types and previously tested deep neural network models.

- For quicker training, the pooling layer (MaxPooling2D) is employed to minimise the size of the input images.
- Batch size (50): the quantity of photos that are processed in a single iteration.
- Initial learning rate (0.001): this rate determines how quickly learning will begin.
- Optimizer (Adam): When training DL models, Adam is a stochastic gradient descent substitution optimization strategy that lowers the loss function. For our investigation, we used the Standard Gradient Descent algorithm with Momentum = 0.999.
- A binary classification problem is built from the dataset using the loss function, also known as binary cross entropy.
- Number of epochs (30): The quantity of times the DNN receives the input dataset.

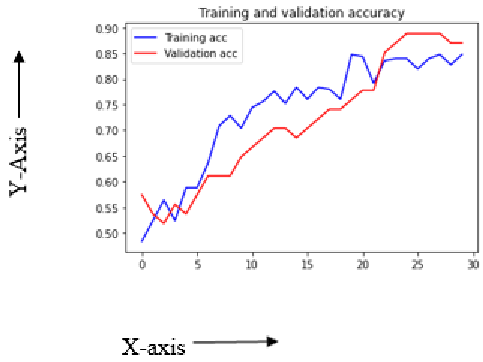
Classification: Computer vision techniques include classification. After extracting features, the function of classification is to categorise the image by newly constructed model.

5. Result & Discussions

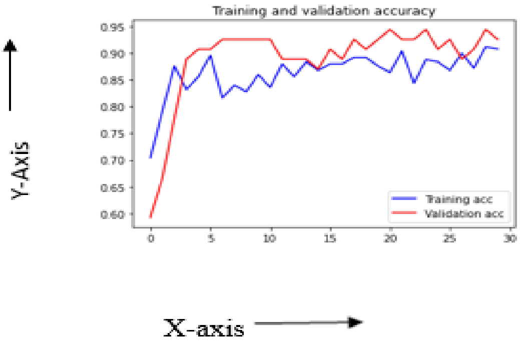
The system is implemented in Python using keras, Numpy and Matplotlib. We ran the experiment in this study using our suggested DCNN model with various activation functions. The result of different activation function was displayed in table 3. We made use of a setup with a 2.10 GHz Intel Core i3 processor and 4 GB of RAM. The training and validation accuracy results are shown for proposed model in Figure 6. Initially, the input images are pre-processed, then training and test set were constructed. Finally, classification is performed using the model constructed above. After training, the model 90.80 % training accuracy and 92.59 % validation accuracy were achieved. In this study, 270 skin images from the Internet were also used by a number of individuals with dermatological diseases. The proposed approach has a 90% accuracy rate for detecting vitiligo skin disorders. The system is efficient. Additionally, early discovery of vitiligo skin growth, particularly in individuals who are not accepted by doctors, can greatly motivate them to receive treatment on time.

tanh activation

Relu activation



The modified tanh activation



Retan activation function

Modified Relu

Modified Relu with Sigmoid

In Table 3 comparative performance of same network with some different activation function is presented. It can be seen that retan activation function is outperformed out of all other activation function. It can be seen that some network suffers from under fitting where as some suffers from overfitting. . Result from the table shows that with Relu activation function data was over fitted whereas when modified sigmoid was used then the data was under fitted. Tanh, modified Tanh and modified relu training and validation accuracy were less.

Table 3: Performance Analysis of different activation function

SN	Activation Function	Training Accuracy	Validation Accuracy
1	Tanh	87.25	85.29
2	Relu	85.91	94.12
3	Modified Tanh	84.80	87.04
4	Modifued Relu	89.20	85.19
5	Modified Relu with Sigmoid	83.60	61.11
6	Retan	90.80	92.59

Table 4: Performance Analysis of different activation function on combination of different layers in neural network

SN	**Name**	**Description**	**Accuracy Sigmoid**	**Accuracy Relu**	**Accuracy tanh**	**Accuracy -Retan**
Model 1	Transfer Learning with Single layer Network	1 input, 1output 1 hidden layer	89.75	91.28	90.06	92.34
Model 2	Transfer Learning with Change Layer1	1 Convolution Layer 2 Inception Layer Global Average Pooling Layer Dense Layer	89.9	96.30	90.01	90.00

Model 3	Transfer Learning with change Layer2	3 Convolution Layer Global Average Pooling Layer Dense Layer	78.29	81.48	80.02	83.60
Model 4	Transfer Learning with change Layer3	2 Convolution Layer 1 Inception Layer Global Average Pooling Layer Dense Layer	87.21	90.74	89.92	94.00
Model 5	Transfer Learning with change Layer4	3 Convolution Layer Activation Function Global Average Pooling Layer Dense Layer	70.07	74.07	71.67	79.20
Model 6	Transfer Learning with change Layer5	1 Convolution Layer 1 Inception Layer Global Average Pooling Layer Dense Layer	86.71	92.59	88.47	90.80
Model 7	Transfer Learning with change Layer6	2 Convolution Layer 2 Inception Layer Global Average Pooling Layer Dense Layer	82.35	88.89	85.56	89.20

6. Conclusion & Future work

The early discovery of skin disorders is a vital initial step in reducing mortality rates, disease dissemination, and the progression of skin diseases. Clinical techniques that are costly and time-consuming are needed for skin disease diagnosis. Image processing methods are useful in the early phases of creating an automated dermatological screening system. Extraction of characteristics plays a significant role in the classification of skin diseases. In this study, a custom layer and a pre trained convolutional neural network (Inception V3) were used to create the detection approach. In our network, a new activation function called "retan" performed relatively well. The activation functions of relu and tanh are combined to form the activation function of retin. The benefit of tanh for mapping negative values as well as dealing with vanishing values is provided by the new activation function.

With the use of this model, other skin disease kinds can be categorised in the future. In the future, it may be possible to diagnose additional skin conditions.

Acknowledgment

The authors wish to thank Dr. Shipra from Vishwam Ayurveda and Dr. Meenakshi from Kushal Skin Clinic Bengaluru for their medical expertise support in data collection and result validation

Author contributions

Author one conducted the result and analyze the result in the supervision of Author B. Paper was written by Author A whereas full guidance was provided by Author B.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Nawal Soliman ALKolifi ALEnezi, A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science, Volume 163,2019, Pages 85-92, ISSN 1877-0509,

[2] Guifang Lin, Wei Shen, Research on convolutional neural network based on improved Relu piecewise activation function, Procedia Computer Science, Volume 131, 2018, Pages 977-984, ISSN 1877-0509.

[3] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, Shimon Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks, Volume 6, Issue 6, 1993, Pages 861-867, ISSN 0893-6080, https://doi.org/10.1016/S0893-6080(05)80131-5.

[4] Johannes Schmidt-Hieber. "Nonparametric regression using deep neural networks with ReLU activation function." Ann. Statist. 48 (4) 1875 - 1897, August 2020.

[5] Yuen, B., Hoang, M.T., Dong, X. *et al.* Universal activation function for machine learning. *Sci Rep* **11,** 18757 (2021).

[6] Siddharth Sharma, Simone Sharma, Anidhya Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS" International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316

[7] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1836-1841.

[8] Yingying Wang, Yibin Li, Yong Song, and Xuewen Rong, "The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition" Appl. Sci. 2020, 10, 1897.

[9] Chigozie E. Nwankpa, Winifred l.Jonah, Anthony Gachagan, Stephen Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning". 124 - 133. 2nd International Conference on Computational Sciences and Technology, Jamshoro, Pakistan

[10] M. Guevara, V. Cruz, O. Vergara, M. Nandayapa, H. Ochoa, H. Sossa, "Study of the Effect of Combining Activation Functions ina Convolutional Neural Network" IEEE LATIN AMERICA TRANSACTIONS, VOL. 19, NO. 5, MAY 2021

[11] Koutsoukas, A., Monaghan, K.J., Li, X. et al. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J Cheminform 9, 42 (2017).

[12] Yao Ying, Ningbo Zhang, Ping He, Silong Peng, "Improving Convolutional Neural Networks with Competitive Activation Function", Security and Communication Networks, vol. 2021, Article ID 1933490, 9 pages, 2021.

[13] https://www.kaggle.com/

[14] https://iq.opengenus.org/inception-v3-model-architecture/