iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.53465

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Detection of Various Skin Diseases Using CNN in Machine Learning

Tushar Gupta[1], Sunil Waghmode[2], Anmol Kamble[3], Atharv Chinchkar[4]
Department of Computer Engineering, PVG's COET &GKPIM, Affiliated to Savitribai Phule Pune University, India

Abstract: ***Skin diseases are the most common diseases. Skin diseases may be caused by bacteria, fungal infection, allergy, or viruses, etc. The term Dermatology means the branch of bioscience that deals with diagnosing and treatment of skin based disorders.***
The dermatologic disorders vary geographically and seasonally because of humidness, alternative environmental factors and temperature.
Human skin is one amongst the most unpredictable to analyze and mechanically synthesize because of its quality of unevenness, tone, presence of hair and alternative mitigating options. Patients generally ignore early symptoms because there aren't enough medical facilities and tools in remote regions, which could make the situation worse over time. Consequently, there is a growing need for an accurate, automatic method for detecting skin diseases. Thus, we have developed a Deep Learning model to differentiate between Healthy Skin and Skin Suffering from a Disease and also Classify Skin Diseases into its seven main classes like Melanoma, Basal cell Carcinoma, MelanocyticNevi, Benign keratosis-like lesions, ActinicKeratoses, Vascular lesion and Dermatofibroma.
Deep Learning is a subset of machine learning that, in contrast to machine learning, employs a large dataset and, as a result, drastically reduces the number of classifiers. The model self-learns, categorizes the provided data into levels of prediction, and provides accurate results in a very short amount of time, fostering and supporting the advancement of dermatology. Convolutional Neural Network (CNN), one of the most popular algorithms for image classification, is the algorithm that we will use. We have integrated this model with a web application, where users could upload images of the infected skin and get output in the form of a report.
Keywords: Deep Learning, Convolutional Neural Network CNN, Epoch, Neurons, Image Processing, Machine Learning, Dermatological Images, Testing, Training.

I. INTRODUCTION

Artificial Intelligence have brought revolution in all fields including the healthcare industry. In recent few years number of people getting affected due to skin diseases have been increased massively. There are various skin abnormalities that needs to be treated and diagnosed at an early stage to stop it from spreading. One of the major reason for increase in such abnormalities is due to direct and excessive exposure to UV Radiation.

A. Existing Technologies

ANN - An Artificial Neural Network (ANN) is a statistical nonlinear predictive modeling technique used to capture intricate relationships between input and output variables. Its structure is inspired by the biological structure of neurons in the human brain. Using Artificial Neural Network, accuracy obtained in various researches is 80% [2]. ANNs consist of three types of computation nodes, and they learn through a process called back-propagation.

The training of an ANN involves using both labeled (trained) and unlabeled (untrained) datasets, which helps achieve accuracy through supervised and unsupervised learning approaches. Different types of neural network architectures, such as feed-forward and backpropagation, utilize the information in a specific manner.

SVM - Support Vector Machines (SVM) is a supervised classifier that aims to create an optimal hyperplane in an n-dimensional space to effectively separate data points into two distinct categories [2]. Selecting an appropriate kernel function in SVM can be challenging. Moreover, training SVM models can be time-consuming, especially for large datasets. One drawback of SVM is that the resulting model may not be easily interpretable, making it challenging to make small adjustments or fine-tune its parameters. SVM has an accuracy of 89% [7]. However, in terms of performance, SVMs often outperform Artificial Neural Networks (ANNs) [3].

II. DATASET

The HAM-10000 dataset is used which contains 10,015 dermatoscopic images obtained from various populations and captured using different imaging techniques. This dataset serves as a valuable resource for academic machine learning research. It encompasses a wide range of diagnostic categories related to pigmented lesions, including actinic keratoses and intraepithelial carcinoma disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv), and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas, and hemorrhage, vasc). The dataset also contains instances where lesions have multiple images, which can be tracked using the lesion_id column provided in the HAM10000_metadata file.

III. METHODOLOGY

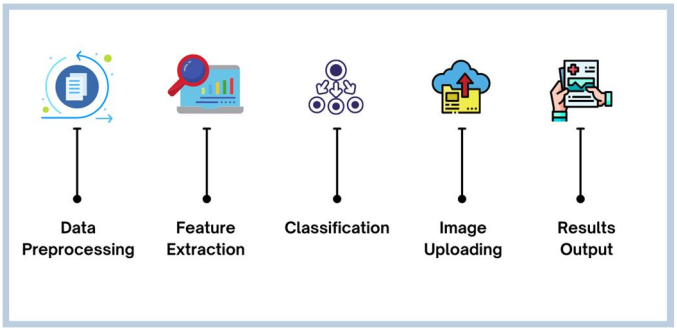


Fig. 1 System Implementation Steps

A. Data Pre-processing

Achieving high performance skin disease detection systems requires overcoming some major difficulties such as creating a database and unifying image dimensions. It includes the following steps.

1) *Data Gathering:* We have gathered the images from the data set HAM10000.
2) *Data Cleaning:* After gathering the data, data cleaning is done to bring all images into the same pixel size, visibility, filling missing values, removing noisy data, etc.
3) *Label Encoding:* The cleaned data is renamed and labeled based on the classes such as MelanocyticNevi, Melanoma, Benign keratosis-like lesions, Basal cell carcinoma, ActinicKeratoses, Vascular lesions, Dermatofibroma, Healthy Skin.
4) *Data Transformation:* It involves transforming data by cleaning and encoded to make it ready for the further processes..

B. Feature Extraction

In this step the preprocessed image will be further used for feature extraction. Here the important features will be extracted using CNN with the help of Filtering, Pooling Layers, etc. Features such as symmetricity, border and color are extracted.

C. Classification

Here after preprocessing and feature extraction on the data is performed we will be able to classify the images from the data set and the input provided by the user by mapping. Based on the classification, the user will get the output as the name of the disease.

D. Image Uploading

The user can upload an image of the infected part of skin

E. Results Output

The final result that is predicated by the model after uploading an image is produced along with the detected disease name if any.

IV. TRAINING

We have divided the data set images into training set, testing set and validation. The split can be in any ratio and the batch size and number of epochs are decided.

V. MODEL BUILDING

CNN is used in building our model. Convolutional Neural Network (CNN or ConvNet) is a type of deep neural network that autonomously learns from data and divides it into hierarchical levels of predictions, providing accurate results in a short period of time. CNNs are specifically designed for image classification tasks and utilize a combination of convolutional and pooling layers, followed by fully connected layers, similar to a multilayer neural network [2]. What sets CNNs apart from other algorithms is their ability to effectively classify images by leveraging key features such as sparse connectivity, shared weights, and pooling techniques. Furthermore, the utilization of Graphical Processing Units (GPUs) has significantly reduced the training time for deep learning methods. This, along with the availability of large labeled datasets and pre-trained networks, has contributed to the widespread adoption of CNNs in image classification tasks.

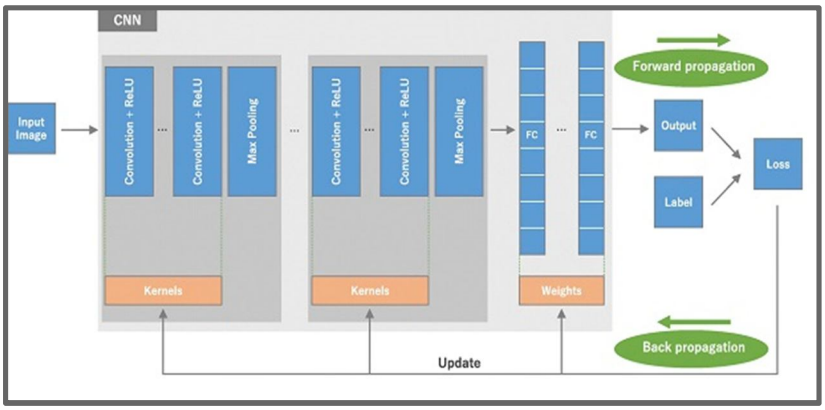


Fig. 2 CNN Architecture

A. Model Explanation

We utilized the Keras Sequential API to build our neural network. This API allows us to add one layer at a time, starting from the input. Our initial layer is a Conv2D layer with 4 hidden layers using different filters. Each filter learns specific features by transforming a portion of the image, defined by the kernel size.

The transformed images are known as filter maps. To downsample the data and reduce overfitting, we employ a pooling layer that is MaxPooling. This layer selects the maximum value among sets of neighboring pixels. It helps in reducing the dimensionality of the data while retaining important features and does de-noising of the data which concludes MaxPooling outperforms average pooling significantly. By combining convolutional and pooling layers, our CNN can effectively combine local features and learn global features from the input data.

To introduce non-linearity to the network, we use the ReLU activation function. It helps in capturing complex patterns and improving the network's ability to learn. To prevent overfitting, we incorporate dropout regularization. This technique randomly ignores a proportion of nodes in each layer during training by setting their weights to zero. By doing so, it encourages the network to generalize better. Next, we have a Fully Connected Layer which consists of Flatten layer, converts the final feature maps into a 1D vector.

This flattening step is necessary to connect fully connected layers, which can then utilize the learned features from previous convolutional layers. In the last layer, we use the Dense() function where activation function SoftMax is used as we are dealing with multiple classes, which produces the output distribution indicating the likelihood of each category. This layer provides the final predictions of our network. After adding the layers, we need to define a scoring function, a loss function, and an optimization algorithm.

For the loss function, we choose binary cross-entropy, which measures the error rate between the observed and predicted labels. To optimize the model's performance, we use the Adam optimizer. Adam combines the advantages of various optimization algorithms and is widely used in machine learning models. For evaluating the model's performance, we employ the accuracy metric, which measures the percentage of correctly classified samples. Learning rate (LR) is another crucial factor. It determines the step size at each iteration during training. We apply an annealing method called ReduceLROnPlateau, which automatically decreases the learning rate to reach the global minimum of the loss function gradually. This approach helps in achieving minimal loss and better convergence.

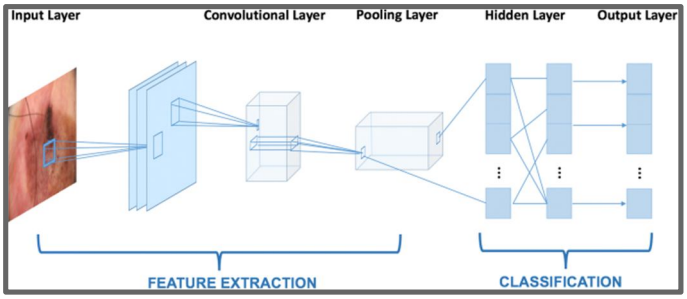


Fig. 3 System Overview

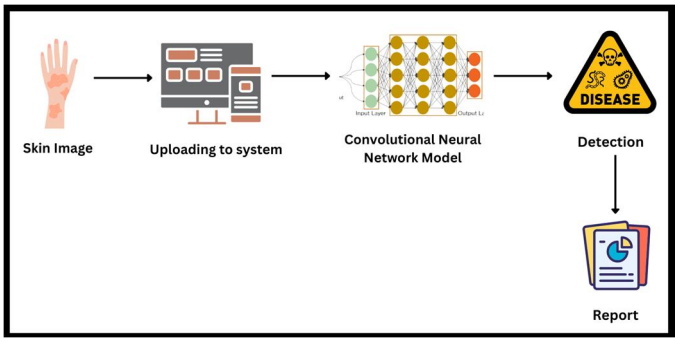


Fig. 4 Application Overview

VI. RESULT AND DISCUSSION

The number of Epochs used in our are 50 with a batch size of 32. The testing and validation accuracy of our model is 98.57% and 98.68% with the loss of 17.33% and 17.11% respectively.

```
from sklearn.metrics import confusion_matrix , classification_report

target_names = [f"{classes[i]}" for i in range(8)]
print(classification_report(y_test , y_pred , target_names =target_names ))
```

[26]

	precision	recall	f1-score	support
('akiec', 'Actinic keratoses and intraepithelial carcinomae')	1.00	1.00	1.00	1673
('bcc', ' basal cell carcinoma')	0.99	1.00	0.99	1765
('bkl', 'benign keratosis-like lesions')	0.97	1.00	0.98	1651
('df', 'dermatofibroma')	1.00	1.00	1.00	1684
('nv', ' melanocytic nevi')	0.99	0.89	0.94	1660
('vasc', 'vascular lesions')	1.00	1.00	1.00	1663
('mel', 'melanoma')	0.95	1.00	0.97	1677
('Normal', 'Healthy Skin')	1.00	1.00	1.00	1637
micro avg	0.99	0.99	0.99	13410
macro avg	0.99	0.99	0.99	13410
weighted avg	0.99	0.99	0.99	13410
samples avg	0.99	0.99	0.99	13410

Fig. 5 Classification Matrix

Sr No.	Parameters	Testing	Validation
1.	Accuracy	98.57%	98.68%
2.	Loss	17.33%	17.11%

Table 1: Proposed Model Evaluation

```
Epoch 46/50
1258/1258 [==============================] - 170s 135ms/step - loss: 1.7779e-11 - accuracy: 1.0000 - val_loss: 0.1670 - val_accuracy: 0.9855 - lr: 1.0000e-04
Epoch 47/50
1258/1258 [==============================] - 161s 128ms/step - loss: 1.1853e-11 - accuracy: 1.0000 - val_loss: 0.1693 - val_accuracy: 0.9855 - lr: 1.0000e-04
Epoch 48/50
1258/1258 [==============================] - 160s 127ms/step - loss: 8.8896e-12 - accuracy: 1.0000 - val_loss: 0.1716 - val_accuracy: 0.9855 - lr: 1.0000e-04
Epoch 49/50
1258/1258 [==============================] - 160s 127ms/step - loss: 8.8896e-12 - accuracy: 1.0000 - val_loss: 0.1712 - val_accuracy: 0.9857 - lr: 1.0000e-04
Epoch 50/50
1258/1258 [==============================] - 157s 125ms/step - loss: 1.1853e-11 - accuracy: 1.0000 - val_loss: 0.1733 - val_accuracy: 0.9857 - lr: 1.0000e-04

[22] model.evaluate(X_test, y_test)

420/420 [==============================] - 17s 41ms/step - loss: 0.1733 - accuracy: 0.9857

[0.173274427652359, 0.9856823086738586]
```

Fig 6. Testing Accuracy

```
[ ] from sklearn.metrics import accuracy_score

    # Assuming you have true labels (y_true) and predicted labels (y_pred) for the validation set
    validation_accuracy = accuracy_score(y_test, y_pred)

    print("Validation Accuracy:", validation_accuracy)

    Validation Accuracy: 0.9868754660700969
```

Fig 7. Validation Accuracy

```
[18] model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        896

 conv2d_1 (Conv2D)           (None, 28, 28, 32)        9248

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 14, 14, 64)        18496

 conv2d_3 (Conv2D)           (None, 14, 14, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 conv2d_4 (Conv2D)           (None, 7, 7, 128)         73856

 conv2d_5 (Conv2D)           (None, 7, 7, 128)         147584

 max_pooling2d_2 (MaxPooling  (None, 3, 3, 128)        0
 2D)

...
Total params: 1,206,184
Trainable params: 1,206,184
Non-trainable params: 0
```

Fig 8. Model Summary

Fig 9. Skin Images

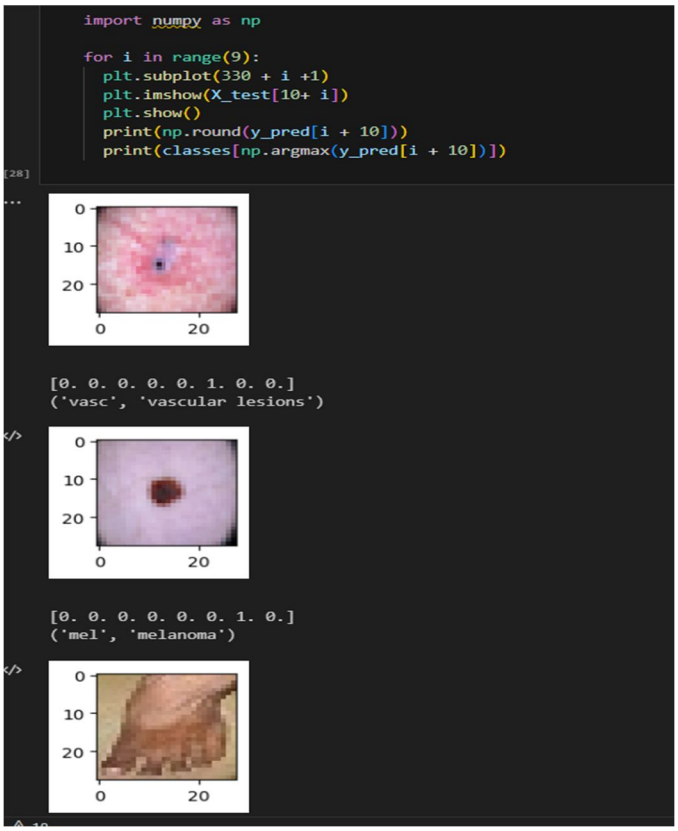


Fig 10. Prediction of Classes.

VII. FUTURE WORK

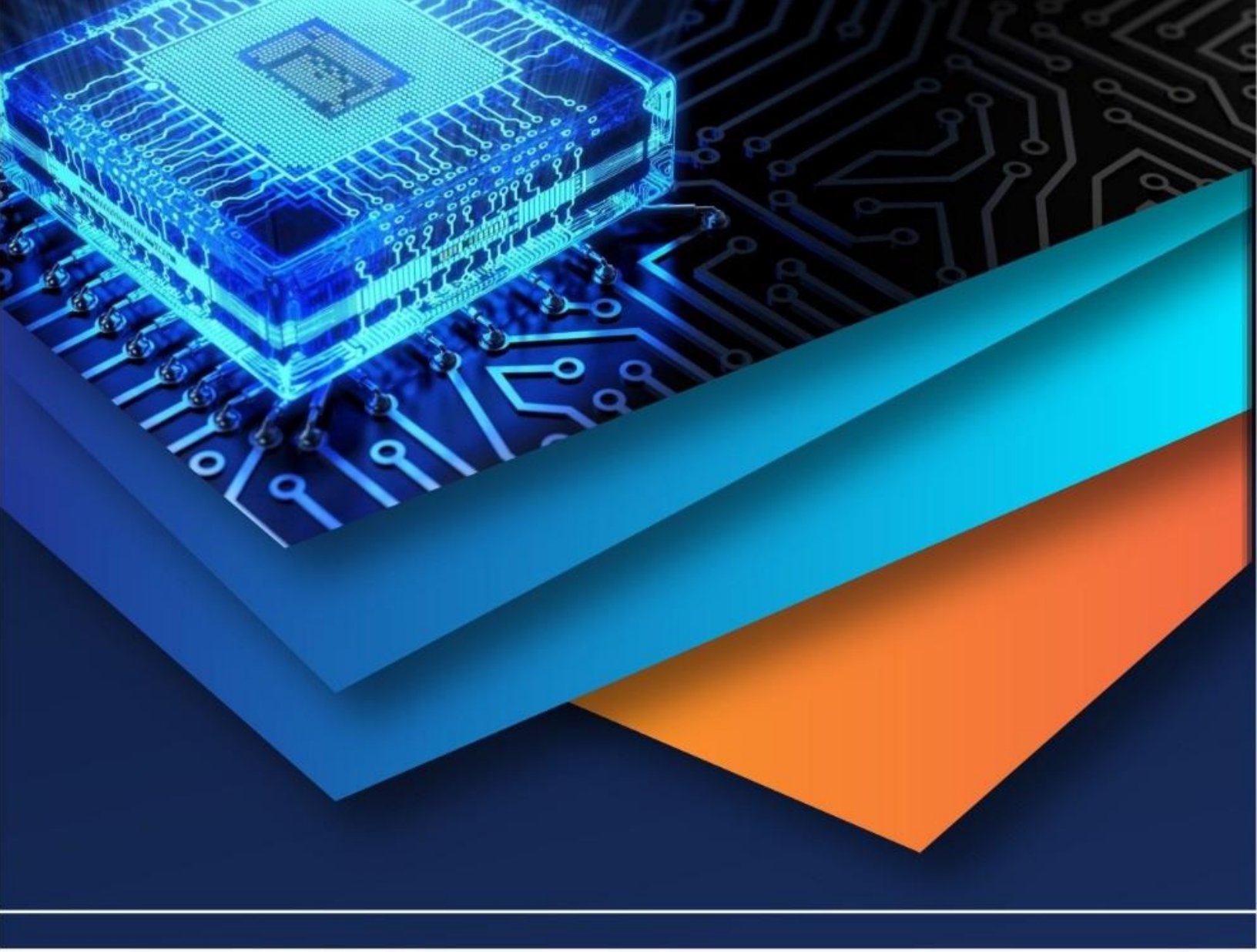
1) *Creating Region Wise Dataset:* Currently used dataset contains a generalized data, but the complexity of skin and its diseases vary from region to region. There is a need to create data based on geographical regions according to skin textures and climatic conditions of that region and its complexity..
2) *Detection of Disease Stage and Complexity:* This model only predicts the disease name, but the stage and complexity of the disease is also important. Further a model can be developed which will find the complexity and the stage of detected disease..

VIII. CONCLUSION

Skin Diseases are ranked fourth most common cause of human illness, but many still do not consult doctors mostly in rural areas. We have proposed a robust and automated method for the detection of dermatological diseases. Treatments for skin are more effective when found early. With the help of this system early detection and diagnosis can be done. The result report generated by the system will be useful for doctors to get an idea and start with further line of treatment.

REFERENCES

[1] Viswanatha Reddy Allugunt, A machine learning model for skin disease classification using Artificial neural network 2021
[2] Jana, E., Subban, R., & Saraswathi, S. (2017). Research on Skin Cancer Cell Detection Using Image Processing. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). doi:10.1109/iccic.2017.8524554.
[3] Mhaske, H. R., & Phalke, D. A. (2013). Melanoma skin cancer detection and classification based on supervised and unsupervised learning. 2013 International Conference on Circuits, Controls and Communications (CCUBE). doi:10.1109/ccube.2013.6718539.
[4] Kritika Sujay Rao,Pooja Suresh Yelkar,Omkar Narayan PisE, Dr. Swapna Borde Skin Disease Detection using Machine Learning ; International Journal of Engineering Research Technology (IJERT)2021 DOI : 10.17577/IJERTCONV9IS03016
[5] Rashmi Patil, Sreepathi Bellary. Machine learning approach in melanoma cancer stage detection, Journal of King Saud University - Computer and Information Sciences, 2020, ISSN:1319-1578, https://doi.org/10.1016/j.jksuci.2020.09.002.
[6] Nawal Soliman ALKolifi ALEnezi- A Method Of Skin Disease Detection Using Image Processing And Machine Learning. doi.org/10.1016/j.procs.2019.12.090
[7] Ilker Ali OZKAN, Murat KOKLU, Skin Lesion Classification using Machine Learning Algorithms. https://doi.org/10.18201/ijisae.20175344201



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY