Skin Disease Detection Using Image Processing

Project Report submitted in the partial fulfillment

of

B-Tech integrated

In

Computer Engineering

By

Vrucha Joshi (C152)

Dhriti Jangla (C154)

Ahaan Mehta (C162)

Sachi Mane (C173)

Under the supervision of

Prof. Vipul Gohil

Assistant Professor

Electronics and Telecommunication

MPSTME

Skin Di

SVKM's NMIMS University

(Deemed-to-be University)

SVKM'S NMIMS Deemed to be UNIVERSITY

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING (MPSTME)

Vile Parle (W),Mumbai-56

(2023-2024)

CERTIFICATE

SVKM'S NMIMS Deemed to be UNIVERSITY

This is to certify that the project entitled **"Skin Disease Detection using Image Processing"**, has been done by **Mr. Ahaan Mehta , Ms. Sachi Mane, Ms. Dhriti Jangla, Ms. Vrucha Joshi** under my guidance and supervision & has been submitted in partial fulfillment of the degree of B.Tech Integrated in Computer Engineering of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

04/04/23

Project mentor (name and Signature) Prof. Vipul Gohil

(Internal Guide)

Date 4/4/23

Place: Mumbai

04/04/24

Examinar(Name and signature) Dr. Avinash Tandle

(HOD)(name and signature) Prof. Abhay Kolhe

ACKNOWLEDGEMENT

Student Names

NAME	ROLL NO.	SAP ID
Vrucha Joshi	C152	70322100204
Dhriti Jangla	C154	70322100173
Ahaan Mehta	C162	70322100046
Sachi Mane	C173	70322100086

ABSTRACT

Using image processing and machine learning technology, the diagnosis of skin diseases has been already proven to have promising prospects of helping doctors make better dermatology examinations. The possibilities for extension of this model entail several areas of concern for development and improvement. Extending the training data set applying to the model creates the possibility of including a multitude of skin types. It also helps to improve the model's capability of perceiving and recognizing a broader variety of conditions. Alongside advanced imaging processing techniques, the ability to locate more accurate features of the skin lesion images and an improvement of accuracy in skin disease diagnosis and classification can be explored. Integration of one or the other machine learning algorithms like ensemble learning or transfer learning can intensify the sharing of the model's performance and acceptance. Moreover, the model will be able to sync with telemedicine platforms and mobile applications, bringing the possibility of remote diagnosis and observation of skin diseases, which in turn will result in the broadening of reach and penetration of healthcare services. These advances could be of great importance for dermatology due to the possibility of finding reliable and effortless data processing tools to differentiate skin diseases based on image processing techniques.

TABLE OF CONTENTS

List of the figures

List of tables

Abbreviations

Sr.No	Abbreviation	Full-Form
1	CNN	Convolutional Neural Network
2	API	API
3	PIL	Python Image Library.
4	VGG	Visual Geometry Group
5	GPU	Graphics Processing Unit
6	TPU	Tensor Processing Units

Chapter 1
INTRODUCTION

1.1 Background of the project topic

Skin diseases are more common than any other disease. Fungi, bacteria, allergies, viruses, etc can cause skin infections. They are common health problems worldwide. In addition, in severe cases, it can lead to skin cancer. Therefore, diagnosing skin diseases using clinical images is one of the most difficult tasks in medical image analysis. Furthermore, the diagnosis of skin diseases is a complex and time-consuming task performed by medical professionals. Therefore, image processing technology is useful to establish an automated dermatological examination system at the beginning[1].

The diagnosis of skin diseases using image processing is a field that combines computer vision, image analysis, and machine learning technologies to develop automated systems for diagnosing and classifying skin diseases. The goal is to provide a faster and more accurate examination of skin conditions, which would be costly and time-consuming if done manually[2].

Feature extraction plays an important role in skin disease classification. Computer vision has the potential to detect skin diseases using various technologies. This research will contribute to skin disease research. A method for the diagnosis of skin diseases based on image processing was proposed. This method takes digital images of the infected skin area and then analyzes the images to identify the type of infection. It's simple, fast, and doesn't require any expensive equipment other than a computer. Finally, results such as disease type, prevalence, and severity are shown to the user. The system detected eight types of skin diseases with a good amount of accuracy[1][2].

1.2 Motivation and scope of the model

Skin disease can affect a person's health and well-being. The motivation behind reporting a skin disease detection model using image processing is to address the need for better and more accurate methods to diagnose and classify skin diseases. Traditional diagnostic methods are time-consuming, expensive, and require specialized knowledge.

The report covers topics such as advances in image analysis, image extraction, and medical technology that have contributed to the development of automated skin disease diagnosis systems. Only a computer is needed, which shows the simplicity and accessibility of the proposed method, suitable for healthcare professionals in different situations[3].

1.3 Problem statement

Skin diseases are common health problems that affect people's lives. Early detection and correct diagnosis of skin diseases are very important for effective treatment and prevention of complications. However, manual analysis by medical professionals is time-consuming and subjective. Therefore, an automated system is needed to support the detection and classification of skin diseases. This project aims to develop a skin disease detection method using image processing technology. The system takes digital photos of the affected skin area for input and uses image analysis to identify the type of infection. The system should be simple, fast, and cost-effective, with just a camera and a computer[4].

1.4 Salient contribution

Reports on skin disease detection using image processing have made many important contributions to the field of dermatology. First, we propose a skin disease detection method based on image processing, which consists of taking a digital image of the affected skin area and identifying the type of disease through image analysis. This method is simple, fast and affordable because it does not require expensive equipment other than a camera and a computer. Secondly, you have the opportunity to perform automatic tests, which can detect skin diseases early. Using computer vision algorithms, this automated system can improve diagnostic accuracy and efficiency, leading to more accurate treatment and better patient outcomes. These gifts advance the field of dermatology by providing effective and efficient solutions for the diagnosis of skin diseases[5].

1.5 Organization of the report

A brief construction of a skin disease detection report using image processing

1. Abstract: A summary of the report, including study objectives, research questions, methods used, key findings, and conclusions.

2. Introduction: Emphasize the importance of early detection and accurate diagnosis of skin diseases.

3. Literature review:- Comprehensive review of current research papers and studies on skin disease detection using image processing.

4. Methodology and Implementation:- Describes a proposed method for skin disease detection using image processing

5. Results and analysis: We present the results obtained by applying the proposed method to a set of skin disease images.Compare the results with existing methods. work Highlight the benefits of the proposed approach.

6. Advantages and Limitations: Discusses the advantages of using image processing for skin disease detection, including non-invasiveness, speed, and potential for automation.

7 . Conclusions and future scope: Summarize the main conclusions and contributions of the report. Identify opportunities for future research, such as improving diagnostic accuracy, enhancing feature extraction, or expanding the dataset.

8. References- Defines the references used throughout the report, as mentioned above.

Chapter 2
LITERATURE REVIEW

2.1 Introduction to the overall topic

The image processing and machine learning models of skin disease detection in future days will hold bright hope. It can be investigated how computerized image technology can lead to the extraction of more accurate features from bacterial infections. This reduces the error in detection and classification rates. Integration of other machine learning models into the system, for example, an ensemble learning or transfer learning solution can significantly improve the model's capabilities and reliability.

The literature review process per se, contributes massively to the identification of some gaps and areas of knowledge, as the literature review process usually enables researchers to give full answers to some key questions and also learn from previously conducted studies. Future research within this field will consist of increasing the number of datasets, involving methodologies beyond imaging processing, integrating other machine learning algorithms, and letting the diagnosed patients under remote monitoring. These advancements open up the door to a new tool for diagnostic skin disease technology which is based on dermatologic imaging processing techniques[7].

2.2 Exhaustive literature survey

The paper 'Digital Dermatology: Skin Disease Detection Model using Image Processing' proposes a skin disease detection method based on image processing techniques. The method is mobile-based, making it accessible even in remote areas, and is noninvasive to the patient's skin[8].

The paper 'To Identify Animal Skin Disease Model Using Image Processing' presents a skin disease detection method based on image processing techniques. The method is mobile-based, making it accessible even in remote areas, and is noninvasive to the patient's skin. It includes acne detection as a specific case study[9].

The paper 'A machine learning model for skin disease classification using convolution neural network' presents a comprehensive review of deep-learning techniques applied to skin disease classification. It discusses various deep-learning architectures and highlights their performance in accurately identifying different skin diseases[10].

The paper 'Plant Disease Detection and Classification using Image' provides a survey of different techniques for detecting and classifying crop diseases using image processing. Although it focuses on plants, it highlights the importance of accurate disease detection and the use of image-processing techniques[12].

The 'Automated Skin Lesion Segmentation and Melanoma Detection: A Deep Learning Approach' research paper focuses on automated skin lesion segmentation and melanoma detection using deep learning techniques. It presents a framework that combines convolutional neural networks (CNNs) and segmentation algorithms to accurately identify and classify melanoma skin lesions[15].

Problem statement: "Building a high-quality and fast model with image recognition techniques to tackle the disparity and variation of skin types, different lighting conditions, and the wide spectrum of skin diseases."

Solution: The relevant answer to this problem can be the perfect model of skin disease detection of the present eras of image recognition algorithms. This solution will include the gathering of a skin-tone-diverse dataset that will include images from different manifestations of lighting conditions whereas many skin disorders as feasible will be taken into consideration. It's very helpful to exploit the capabilities of deep learning techniques; CNNs (Convolutional Neural Networks) will be used to achieve high-level accuracy in recognizing and sorting skin diseases with the tolerance taking place for distinct skin types and different lighting conditions. By the efforts to optimize the model structure using the Data-Augmentation and Transfer-Learning techniques, the model will be able to prove the successful analysis of skin disease images with minimal human intervention thus helping in early diagnosis and treatment.

Chapter 3

METHODOLOGY AND IMPLEMENTATION

This section outlines the methods of the suggested approach for the extraction, identification, and categorization of photos showing skin-diseased conditions. The technique will be very helpful in identifying cases of chickenpox, shingles, cellulitis, impetigo, athlete's foot, nail fungus, ringworm, and cutaneous larva migrans.

Preprocessing, feature extraction, and classification makes up the entirety of the architecture.

3.1 Block diagram

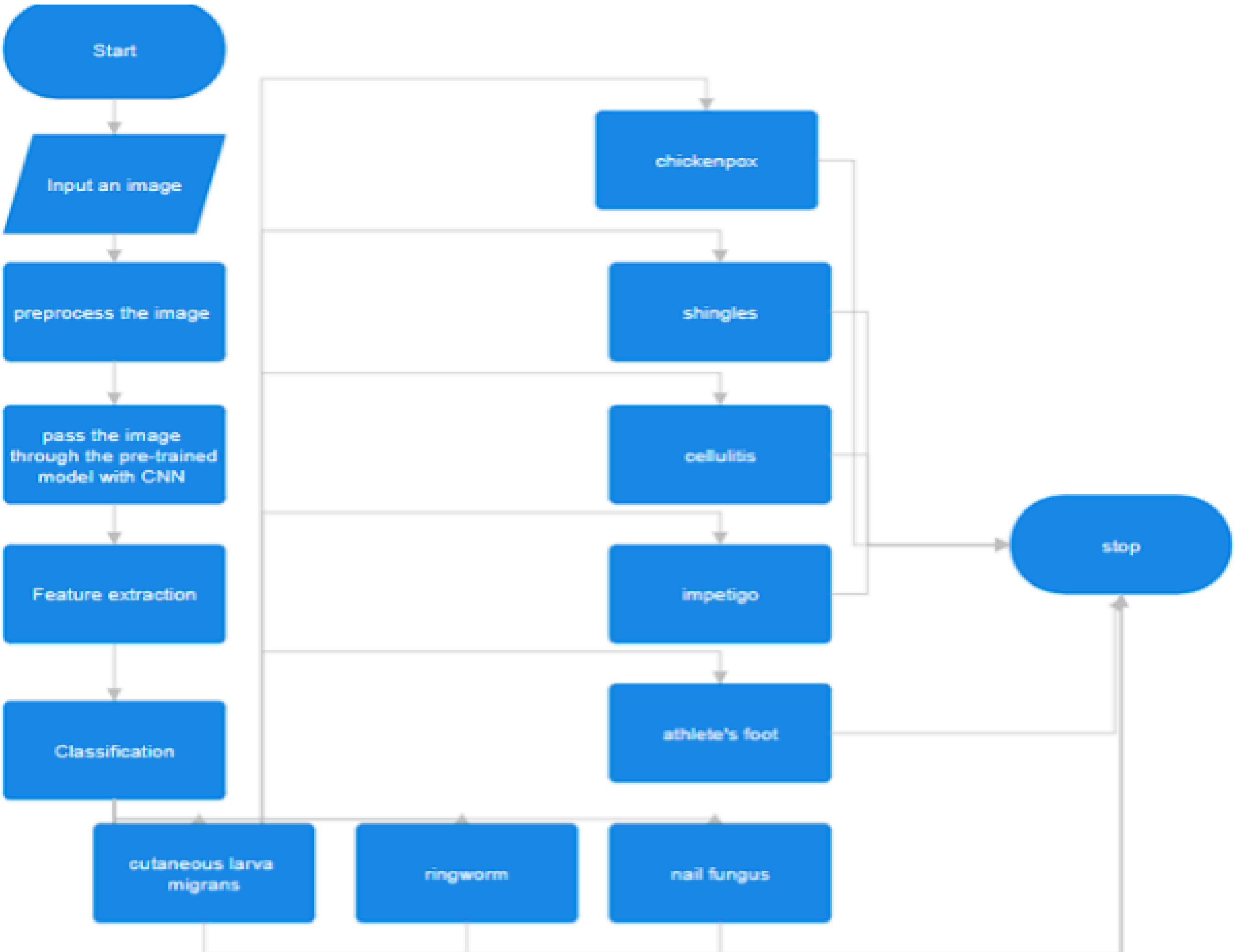


Fig 1.1 The proposed system block diagram

3.2 Algorithm

Image Acquisition: In this step, digital images of the skin are acquired using different imaging modalities, such as cameras, dermatoscopes. The quality of the acquired images can significantly impact the accuracy of subsequent analysis. These images are then stored in folders and thus it creates a Dataset. This Dataset contains 2 types of images: Train images : Consist of images to train the model.

Validation images: Consists of images to validate the model in every epoch.

After uploading the Dataset to Google Drive, it is required to mount the drive to the colab notebook. The following code can be used to mount it to drive:

```
from google.colab import drive
drive.mount('/content/drive')
```

Table 1. Dataset Details

	number of images	belonging to classes
train set	924	8
test set	233	8
total	1057	8

Pre-processing: Pre-processing involves enhancing the acquired images and reducing noise or artifacts that might affect the analysis. This step includes techniques such as filtering, image enhancement, and color correction.

```
train = train_datagen.flow_from_directory('path of the train directory',
                                          target_size = (224, 224),
                                          batch_size = 32
                                          )

val = val_datagen.flow_from_directory('path of the validation directory',
                                      target_size = (224, 224),
                                       batch_size = 16
                                      )
```

in the above snippet

target_size= Sets the size of the input image

batch_size= Sets the size of the batches of data

```
def plotImage(img_arr, label):
 for im , l in zip(img_arr, label):
  plt.figure(figsize=(5,5))
  plt.imshow(im/200)
  plt.show()
```

This function helps in applying filters to the image

Fig 2.1. Original image

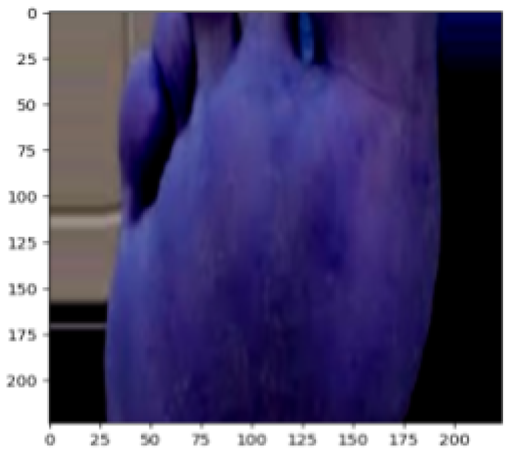


Fig 2.2 Filtered image

Segmentation: Skin lesion segmentation is the process of distinguishing the lesions from the surrounding healthy skin areas. Segmentation plays a vital role in accurately analyzing the disease characteristics and measuring its size, shape, or color. The process of dividing an image into segments makes image analysis easier. Segments are made up of sets of one or more pixels. In the Implementation, we have Segmented the lesions using the CNN architecture.

Feature Extraction: After segmentation, relevant features are extracted from the segmented regions. These features can include texture, color, shape, or statistical descriptors that represent the distinguishing characteristics of different skin diseases.

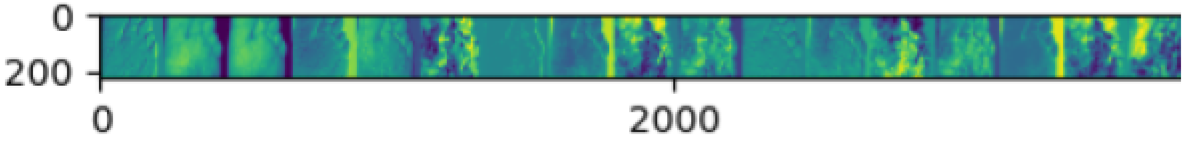


fig 3. a few features extracted when image is passed to the CNN

The above image shows how the features are extracted from an image when it is passed through the first layer. Similarly, more features are extracted when the image is passed on to the successive layer after the image is further segmented.

Classification: In the final step, the extracted features are used to classify the skin lesions into different disease categories. Machine learning algorithms, such as support vector machines, neural networks, or decision trees, can be trained and utilized to perform this classification task. By training these algorithms we can achieve a much higher accuracy of our model's prediction. In our implementation, we have created a neural network that is trained 50 times to attain a higher level of accuracy and help in the accurate classification of the 8 classes of disease mentioned above.

Chapter 4

RESULTS AND ANALYSIS

4.1 Model & CNN layers

The base model we used is VGG19. VGG-19 is a convolutional neural network(CNN) that is 19 layers deep. The pre-trained network can classify images into 8 object categories, such as Cellulitis, Impetigo, Athlete’s Foot, Nail fungus, Ringworm, Cutaneous larva migrans, Chickenpox, and Shingles. In this study, a Google Colab environment was used for CNN implementation. CNN's detection phase consists of CNN modeling, data preparation, data visualization, and data input. Experiments were conducted utilizing different values of the learning rate and the epoch parameter during the training phase. We set the number of epochs to train the model to 50, so the model will train for 50 epochs.

4.2 Evaluation Metric

The evaluation metric we used is accuracy. The evaluation metric Accuracy measures the proportion of correctly classified instances out of the total number of instances. It is calculated as the ratio of the number of correct predictions to the total number of predictions.

```
   acc = model.evaluate_generator(val)[1]
the acc is 90.12875556945801%
```

Fig .5 shows the accuracy of the model

evaluate_generator Evaluate the loaded model's performance on the validation data. This method computes the loss and any specified metrics for the provided data generator. Here it is clear that the accuracy of our model is approximately 90.128% for a particular classification.

Fig 6.1. Input image

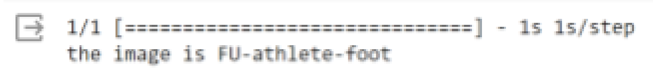


Fig 6.2 shows the output shown to the user after classifying the disease

4.3 Accuracy over Epochs

We extracted a training history dictionary that contains metrics collected during training, such as accuracy and loss, for both training and validation datasets and plotted a graph of training accuracy over epochs using the **matplotlib.pyplot.plot()** function. The '**accuracy**' key is used to access the training accuracy values from the history dictionary.

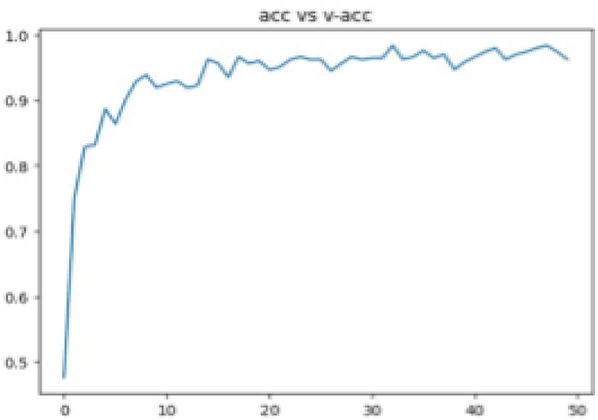


fig 7.1 compares training and validation accuracy.

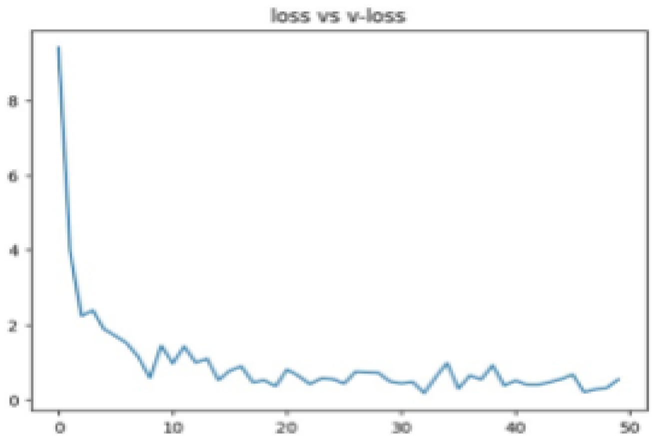


fig 7.2 compares training and validation loss.

Additional regularization methods can be required if the training accuracy/loss keeps getting better/worse while the validation accuracy/loss stays the same or gets worse. This could be a sign of overfitting. Our goal is to increase our model's accuracy and to make it more reliable for health professionals. We also need to avoid any chances of overfitting and underfitting and figure out strategies to eradicate both the previously mentioned issues in our model.

Chapter 5

ADVANTAGES, LIMITATIONS, and APPLICATIONS

5.1Advantages

Non-invasive Diagnosis: Utilizing photographic imaging, dermatology that detects skin diseases using image processing methods offers a way of diagnosing several skin diseases without the need for further invasive testing and treatments. For the patients, there is no longer the requirement for invasive procedures to be used and they can have testing done as quickly as possible without actually having to suffer from the usual discomfort and complications that come with the traditional forms of diagnosis[18].

Non-Invasive and Fast: The method of skin nondisease diagnosis by using imaging processing is completely safe since it does not require different physical contact with the patient's skin that in general may result in some complications. It is also a rapid process that offers immediate results in the way of quick diagnosis and treatment for on-time action[18].

Cost-Effective: Imaging-based methods for diagnostic purposes of skin diseases usually entail utilizing simple gadgets, consisting of a camera or a computer. This brings about the approach that is affordable and feasible where accessibility is difficult as in a resource-poor setting[18].

Early Detection: Through the utilization of capable image processing algorithms, detection models on skin disease can be a major factor in the early detection of dermatological problems. With early discovery, we have an opportunity and chance to provide timely assistance and proper treatment as this can prevent skin diseases from getting worse[19].

Handling Disparity and Variation: Skin tone and lighting adaptation, which can be dealt with by computer vision methods, are a few of the tasks we can handle. Thanks to these methods, the processing can be successful. With the help of identifying robust algorithms and preprocessing steps, such techniques perform accurateness in the identification of skin color, texture and lighting time, and skin diseases[19].

5.2Limitations

Variability in Image Quality: The problem of the heterogeneity in image quality is the second constraint but it has many factors like light, the camera specificity, and the way the patient is positioned in the room. Lack of quality in imaging can obstruct the process of specific feature extraction and classification that may affect correct diagnosis[20].

Limited Dataset Diversity: Heterogenous, rich, and vociferous inputs supporting the training of the AI system for skin disease recognition are getting short supplies. Restricted diversity of skin types, disorders, and demographics in the data set will impede an algorithm's ability to generalize well to foreseeable issues and cases, especially those that are not well-represented[20].

The complexity of Skin Diseases: The most significant sign of a skin disease (as it greatly affects its detection and identification) is a definite picture of the appearance. Microscopic images sometimes show only similar appearances of different diseases, implying it will be difficult to correctly categorize or spot a specific one based simply on a general view[21].

Interpretability of Deep Learning Models: Major issues with deep learning models playing the role of a simple black box, alongside, the inability of humans to understand the model decisions in many situations, are the main obstacles we face in the model interpretation process. Being potentially entitled to the extent that the model doesn't know how the predictions are correct fundamentally shatters the comprehension of the model and removes the probability of trust in clinical settings[21].

Need for Validation and Clinical Trials: Although such technology has the benefit of identifying dermatological diseases purely through skin imaging, it is necessary to increase the scope of experiences and clinical trials to step closer to real-life use cases through the technology. The integration of these tools into the diagnosis process stationed great wants for accurate assessment in terms of both their reliability and effectiveness.[21][22]

5.3Applications

Automated Screening System: Imaging technology methods may be adapted to develop automated disease surveillance methods in the dermatology field. These devices are used to look at images of the skin infected and analyze images to identify complex types of the disease, hence they screen general skin cases rapidly and efficiently. [22]

Research and Development: Skin disease detection utilizing image processing serves as a contribution to the R&D sphere in the health domain. Thanks to this, new algorithms, approaches, and models can be developed in a bid to improve the skin disease detection system's precision and efficiency. [22]

Diagnosis Support: The brilliance of image processing-based methods may help medical professionals with skin disease diagnosis. Skin images are being studied by these techniques and they generate objective information and provide necessary support in the decision process; the approved diagnosis can be made more accurately and promptly by physicians. [23]

Mobile Applications: Mobile applications have a possibility for image processing techniques to be built in to generate a diagnosis of skin disease detection. This usage of computer vision technology makes image recognition of skin diseases possible, thereby giving people knowledge about the nature of the disease and its intensity. [24]

Telemedicine and Remote Diagnosis: Image processing-based skin disease targeting will be ideal for telemedicine and remotely situated diagnostic cases. Healthcare professionals can also help patients who cannot visit the doctor's office by employing the methods of taking skin images and analyzing them remotely so that a diagnosis and treatment can be done accordingly.[25]

Chapter 6

CONCLUSION AND FUTURE SCOPE

In conclusion, the utilization of teledermatology imaging and image processing techniques has led to significant advancements in the field of dermatology and telemedicine. These developments have improved the quality of skin disease evaluation and successfully extracted the characteristics of various skin diseases. Additionally, the introduction of an automatic disease detection and severity measurement model has revolutionized dermatology by providing a robust solution for diagnosing and treating conditions like eczema. The model efficiently detects affected areas, differentiates between mild and severe areas based on color and texture, and automatically computes severity indices. Overall, these breakthroughs have the potential to transform dermatological care by improving precision, speed, and tailored treatment options for chronic skin disorders. They embody the principles of early detection, accurate evaluation, and patient-focused therapy, offering essential insights for decision-making in dermatological care[26].

The future for the model of detecting skin diseases with image processing is hopeful and looks promising. If we bring together more extensive and variously sourced datasets, the model can be made to identify a greater range of skin conditions more correctly and consistently. Using advanced image processing techniques, there are possibilities to extract more accurate and detailed features from images of skin lesions, which will improve the accuracy of disease detection and classification. Further integrating with other machine learning algorithms such as ensemble learning or transfer learning can enhance the generality ability and performance of the model. The model can also be integrated into telemedicine platforms or mobile applications for remote diagnosis and monitoring of these conditions thereby enhancing accessibility to health services. This technological progress in dermatology allows for reliable accessible means to recognize skin diseases using imaging techniques[27].

REFERENCES

[1] A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science, Volume 163, 2019, Pages 85-92

[2] A machine learning approach for skin disease detection and classification using image segmentation, Healthcare Analytics, Volume 2, November 2022, 100122.

[3] Skin Disease Recognition Method Based on Image Color and Texture Features, National Library of Medicines, August 2018.

[4] A Method Of Skin Disease Detection Using Image Processing, ScienceDirect, 2019.

[5] A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science, Volume 163, 2019, Pages 85-92

[6] Literature Reviews for Education and Nursing Graduate Students, Rebus Press,2017.

[7] Literature Review Tips for the Introduction and Discussion Sections, Enago Academy,2024.

[8] Digital dermatology: Skin disease detection model using image processing, IEE,2017.

[9] To Identify Animal Skin Disease Model Using Image Processing, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2019.

[10] A machine learning model for skin disease classification using convolution neural network, International Journal of Computing, Programming and Database Management 2022

[11] Intrusion detection system in a cloud environment: Literature survey & future research directions, ScienceDirect, 2022.

[12] Plant Disease Detection and Classification using Image Processing: A Review, UIET, CSE Department, Panjab University, 2019.

[13] Literature survey on impulse noise reduction, Signal & Image Processing: An International Journal (SIPIJ) - WJCI Indexed, 2013.

[14] A Review on Grape Disease Detection and Classification using Image Processing, International Research Journal of Engineering and Technology, 2021.

[15] Automatic Skin Lesion Segmentation and Melanoma

[16] Face Recognition using Face Alignment and PCA Techniques: A Literature Survey, IOSR Journals, 2015.

[17] Proceedings vs Special Issue, AIJR, 2011.

[18] A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science Volume 163, 2019

[19] A machine learning approach for skin disease detection and classification using image segmentation, Healthcare Analytics, Volume, 2023

[20] A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science Volume 173, 2019

[21] Systematic review of deep learning image analyses for the diagnosis and monitoring of skin disease, npj Digital Medicine, 2013.

[22] A Method Of Skin Disease Detection Using Image Processing And Machine Learning, Procedia Computer Science Volume 163, 2019

[23] A machine learning approach for skin disease detection and classification using image segmentation, Healthcare Analytics, Volume 3, 2022

[24] Detection of skin disease by using image processing, ResearchGate, 2019

[25] Automated Skin Disease Identification using Deep Learning Algorithm, Biomedical and Pharmacology Journal, 2018.

[26] Automatic Detection and Severity Measurement of Eczema Using Image Processing, IEEE, 2016, Feature Based Skin Disease Estimation Using Image Processing for Teledermatology, University of Dhaka, 2017.

[27] A machine learning approach for skin disease detection and classification using image segmentation, Healthcare Analytics, Volume 2, 2022.

Appendix I: Soft Code Flowcharts

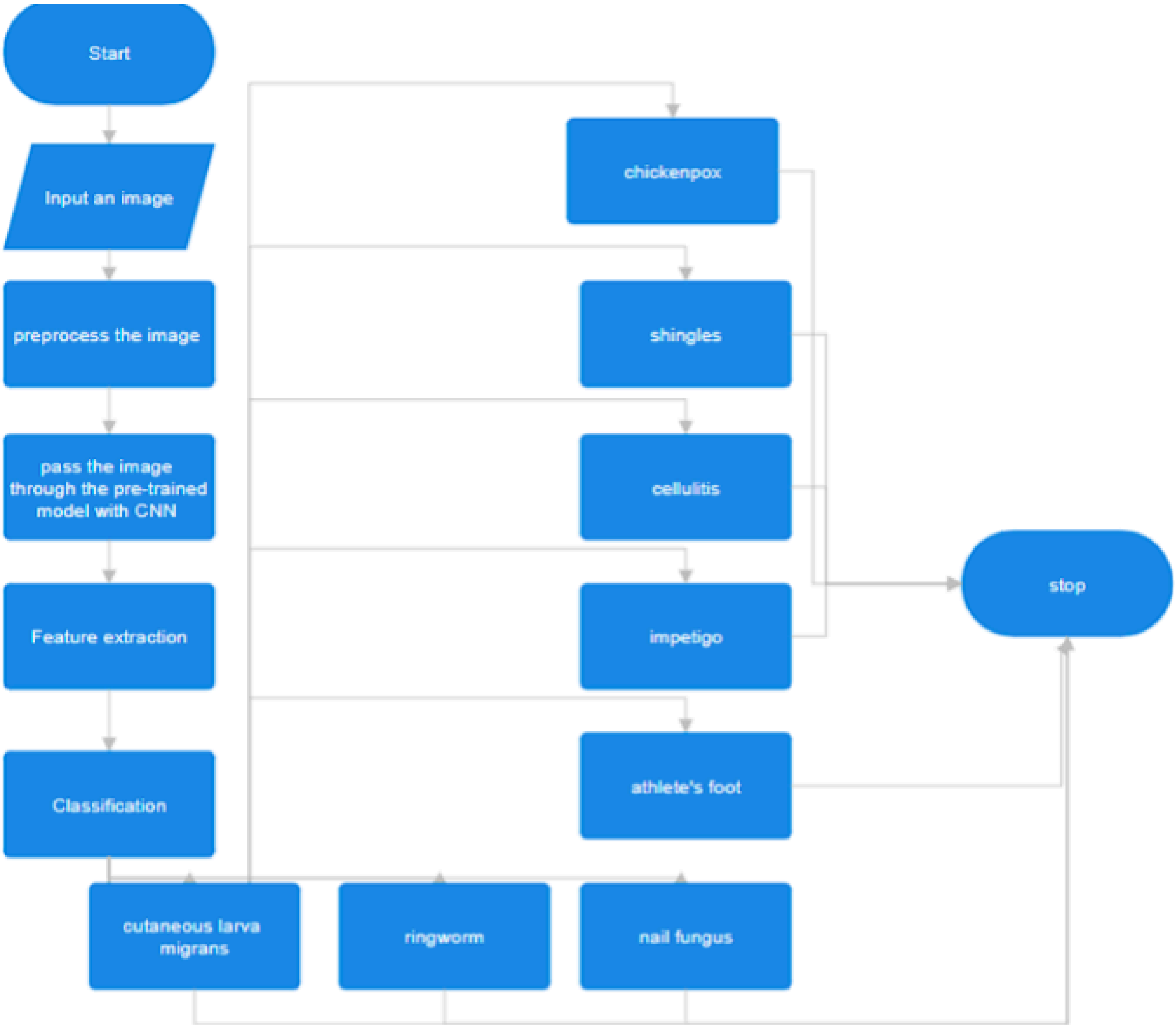


Fig 1. The proposed system block diagram

Appendix II: Specification of components/ libraries used

- **NumPy** is a Python library that supports massive, multidimensional arrays and matrices, as well as a vast set of mathematical functions for manipulating these arrays.
- **Pandas** is a Python package that provides high-performance, user-friendly data structures and analytic tools for structured (tabular, multidimensional, possibly heterogeneous) and time series datasets.
- **Matplotlib.pyplot** is a Python package that offers a MATLAB-style plotting framework. It generates publication-quality numbers in a range of hardcopy and interactive formats across platforms.
- **OS**: is a Python module that provides a portable mechanism to access operating system-specific functions.
- **Keras** is a Python package that implements a high-level neural network API. It simplifies the process of building and training neural networks, eliminating the need to worry about low-level implementation details.
- **Keras.preparing.image**: is a Keras module that contains tools for preprocessing images.
- **ImageDataGenerator** is a Keras class that generates picture data in batches using real-time data augmentation.
- **Img_to_array**: This Keras function transforms a PIL Image instance to a NumPy array.
- **Load_img**: This Keras function loads an image from a file.
- **Keras.applications.vgg19** is a Keras module that includes pre-trained VGG19 models.
- **VGG19** is a convolutional neural network architecture designed by the Visual Geometry Group at the University of Oxford.
- **Preprocess_input** is a Keras function that prepares an image for usage with the VGG19 model.
- **Decode_predictions** is a Keras function that decodes the predictions of a VGG19 model.

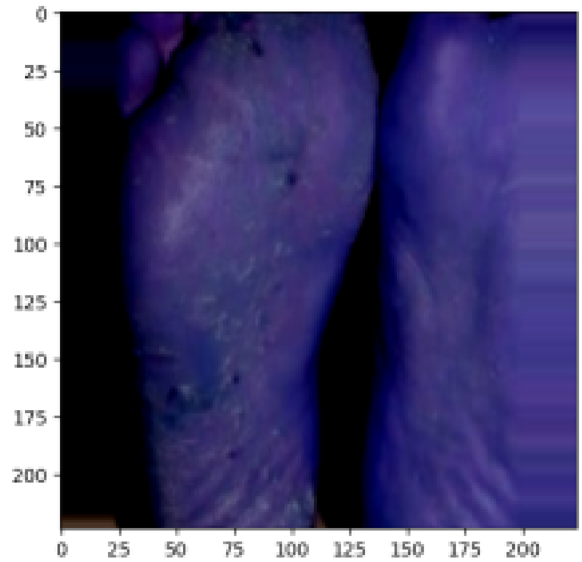
Appendix C: Source code

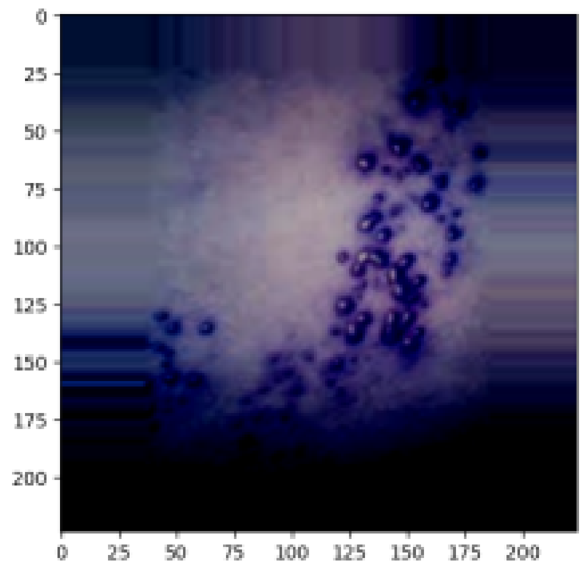
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import keras
from keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img
from keras.applications.vgg19 import VGG19, preprocess_input, decode_predictions

from google.colab import drive
drive.mount('/content/drive')

train_datagen = ImageDataGenerator(zoom_range= 0.5, shear_range= 0.3,
horizontal_flip = True, preprocessing_function= preprocess_input) val_datagen =
ImageDataGenerator(preprocessing_function=preprocess_input)

train = train_datagen.flow_from_directory('/content/drive/MyDrive/college/tech
project/skin-disease-datasaet/train_set', target_size = (224, 224),
batch_size = 32
)
val = val_datagen.flow_from_directory('/content/drive/MyDrive/college/tech
project/skin-disease-datasaet/test_set', target_size = (224, 224),
batch_size = 16
)
```

Found 934 images belonging to 8 classes.
Found 233 images belonging to 8 classes.

```
t_img, label = train.next()
def plotImage(img_arr, label):
for im , l in zip(img_arr, label):
plt.figure(figsize=(5,5))
```

```
plt.imshow(im/200)
plt.show()
plotImage(t_img[:3], label[:3])
```





```
from keras.layers import Dense, Flatten
from keras.models import Model
from keras.applications.vgg19 import
VGG19, preprocess_input,
decode_predictions import keras
base_model = VGG19(input_shape=(224,224,3), include_top= False)
```

```
for layer in base_model.layers:
  layer.trainable = False
base_model.summary()
```

```
Model: "vgg19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 20024384 (76.39 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

```
x = Flatten()(base_model.output)
x = Dense(units= 8 , activation = 'softmax')(x)
model = Model(base_model.input, x)
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 8)                 200712

=================================================================
Total params: 20225096 (77.15 MB)
Trainable params: 200712 (784.03 KB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

Early stopping and checkpoint

```
from keras.callbacks import ModelCheckpoint, EarlyStopping
es = EarlyStopping(monitor='val_accuracy', min_delta= 0.01, patience = 3, verbose = 1)
mc = ModelCheckpoint(filepath="Edureka.h5",
                     monitor='val_accuracy',
                     min_delta= 0.01,
                     patience = 3,
                     verbose = 1,
                     save_best_only= True)
cb = [es, mc]
```

Training the model

his = model.fit_generator(train ,

steps_per_epoch = 16,

epochs= 50,

verbose=1,

callbacks= cb,

validation_data= val,

validation_steps =16)

```
16/16 [============================] - ETA: 0s - loss: 0.3255 - accuracy: 0.9844WARNING:tensorflow:Early stopping
WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
16/16 [============================] - 8s 505ms/step - loss: 0.3255 - accuracy: 0.9844
Epoch 48/50
16/16 [============================] - ETA: 0s - loss: 0.2901 - accuracy: 0.9746WARNING:tensorflow:Early stopping
WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
16/16 [============================] - 10s 594ms/step - loss: 0.2901 - accuracy: 0.9746
Epoch 49/50
16/16 [============================] - ETA: 0s - loss: 0.3570 - accuracy: 0.9815WARNING:tensorflow:Early stopping
WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
16/16 [============================] - 9s 562ms/step - loss: 0.3570 - accuracy: 0.9815
Epoch 50/50
16/16 [============================] - ETA: 0s - loss: 0.4894 - accuracy: 0.9691WARNING:tensorflow:Early stopping
WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
16/16 [============================] - 8s 477ms/step - loss: 0.4894 - accuracy: 0.9691
```

h = his.history

h.keys()

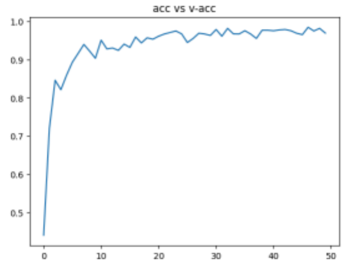
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

plt.plot(h['accuracy'])

plt.plot(h['val_accuracy'], c = "red")

plt.title("acc vs v-acc")

plt.show()

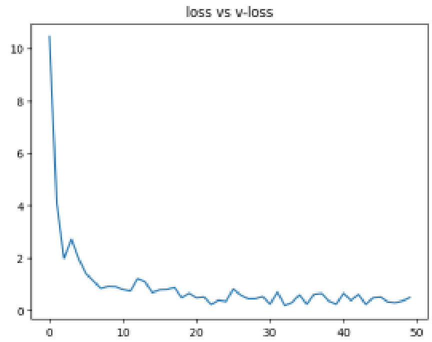


plt.plot(h['loss'])

plt.plot(h['val_loss'], c = "red")

plt.title("loss vs v-loss")

plt.show()




```
from keras.models import load_model
model = load_model("/content/Edureka.h5")
acc = model.evaluate_generator(val)[1]
print(f"the acc is {acc*100}%")
```

<ipython-input-21-73754078cd2d>:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version.
Pl acc = model.evaluate_generator(val)[1]
the acc is 70.38626670837402%

```
ref = dict(zip(list(train.class_indices.values()), list(train.class_indices.keys())))

def prediction(path):
  img = load_img(path, target_size= (224,224))
  i = img_to_array(img)
  im = preprocess_input(i)
  img = np.expand_dims(im, axis = 0)
  pred = np.argmax(model.predict(img))
  print(f"the image is {ref[pred]}")
```

upload the image to google and copy the path and paste in the path.

```
img_path= "/content/drive/MyDrive/college/tech project/try1.jpeg"
prediction(img_path)
```

1/1 [==============================] - 1s 1s/step
the image is FU-athlete-foot