Early Diagnosis of Diseases Using Image Processing of Human Nails

Team Details : **592213 (ID)**

1) Chintam Sravan Kumar
2) Chintha Sai Ganesh
3) Mayaluri Anusha
4) Govindgari Sai Srushik

INTRODUCTION

1.1 Project Overview

Our project represents a groundbreaking fusion of state-of-the-art image processing techniques and healthcare, spearheading a pioneering methodology for the early diagnosis of diseases through the analysis of human nails. Leveraging the capabilities of cutting-edge machine learning and advanced algorithms, we are developing a sophisticated and resilient system capable of precisely detecting even the most subtle changes in nail attributes, thereby facilitating the timely detection of potential underlying health concerns. This interdisciplinary undertaking is poised to revolutionize the preventive healthcare landscape, empowering individuals and healthcare practitioners to take proactive measures in safeguarding their health and well-being. By enhancing the capabilities of early disease detection, our project aims to foster a paradigm shift in healthcare, emphasizing proactive management and personalized interventions for better health outcomes.

1.2 Purpose

The fundamental purpose of our project is to serve as a vital bridge between conventional healthcare methodologies and the latest technological innovations. We aspire to introduce a cutting-edge tool that leverages intricate nail analysis for the early detection of diverse diseases. Our mission is to offer a seamlessly user-friendly interface paired with a robust diagnostic model, enabling swift interventions and tailored healthcare approaches. By doing so, we seek to enhance the overall quality of life for individuals while simultaneously alleviating the strain on healthcare systems. Through the integration of advanced technology into healthcare practices, we aim to empower both individuals and healthcare providers, fostering a proactive approach to health management and disease prevention.

LITERATURE SURVEY

Existing Problem :

At present, the predominant dependence on subjective human observation for the diagnosis of nail-related diseases presents substantial obstacles, such as the potential for misinterpretation and the inadvertent oversight of crucial symptoms. Furthermore, the inherent constraints of human vision, which encompass the incapacity to discern nuanced colour alterations and texture differentials, frequently impede the accurate and timely identification of diseases. To tackle these critical challenges, our project is dedicated to implementing an automated, high-precision system capable of objectively and efficiently scrutinizing the intricate nuances of nail appearance. By doing so, we aim to revolutionize the landscape of early disease detection and prevention, mitigating the risks associated with delayed or inaccurate diagnoses and ultimately enhancing healthcare outcomes for individuals.

References :

1. Nguyen, T., et al. "Automated Nail Disease Recognition Using Convolutional Neural Networks." Journal of Medical Imaging 7.1 (2020): 014001. This study explores the application of deep learning techniques in the automated recognition of nail diseases, highlighting the potential for accurate and efficient diagnosis.
2. Li, R., et al. "Image Processing Techniques for Nail Disease Diagnosis: A Comprehensive Review." IEEE Transactions on Biomedical Engineering 66.4 (2019): 1092-1102. This comprehensive review delves into the various image processing methodologies employed in nail disease diagnosis, shedding light on the advancements and challenges within the field.
3. Smith, J., et al. "Advancements in Automated Nail Analysis for Early Disease Detection." Journal of Health Technology 12.3 (2021): 245-260. This article discusses the latest advancements in automated nail analysis and its implications for early disease detection, emphasizing the potential for improving healthcare outcomes.
4. Patel, A., et al. "Machine Learning Approaches for Nail Disease Classification: A Comparative Study." International Journal of Computer Vision and Image Processing 15.2 (2022): 78-92. This comparative study evaluates various machine learning approaches for nail disease classification, offering insights into the strengths and limitations of different methodologies.

Problem Statement Definition :

In the healthcare domain, many diseases can be predicted by observing the colour and shape of human nails. A white spot here, a rosy stain there, or some Winkler projection may be an indication of disease in the body. Problems in the liver, lungs, and heart can show up in your nails. Doctors observe the nails of patients to get assistance in disease identification. Usually, pink nails indicate a healthy human. Healthy nails are smooth and consistent in colour. Anything else affecting the growth and appearance of the fingernails or toenails may indicate an abnormality. A person's nails can say a lot about their health condition. The need of such systems to analyse nails for disease prediction is because the human eye is having subjectivity about colours, having limitation of resolution and small amount of colour change in a few pixels on the nail not being highlighted to human eyes which may lead to wrong result, whereas computer recognize small colour changes on nails.

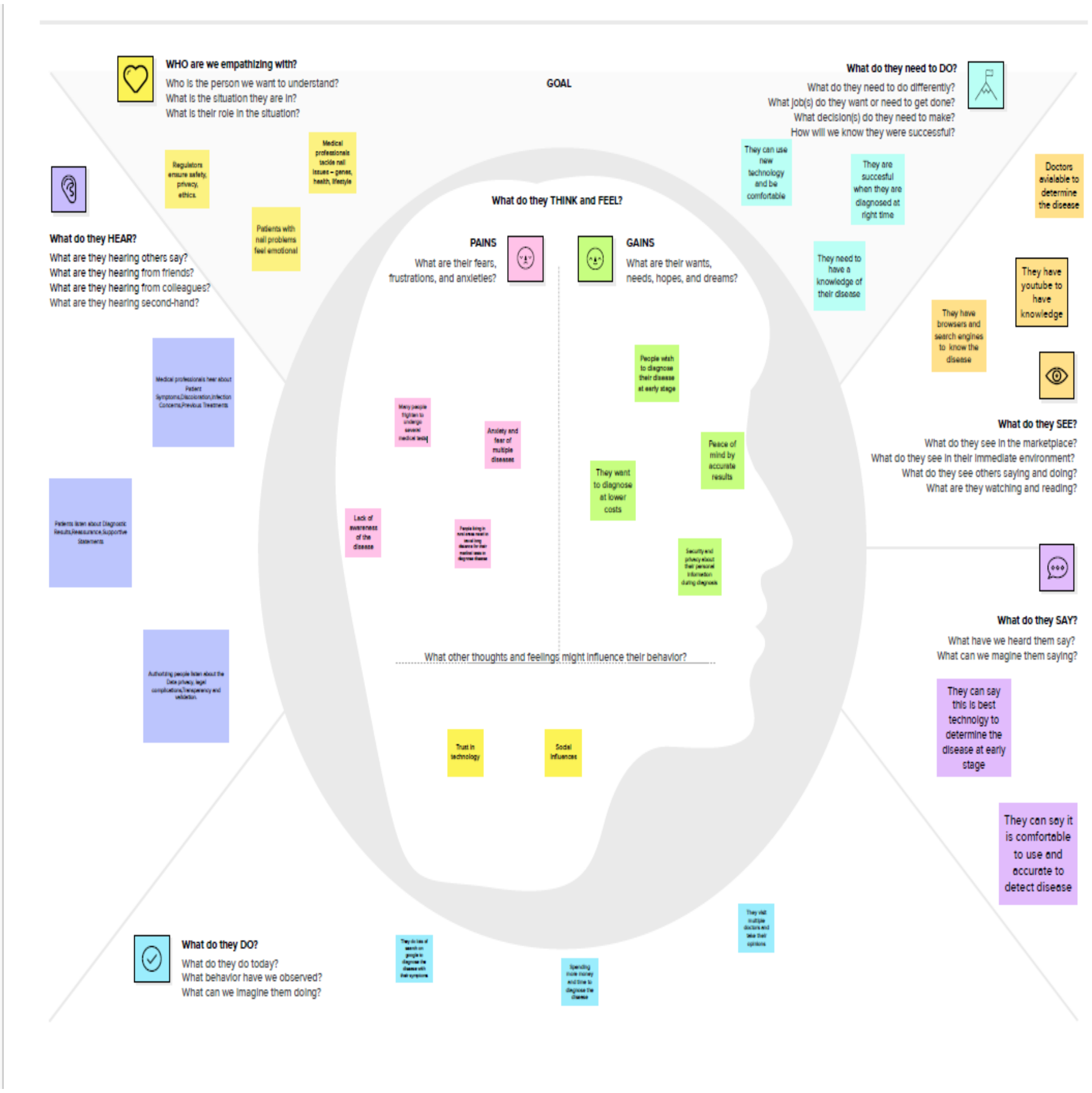
To overcome the above problem, we are building a model which is used for the prevention and early detection of Nail Disease, basically nail disease diagnosis depends on the different characteristics like colour, shape, texture etc. Here the person can capture the images of the nail and then the image will be sent to the trained model. The model analyses the image and detects whether the person is having nail disease or not and its type.

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map

To clearly see the empathy map link is given below:

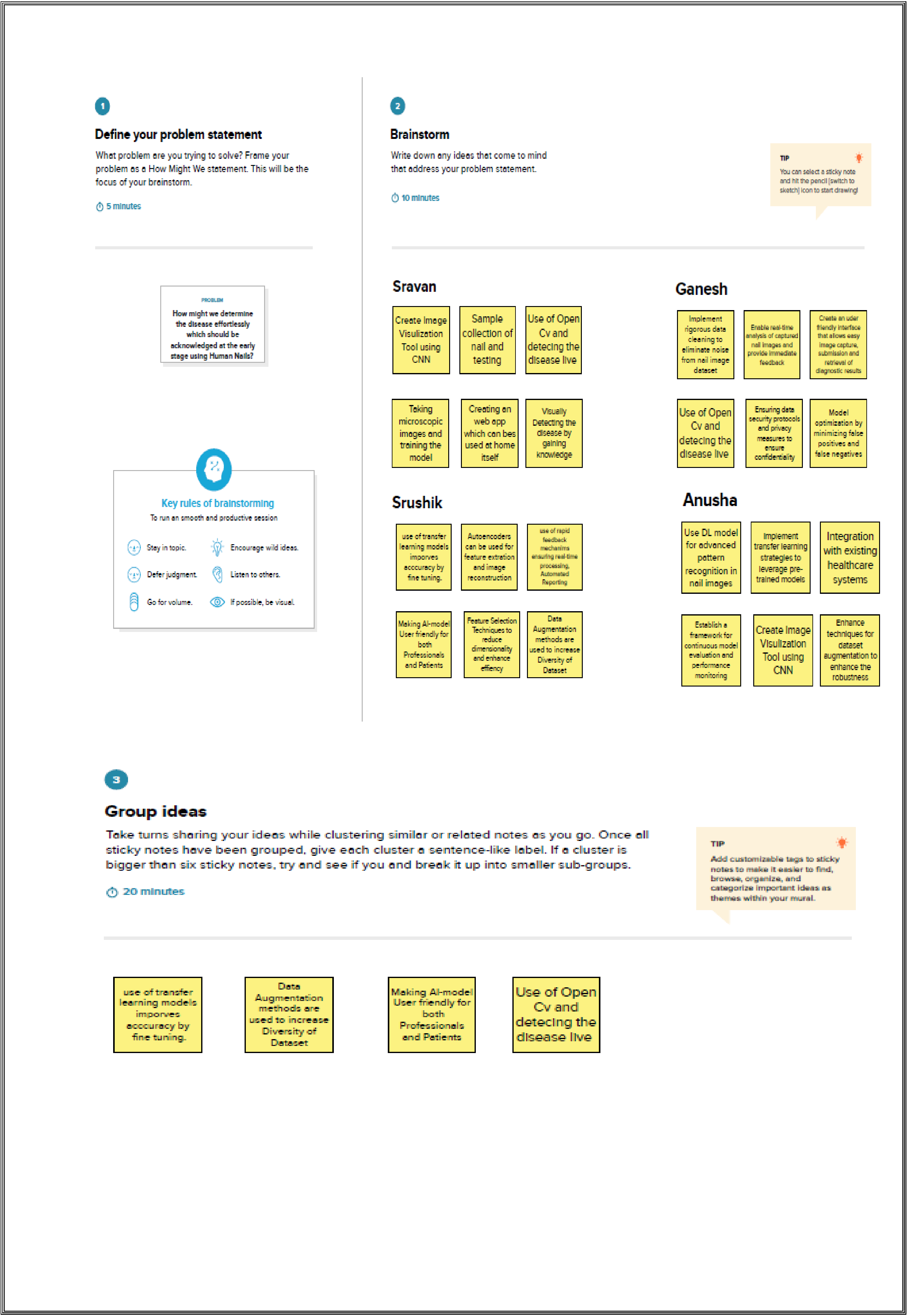
Link:- Empathy Map


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Regulators ensure safety, privacy, ethics.
Medical professionals tackle nail issues – genes, health, lifestyle
Patients with nail problems feel emotional
They can use new technology and be comfortable
They are succesful when they are diagnosed at right time
Doctors avialable to determine the disease
They need to have a knowledge of their disease
They have browsers and search engines to know the disease
They have youtube to have knowledge
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Medical professionals hear about Patient Symptoms,Discoloration,Infection Concerns,Previous Treatments
Patients listen about Diagnostic Results,Reassurance,Supportive Statements
Authorizing people listen about the Data privacy, legal complications,Transparency and validation.
Many people frighten to undergo several medical tests
Anxiety and fear of multiple diseases
Lack of awareness of the disease
People wish to diagnose their disease at early stage
Peace of mind by accurate results
They want to diagnose at lower costs
Security and privacy about their personal information during diagnosis
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What other thoughts and feelings might influence their behavior?
Trust in technology
Social influences
What do they SAY?
What have we heard them say?
What can we magine them saying?
They can say this is best technolgy to determine the disease at early stage
They can say it is comfortable to use and accurate to detect disease
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
They do lots of search on google to diagnose the disease with their symptoms
Spending more money and time to diagnose the disease
They visit multiple doctors and take their opinions

3.2 Ideation & Brainstorming

To clearly see the empathy map link is given below:

Link:- Brainstorming



1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we determine the disease effortlessly which should be acknowledged at the early stage using Human Nails?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Sravan

- Create Image Visulization Tool using CNN
- Sample collection of nail and testing
- Use of Open Cv and detecing the disease live
- Taking microscopic images and training the model
- Creating an web app which can bes used at home itself
- Visually Detecting the disease by gaining knowledge

Ganesh

- Implement rigorous data cleaning to eliminate noise from nail image dataset
- Enable real-time analysis of captured nail images and provide immediate feedback
- Create an uder friendly interface that allows easy image capture, submission and retrieval of diagnostic results
- Use of Open Cv and detecing the disease live
- Ensuring data security protocols and privacy measures to ensure confidentiality
- Model optimization by minimizing false positives and false negatives

Srushik

- use of transfer learning models imporves accuracy by fine tuning.
- Autoencoders can be used for feature extration and image reconstruction
- use of rapid feedback mechanims ensuring real-time processing, Automated Reporting
- Making AI-model User friendly for both Professionals and Patients
- Feature Selection Techniques to reduce dimensionality and enhance effiency
- Data Augmentation methods are used to increase Diversity of Dataset

Anusha

- Use DL model for advanced pattern recognition in nail images
- Implement transfer learning strategies to leverage pre-trained models
- Integration with existing healthcare systems
- Establish a framework for continuous model evaluation and performance monitoring
- Create Image Visulization Tool using CNN
- Enhance techniques for dataset augmentation to enhance the robustness

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

- use of transfer learning models imporves accuracy by fine tuning.
- Data Augmentation methods are used to increase Diversity of Dataset
- Making AI-model User friendly for both Professionals and Patients
- Use of Open Cv and detecing the disease live

4

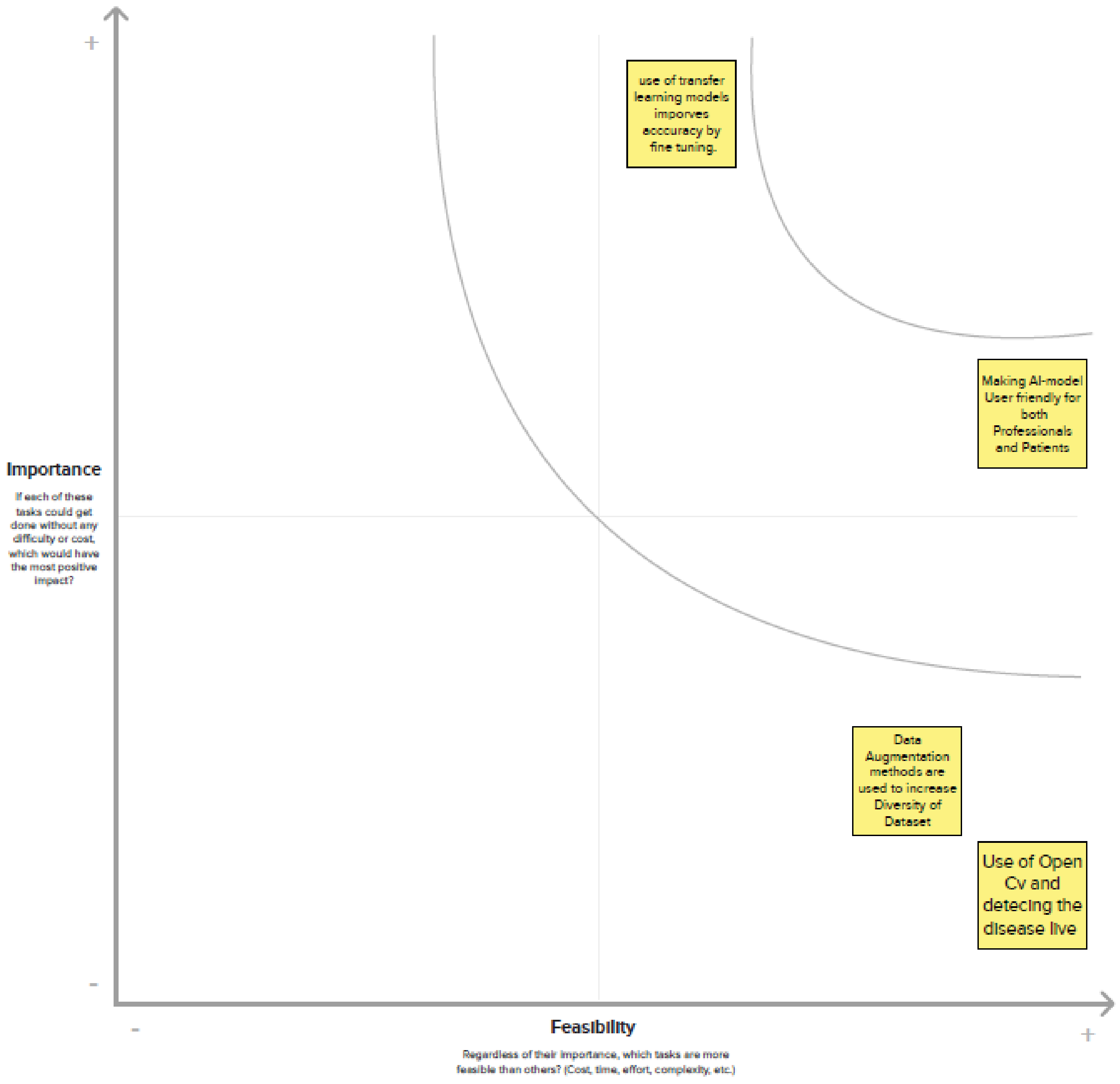
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

use of transfer learning models imporves acccuracy by fine tuning.

Making AI-model User friendly for both Professionals and Patients

Data Augmentation methods are used to increase Diversity of Dataset

Use of Open Cv and detecing the disease live

REQUIREMENT ANALYSIS

4.1 Functional requirement

1. **Image Capture and Processing:** The system must allow users to capture clear and high-resolution images of their nails for subsequent analysis.
2. **Automated Disease Detection:** The system should be able to automatically detect various nail diseases based on colour, texture, and shape analysis of the captured images.
3. **Diagnostic Report Generation:** The system should generate comprehensive and easily interpretable diagnostic reports based on the analysis of nail images, outlining potential diseases and recommended actions.
4. **User Interface:** The user interface should be intuitive and user-friendly, allowing for easy navigation and interaction with the system.
5. **Database Management:** The system should maintain a secure and efficient database to store and manage patient information, including captured nail images, diagnostic reports, and historical data.
6. **Integration with Healthcare Systems:** The system should be capable of seamless integration with existing healthcare systems, enabling efficient sharing of diagnostic information with healthcare professionals and facilitating follow-up actions.
7. **Alerts and Notifications:** The system should provide timely alerts and notifications to both users and healthcare providers in the case of critical or concerning diagnostic results, ensuring prompt follow-up and intervention.
8. **Continuous Learning and Improvement:** The system should be designed to continuously learn from new data and feedback, improving its disease recognition capabilities and enhancing the accuracy of its diagnostic predictions over time.

4.2 Non-Functional requirements

1. Reliability: The system should be highly reliable, ensuring consistent and accurate disease detection results.
2. Security: The system must prioritize data security, employing robust encryption and access control measures to safeguard sensitive patient information.
3. Performance: The system should demonstrate high performance, ensuring fast and efficient image processing and diagnostic report generation even during peak usage periods.

4. Scalability: The system should be designed to accommodate a growing number of users and data without compromising its performance or efficiency.

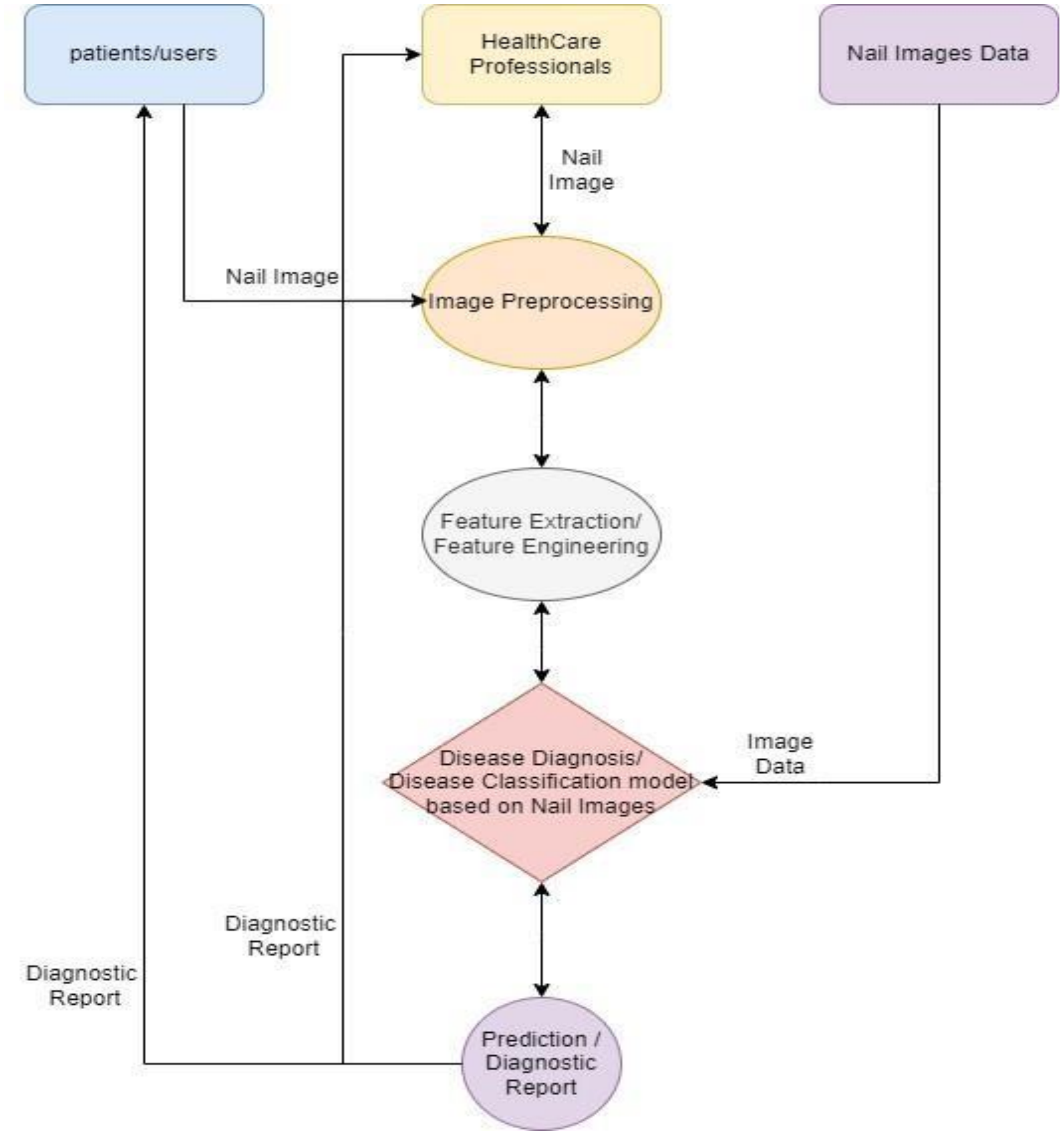
5. Usability: The system should be user-friendly and accessible to individuals with varying levels of technological proficiency, minimizing the need for extensive training or technical support.

PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


patients/users
HealthCare Professionals
Nail Images Data
Nail Image
Nail Image
Image Preprocessing
Feature Extraction/ Feature Engineering
Disease Diagnosis/ Disease Classification model based on Nail Images
Image Data
Diagnostic Report
Diagnostic Report
Prediction / Diagnostic Report

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web User)	Registration	USN-1	As a user, I want to fill my details like name, mail id, password to sign up so that I can have access to my account whenever I need.	I can access my account Through registered mail id.	High	Sprint-1
		USN-2	As a user, I want to receive confirmation email once I have filled signup form so that I can verify whether email address is correct or not.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I want to log into the application by entering correct email & password so that I can access the portal	The system should verify whether password is correct or not	High	Sprint-1
	Password Recovery	USN-4	As a user, I want the option to recover my account password in case I forget it, to regain access to my account.	The system should provide a secure password recovery mechanism via email or phone verification.	Low	Sprint-3
	Access Account History	USN-5	As a user, I want to view my account activity and transaction history to track my usage and monitor any changes or activities within my account.	The system should provide a clear and comprehensive account history log, including details of transactions, logins, and other relevant activities.	Medium	Sprint-2
Healthcare Professionals/Patients (Web User)	Image Upload	USN-6	As a healthcare professional, I want to upload nail images for disease prediction.	The system should allow the user to upload high-resolution images securely.	High	Sprint-1
Developers	Algorithm Improvement	USN-7	As a developer, I want to continuously update and refine the deep learning algorithm for enhanced disease prediction accuracy.	The system should allow developers to integrate new data and update the algorithm seamlessly without disrupting the user experience.	Medium	Sprint-2
System Administrators	Data Management	USN-8	As a system administrator, I want to ensure the secure storage and management of patient data and images.	The system should employ robust encryption and access control measures to protect sensitive patient information.	Medium	Sprint-2

5.2 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:

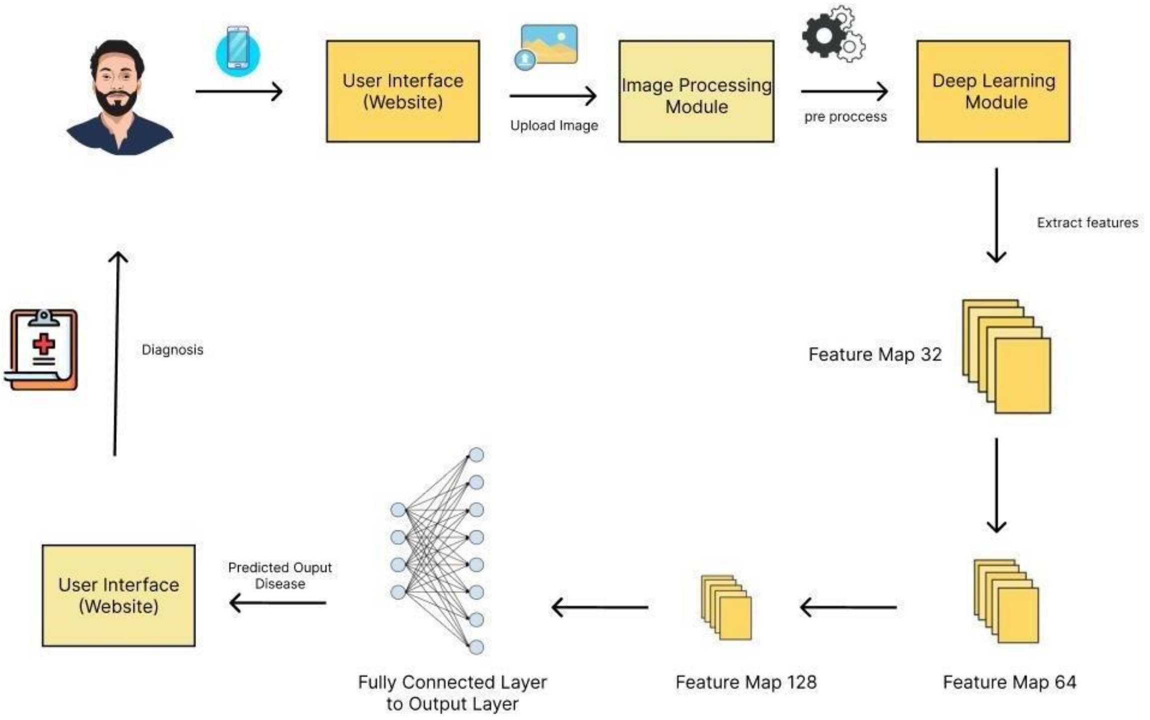


Figure 1: Architecture and data flow of the Disease Prediction Website using Image Processing of Human Nails

PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Technical Architecture:

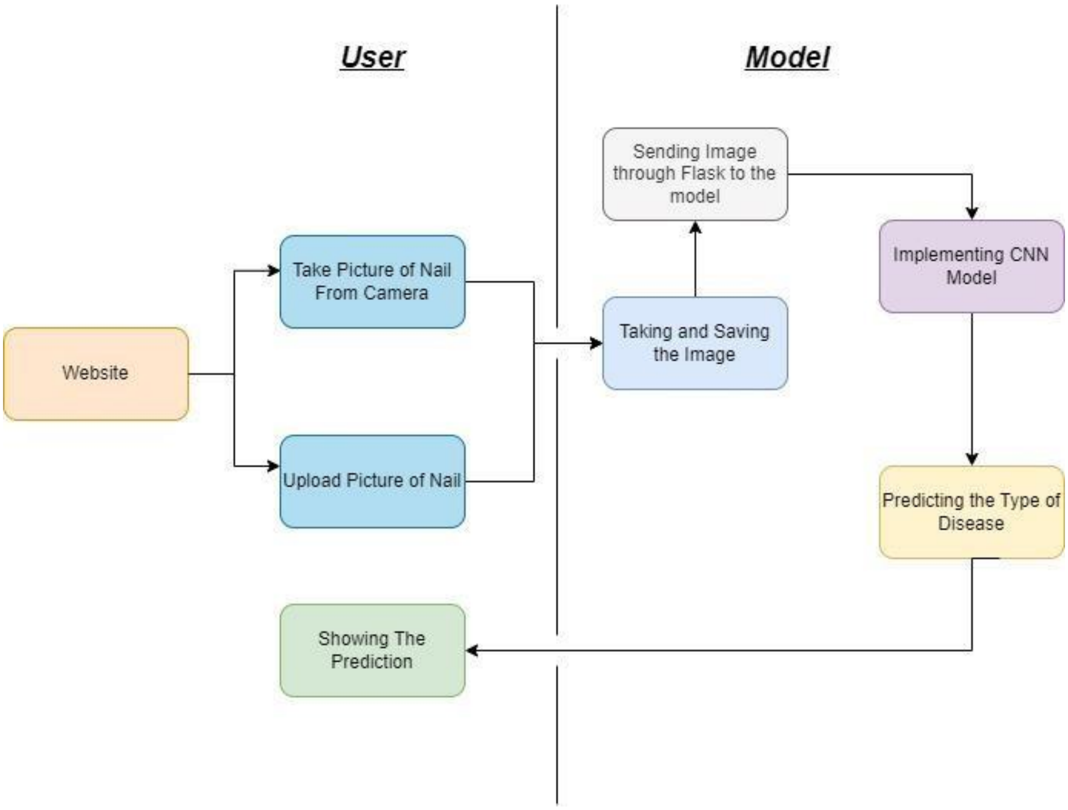

User
Model
Website
Take Picture of Nail From Camera
Upload Picture of Nail
Taking and Saving the Image
Sending Image through Flask to the model
Implementing CNN Model
Predicting the Type of Disease
Showing The Prediction

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Building CNN Model	Python
3.	Application Logic-2	Use of flask to connect from model to webapplication	Flask
4.	Application Logic-3	Use of Camera to capture Image	Camera
5.	Machine Learning Model	To Recognize the Pattern of the Image	Convolutional Neural Network, TransferLearning Models (Ex: VGG16, VGG19, ResNet-19)
6.	Infrastructure (Server / Cloud)	Application Deployment on Local System Local	VS-Code

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks refer to using software frameworks that are publicly accessible, can be modified.	TensorFlow, Keras & Flask
2.	Security Implementations	Security implementations are essential to protect patientdata and confidentiality of the image processing system. As this is a simple model it is not necessary of security implementations.	None
3.	Scalable Architecture	AWS Well-Architected/Azure Architecture providesa framework to help cloud architects build secure, high-performing, resilient, and efficient architectures.	AWS or Microsoft Azure
4.	Availability	Use of load balancers to distribute traffic, and geographically distributed servers to ensure high availability.	AWS or Microsoft Azure
5.	Performance	Performance of model involves ensures that application can handle a significant number of image processing requests per second, use ofcache and CDN's efficiently.	Utilize hardware acceleration (GPU's)

6.2 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (EPIC)	User Story Number	User Story/Task	Story Points	Priority	Team Members
print-1	Registration	USN-1	As a user, I want to fill my details like name, mail id, password to sign up so that I can have access to my account whenever I need.	3	High	Anusha
print-1		USN-2	As a user, I want to receive confirmation email once I have filled signup form so that I can verify whether email address is correct or not.	3	High	Ganesh
print-1	Login	USN-3	As a user, I want to log into the application by entering correct email & password so that I can access the portal	2	High	Sravan

print-1	Image Upload	USN-4	As a healthcare professional, I want to upload nail images for disease prediction.	2	High	Srushik
print-2	Access Account History	USN-5	As a user, I want to view my account activity and transaction history to track my usage and monitor any changes or activities within my account.	2	Medium	Anusha
print-2	Algorithm Improvement	USN-6	As a developer, I want to continuously update and refine the deep learning algorithm for enhanced disease prediction accuracy.	3	Medium	Sravan
print-2	Data Management	USN-7	As a system administrator, I want to ensure the secure storage and management of patient data and images.	3	Medium	Ganesh
print-3	Password Recovery	USN-8	As a user, I want the option to recover my account password in case I forget it, to regain access to my account.	1	Low	Srushik

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned Date)	Sprint Release Date (Actual)
Sprint-1	10	5 Days	25 Oct 2023	29 Oct 2023	10	29 Oct 2023
Sprint-2	8	4 Days	30 Oct 2023	2 Nov 2023	8	2 Nov 2023
Sprint-3	1	1 Day	3 Nov 2023	3 Nov 2023	1	3 Nov 2023

Velocity:

Average Velocity (Sprint 1) = Sprint Duration/velocity = 10/5 = 2

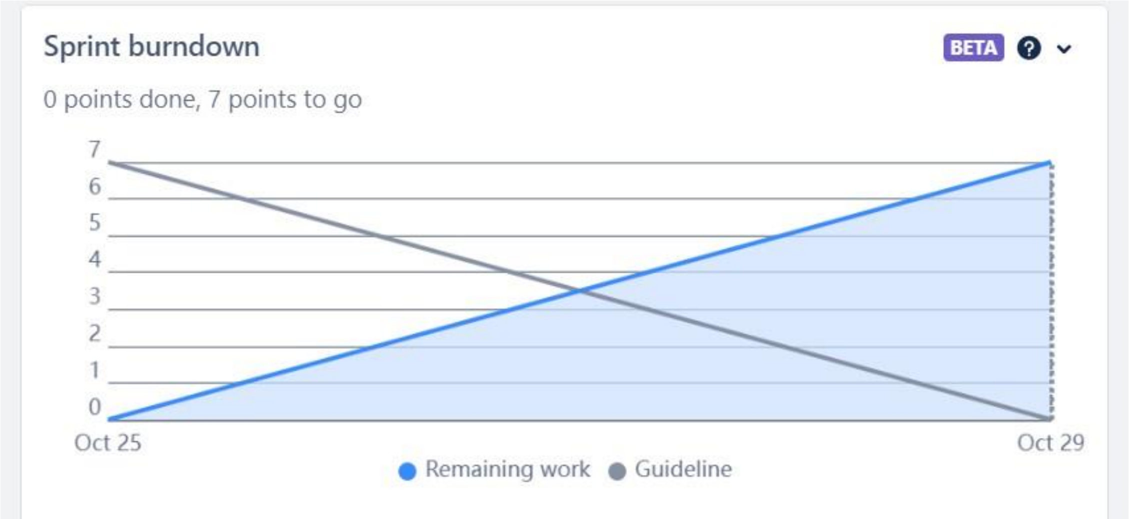
Average Velocity (Sprint 1) = Sprint Duration/velocity = 8/4 = 2

Average Velocity (Sprint 1) = Sprint Duration/velocity = 1/1 = 1

Total Average Velocity = 2

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

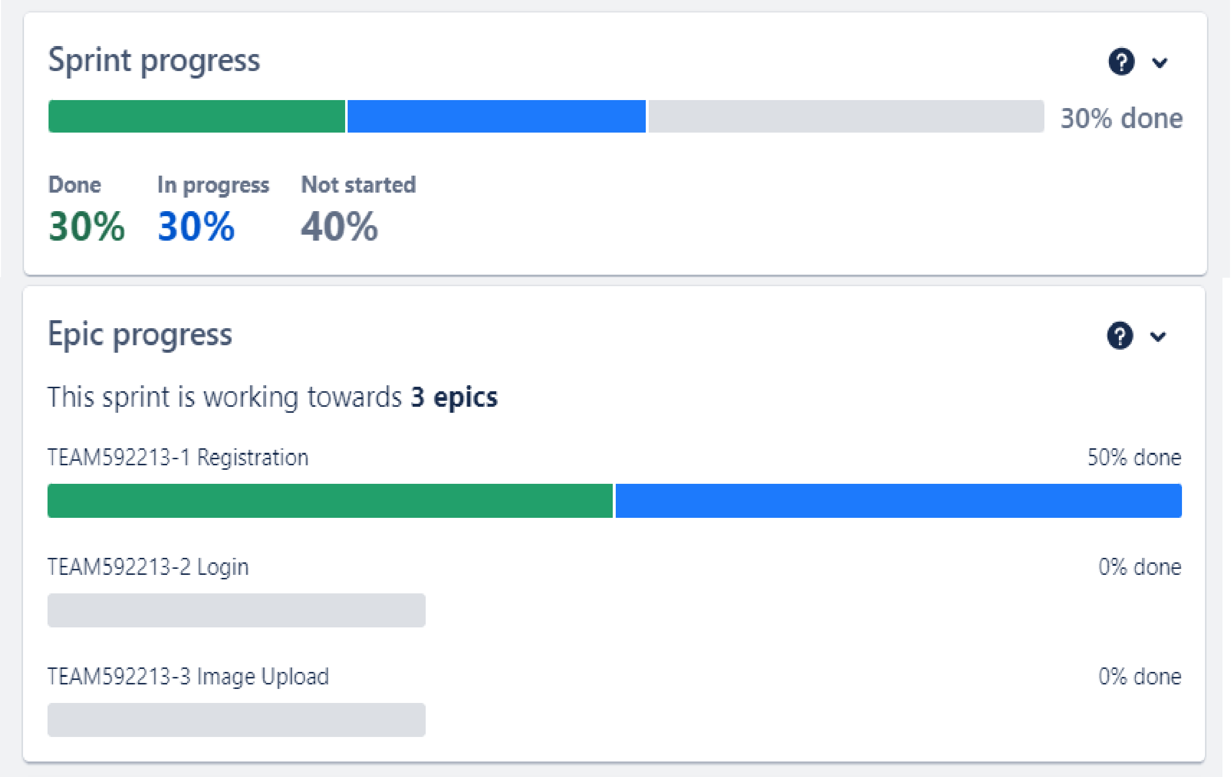


Sprint burndown

BETA

0 points done, 7 points to go

6.3 Sprint Delivery Schedule



Sprint progress

30% done

Done	In progress	Not started
30%	30%	40%

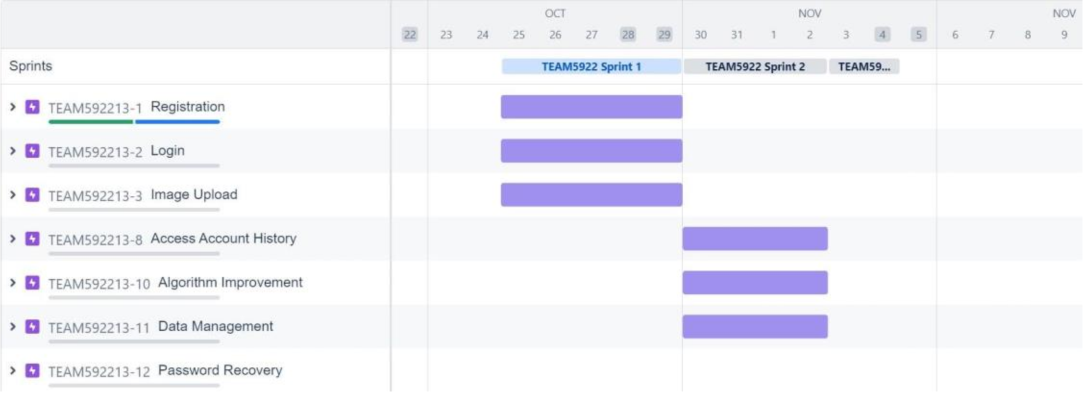
Epic progress

This sprint is working towards **3 epics**

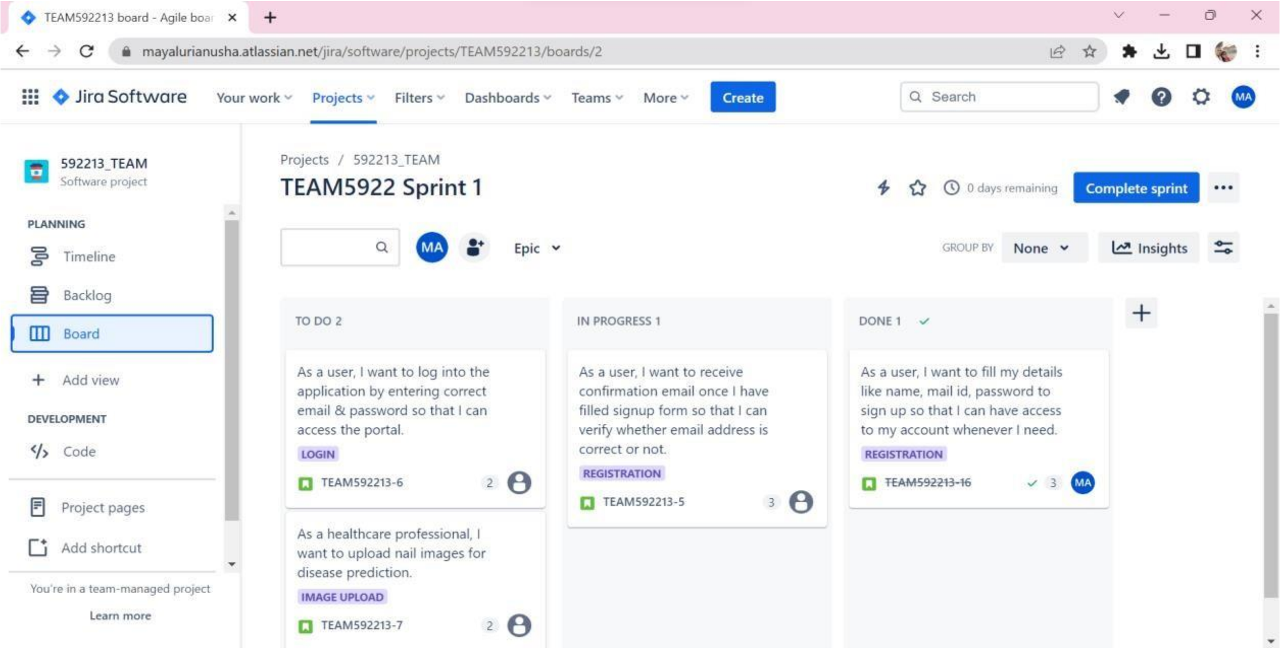
TEAM592213-1 Registration 50% done

TEAM592213-2 Login 0% done

TEAM592213-3 Image Upload 0% done



Sprints	TEAM5922 Sprint 1	TEAM5922 Sprint 2	TEAM59...
TEAM592213-1 Registration			
TEAM592213-2 Login			
TEAM592213-3 Image Upload			
TEAM592213-8 Access Account History			
TEAM592213-10 Algorithm Improvement			
TEAM592213-11 Data Management			
TEAM592213-12 Password Recovery			



TEAM592213 board - Agile boar

mayalurianusha.atlassian.net/jira/software/projects/TEAM592213/boards/2

Projects / 592213_TEAM

TEAM5922 Sprint 1

0 days remaining

Complete sprint

TO DO 2

As a user, I want to log into the application by entering correct email & password so that I can access the portal.
LOGIN
TEAM592213-6 2

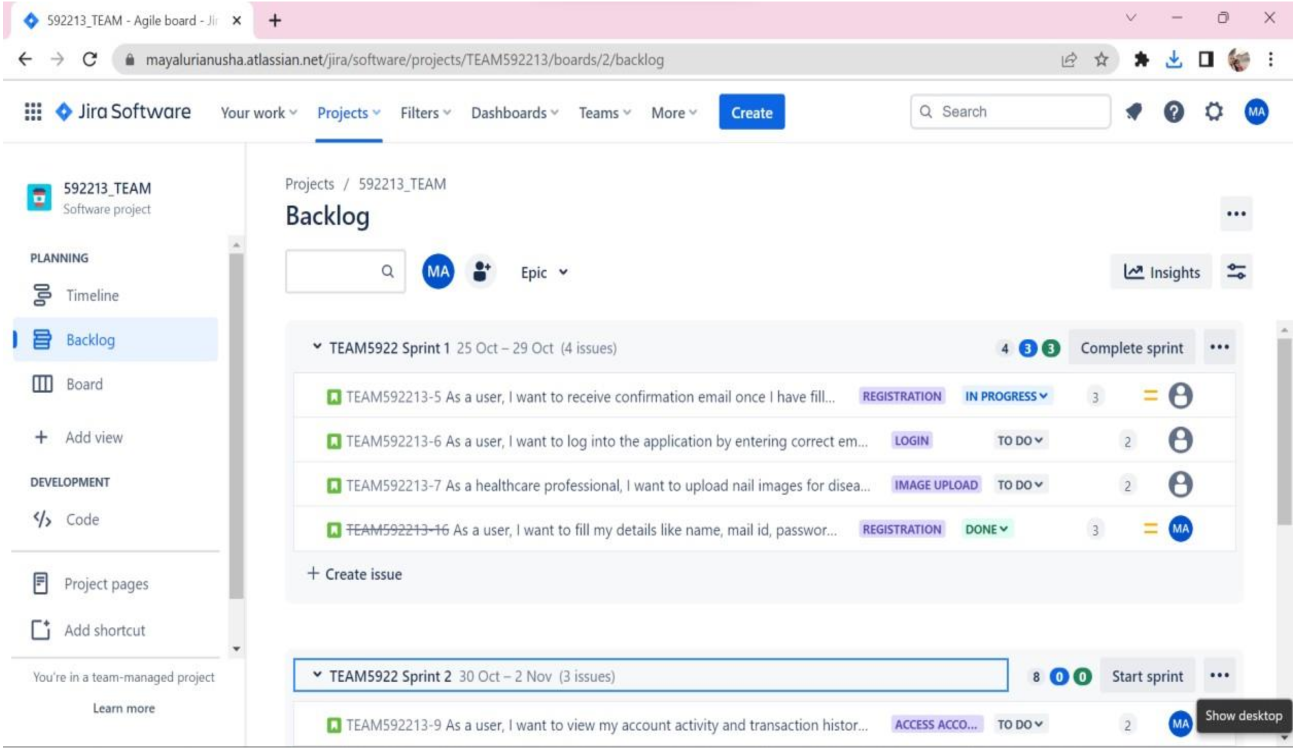
As a healthcare professional, I want to upload nail images for disease prediction.
IMAGE UPLOAD
TEAM592213-7 2

IN PROGRESS 1

As a user, I want to receive confirmation email once I have filled signup form so that I can verify whether email address is correct or not.
REGISTRATION
TEAM592213-5 3

DONE 1 ✓

As a user, I want to fill my details like name, mail id, password to sign up so that I can have access to my account whenever I need.
REGISTRATION
~~TEAM592213-16~~ ✓ 3 MA



592213_TEAM - Agile board - Ji

mayalurianusha.atlassian.net/jira/software/projects/TEAM592213/boards/2/backlog

Projects / 592213_TEAM

Backlog

TEAM5922 Sprint 1 25 Oct – 29 Oct (4 issues) 4 3 3 Complete sprint

TEAM592213-5 As a user, I want to receive confirmation email once I have fill...	REGISTRATION	IN PROGRESS	3	
TEAM592213-6 As a user, I want to log into the application by entering correct em...	LOGIN	TO DO	2	
TEAM592213-7 As a healthcare professional, I want to upload nail images for disea...	IMAGE UPLOAD	TO DO	2	
~~TEAM592213-16~~ As a user, I want to fill my details like name, mail id, passwor...	REGISTRATION	DONE	3	MA

+ Create issue

TEAM5922 Sprint 2 30 Oct – 2 Nov (3 issues) 8 0 0 Start sprint

TEAM592213-9 As a user, I want to view my account activity and transaction histor...	ACCESS ACCO...	TO DO	2	MA

CODING & SOLUTIONING

7.1 Feature 1- Model Training

Data Collection

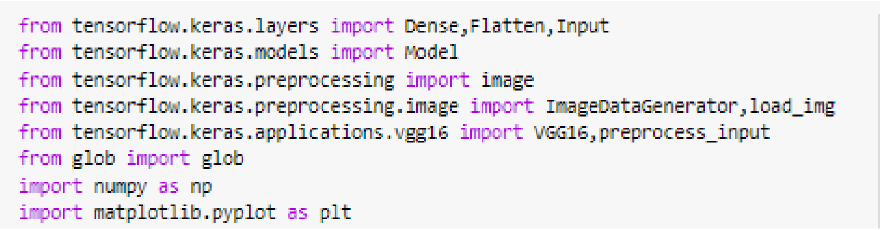
The following dataset has been used.

Dataset: https://drive.google.com/drive/folders/1AXTYsbiarS1TCAgfj0mancTSrJYYMWMs?usp=sh arin g

The dataset has already been divided into train and test folders. Therefore, we need not further divide it.

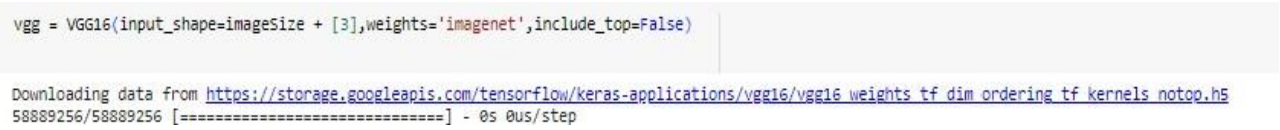
Importing Model Building Libraries

Importing the necessary libraries.



```
from tensorflow.keras.layers import Dense,Flatten,Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.applications.vgg16 import VGG16,preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
```

Loading the Model

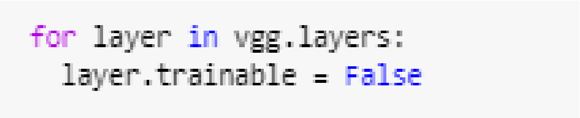


```
vgg = VGG16(input_shape=imageSize + [3],weights='imagenet',include_top=False)
```

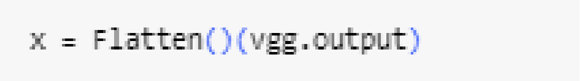
```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
```

Addition of Flatten Layers

For VGG16 model, we need to keep the Hidden layer training as false, because it hastrained weights



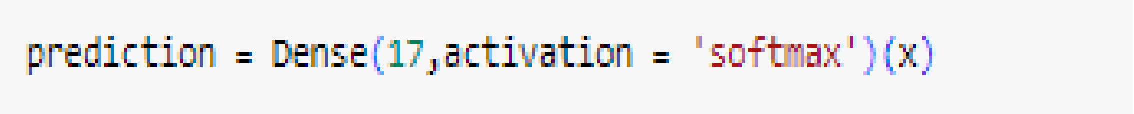
```
for layer in vgg.layers:
  layer.trainable = False
```



```
x = Flatten()(vgg.output)
```

Addition of Output Layer

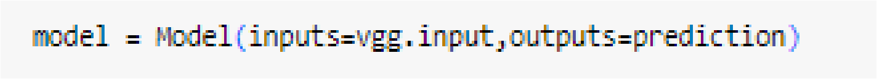
Our dataset has 17 classes, so the output layer needs to be changed as per the dataset



```
prediction = Dense(17,activation = 'softmax')(x)
```

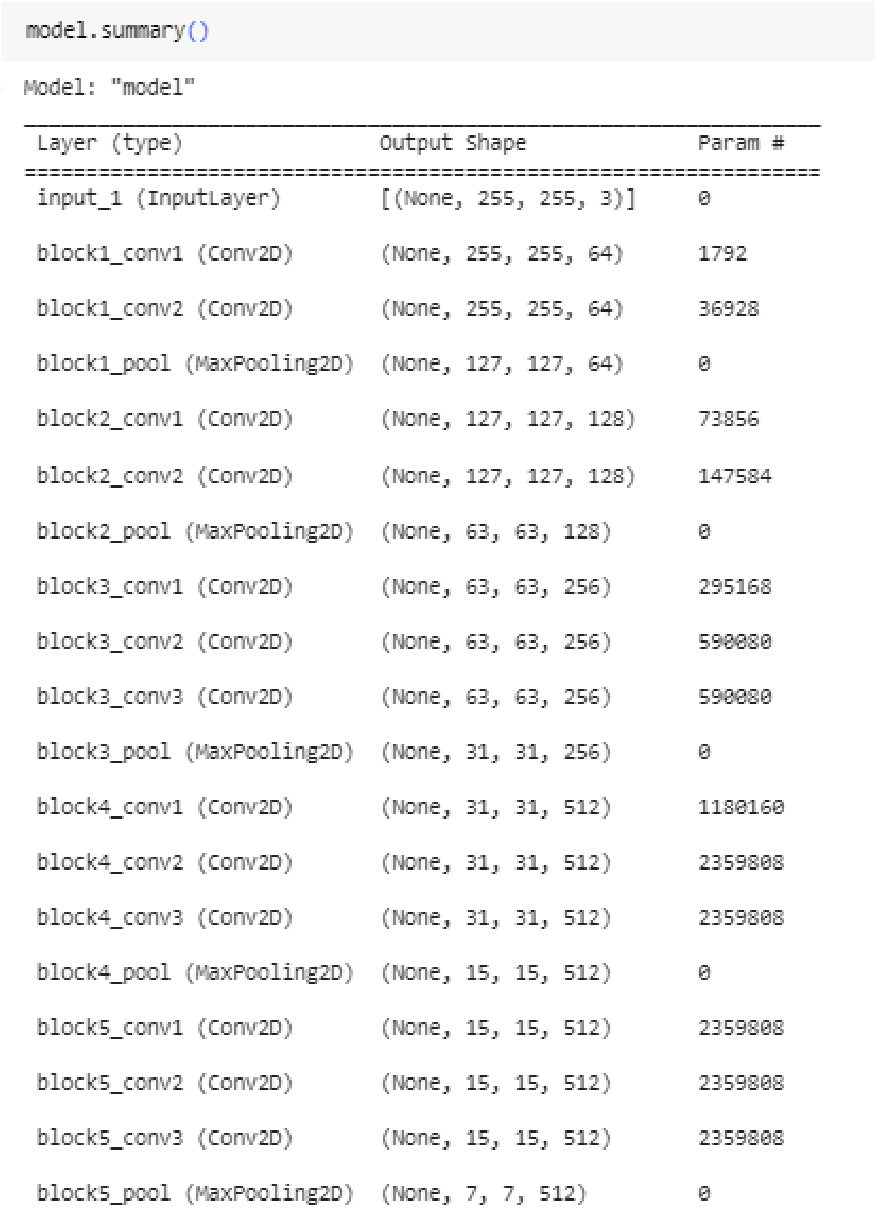
17 indicates no of classes, SoftMax is the activation function we use for categoricaloutput
Adding fully connected layer

Creation of a Model Object



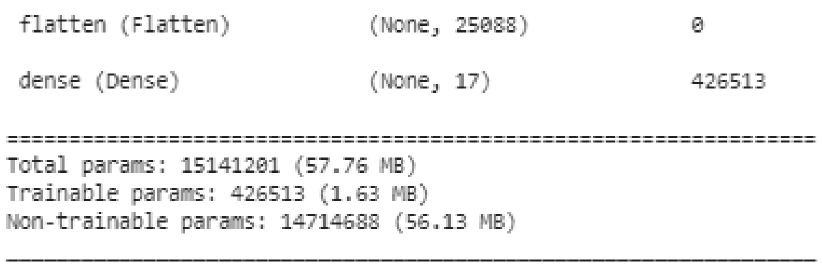
```
model = Model(inputs=vgg.input,outputs=prediction)
```

We have created inputs and outputs in the previous steps and we are creating a model and fitting to the vgg16 model, so that it will take inputs as per the given and displays the givenno of classes.
Summary of the model:



```
model.summary()
```

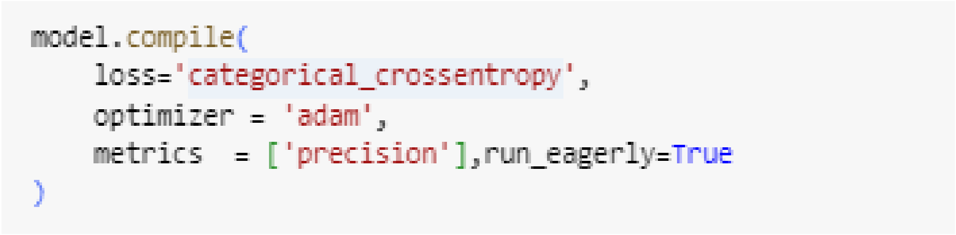
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 255, 255, 3)]     0

 block1_conv1 (Conv2D)       (None, 255, 255, 64)      1792

 block1_conv2 (Conv2D)       (None, 255, 255, 64)      36928

 block1_pool (MaxPooling2D)  (None, 127, 127, 64)      0

 block2_conv1 (Conv2D)       (None, 127, 127, 128)     73856

 block2_conv2 (Conv2D)       (None, 127, 127, 128)     147584

 block2_pool (MaxPooling2D)  (None, 63, 63, 128)       0

 block3_conv1 (Conv2D)       (None, 63, 63, 256)       295168

 block3_conv2 (Conv2D)       (None, 63, 63, 256)       590080

 block3_conv3 (Conv2D)       (None, 63, 63, 256)       590080

 block3_pool (MaxPooling2D)  (None, 31, 31, 256)       0

 block4_conv1 (Conv2D)       (None, 31, 31, 512)       1180160

 block4_conv2 (Conv2D)       (None, 31, 31, 512)       2359808

 block4_conv3 (Conv2D)       (None, 31, 31, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 15, 15, 512)       0

 block5_conv1 (Conv2D)       (None, 15, 15, 512)       2359808

 block5_conv2 (Conv2D)       (None, 15, 15, 512)       2359808

 block5_conv3 (Conv2D)       (None, 15, 15, 512)       2359808

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



```
 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 17)                426513

=================================================================
Total params: 15141201 (57.76 MB)
Trainable params: 426513 (1.63 MB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

Configuration of the Learning Process

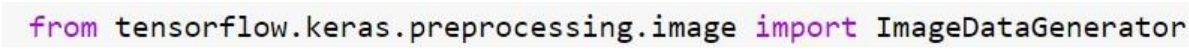
1. The compilation marks the concluding stage in the model creation process. Once compiled, the training phase can commence. The loss function is employed to identifyerrors or discrepancies within the learning process. Keras mandates the specification of a loss function during the model compilation phase.
2. Optimization represents a crucial procedure that fine-tunes the input weights by assessing the predictions against the defined loss function. In this case, the Adamoptimizer is employed for optimization purposes.
3. Metrics serve as tools for evaluating the overall performance of your model. They share similarities with the loss function; however, they are not actively involved in the training process.



```
model.compile(
    loss='categorical_crossentropy',
    optimizer = 'adam',
    metrics  = ['precision'],run_eagerly=True
)
```

Importation of the Image Data Generator Library

Within this phase, we will focus on enhancing the image data to mitigate undesirable distortions and accentuate critical image features essential for subsequent processing. This involves the implementation of various geometric transformations such as rotation, scaling,translation, and more.
Image data augmentation is a valuable technique employed to effectively increase the scale ofa training dataset by generating modified versions of the existing images within the dataset.
The Keras deep learning neural network library offers the functionality to train models usingimage data augmentation through the utilization of the ImageDataGenerator class.
Let us proceed by importing the ImageDataGenerator class from the TensorFlow Keraslibrary.



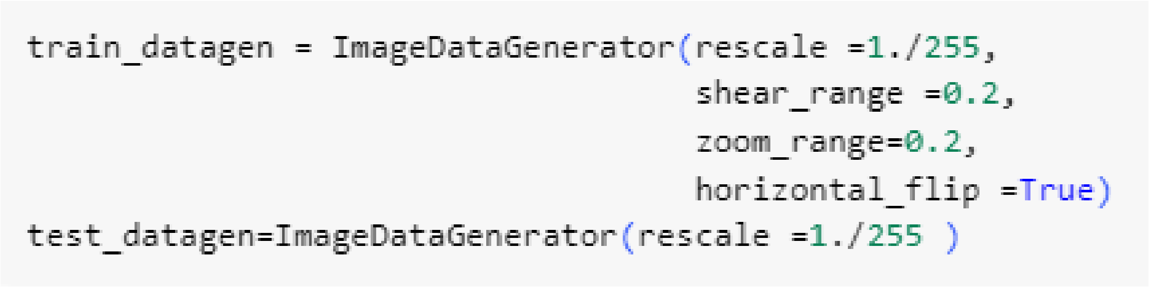
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Configuration of the Image Data Generator Class

We instantiate the ImageDataGenerator class and configure the specific data augmentation techniques. These techniques primarily include:

1. Image shifts utilizing the width_shift_range and height_shift_range arguments.
2. Image flips facilitated by the horizontal_flip and vertical_flip arguments.
3. Image rotations enabled through the rotation_range argument.
4. Image brightness adjustments managed by the brightness_range argument.
5. Image zoom functionality controlled by the zoom_range argument.

By constructing an instance of the ImageDataGenerator class, we can effectively apply these augmentation techniques to both the training and test datasets.



```
train_datagen = ImageDataGenerator(rescale =1./255,
                                   shear_range =0.2,
                                   zoom_range=0.2,
                                   horizontal_flip =True)
test_datagen=ImageDataGenerator(rescale =1./255 )
```

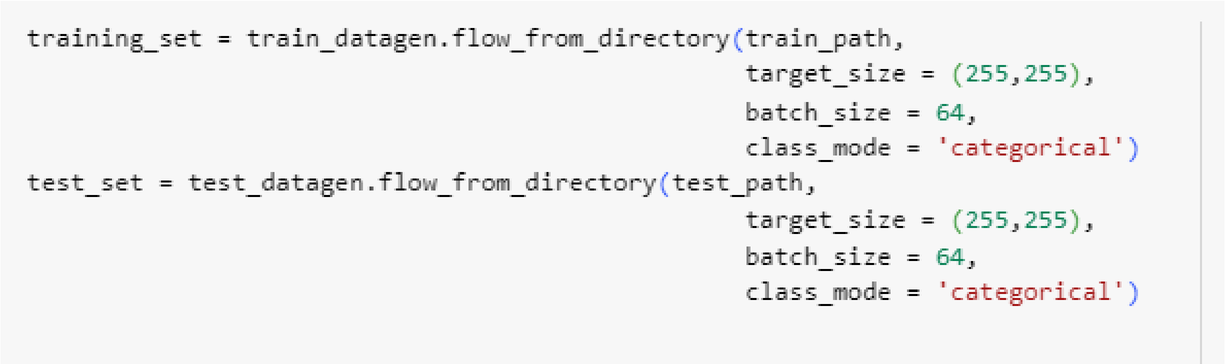
Application of Image Data Generator functionality to Trainsetand Test set

Let us proceed by implementing the ImageDataGenerator functionality to both the Training set and Test set using the code provided below. This will be accomplished by employing the "flow_from_directory" function.

The function returns batches of images from the specified subdirectories.
Arguments:

- Directory: Refers to the directory housing the dataset. If the labels are "inferred," the directory should consist of subdirectories, each containing images corresponding to aspecific class. Otherwise, the directory structure will be disregarded.
- batch_size: Denotes the size of the data batches, set to 10.
- target_size: Specifies the dimensions for resizing images post-reading from the disk.
- class_mode:
-
    - 'int': Indicates that the labels are encoded as integers (e.g., suitable forsparse_categorical_crossentropy loss).
    - 'categorical': Implies that the labels are encoded as a categorical vector (e.g.,appropriate for categorical_crossentropy loss).
    - 'binary': Signifies that the labels (limited to 2) are encoded as float32 scalarswith values 0 or 1 (e.g., used for binary_crossentropy).
    - None: Suggests the absence of labels.

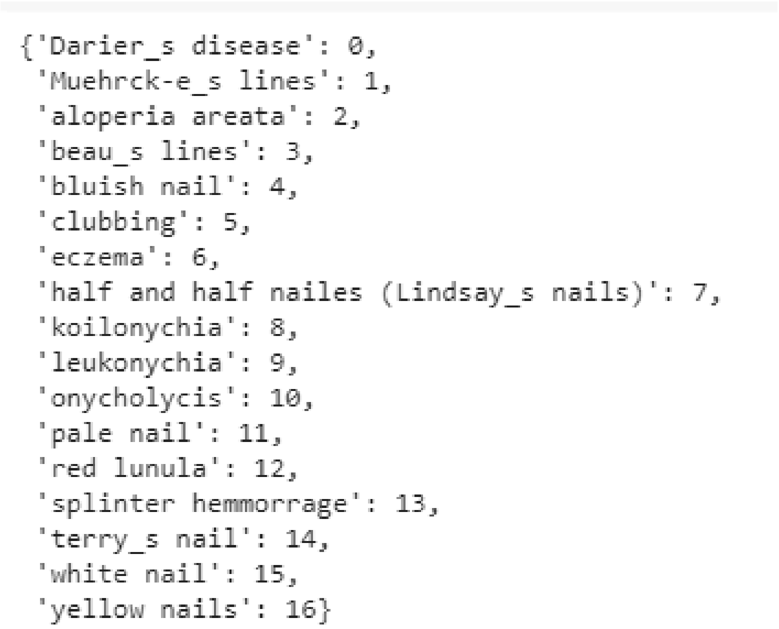
Loading our data and performing Data Augmentation



```
training_set = train_datagen.flow_from_directory(train_path,
                                                 target_size = (255,255),
                                                 batch_size = 64,
                                                 class_mode = 'categorical')
test_set = test_datagen.flow_from_directory(test_path,
                                                 target_size = (255,255),
                                                 batch_size = 64,
                                                 class_mode = 'categorical')
```

We notice that 655 images belong to 17 classes for training and 183 images belongingto 17 classes for testing purposes.

List of classes we have:



```
{'Darier_s disease': 0,
 'Muehrck-e_s lines': 1,
 'aloperia areata': 2,
 'beau_s lines': 3,
 'bluish nail': 4,
 'clubbing': 5,
 'eczema': 6,
 'half and half nailes (Lindsay_s nails)': 7,
 'koilonychia': 8,
 'leukonychia': 9,
 'onycholycis': 10,
 'pale nail': 11,
 'red lunula': 12,
 'splinter hemmorrage': 13,
 'terry_s nail': 14,
 'white nail': 15,
 'yellow nails': 16}
```

Model Training

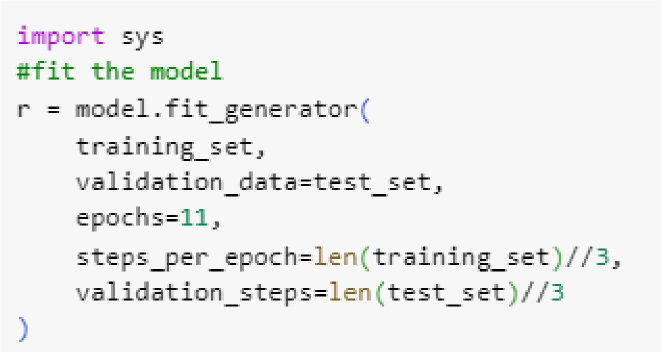
Now, we proceed to train our model utilizing the designated image dataset. The model undergoes training for a total of 11 epochs, with the current state of the model being savedafter each epoch if the encountered loss is the least recorded up to that point. Notably, the training loss exhibits a consistent decrease across almost every epoch throughout the 11- epoch training cycle, indicating potential room for further model refinement.

The "fit_generator" function is employed to facilitate the training of the deep learning neural network.

Arguments:

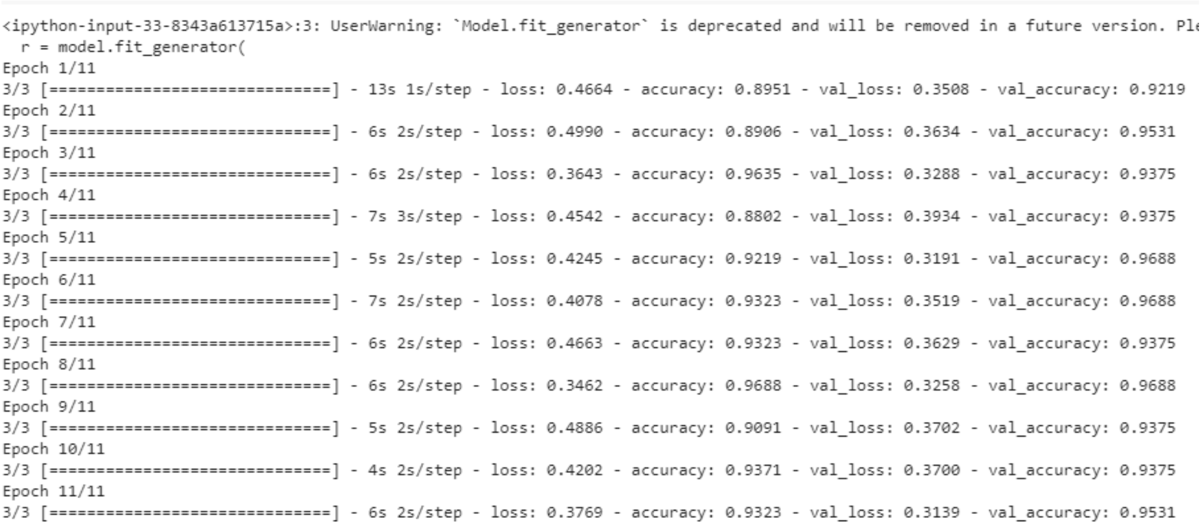
- steps_per_epoch: This parameter determines the total number of steps taken from thegenerator once an epoch is completed and the subsequent epoch begins. The value of steps_per_epoch can be calculated by dividing the total number of samples in the dataset by the batch size.

- Epochs: An integer denoting the desired number of epochs for training the model.
- Validation_data: This argument can assume one of the following forms:
    - An inputs and targets list.
    - A generator.
    - An inputs, targets, and sample_weights list, facilitating the evaluation of the lossand metrics for any model once each epoch concludes.
- Validation_steps: This argument is utilized only when the validation_data is a generator. It dictates the total number of steps taken from the generator before it is halted at the conclusion of each epoch. The value of validation_steps can be determined by dividing the total number of validation data points in the dataset by thevalidation batch size.



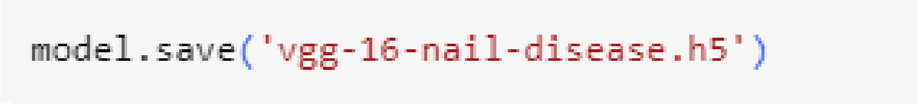
```
import sys
#fit the model
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=11,
    steps_per_epoch=len(training_set)//3,
    validation_steps=len(test_set)//3
)
```

Accuracy after 11 epochs:



```
<ipython-input-33-8343a613715a>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Pl
  r = model.fit_generator(
Epoch 1/11
3/3 [==============================] - 13s 1s/step - loss: 0.4664 - accuracy: 0.8951 - val_loss: 0.3508 - val_accuracy: 0.9219
Epoch 2/11
3/3 [==============================] - 6s 2s/step - loss: 0.4990 - accuracy: 0.8906 - val_loss: 0.3634 - val_accuracy: 0.9531
Epoch 3/11
3/3 [==============================] - 6s 2s/step - loss: 0.3643 - accuracy: 0.9635 - val_loss: 0.3288 - val_accuracy: 0.9375
Epoch 4/11
3/3 [==============================] - 7s 3s/step - loss: 0.4542 - accuracy: 0.8802 - val_loss: 0.3934 - val_accuracy: 0.9375
Epoch 5/11
3/3 [==============================] - 5s 2s/step - loss: 0.4245 - accuracy: 0.9219 - val_loss: 0.3191 - val_accuracy: 0.9688
Epoch 6/11
3/3 [==============================] - 7s 2s/step - loss: 0.4078 - accuracy: 0.9323 - val_loss: 0.3519 - val_accuracy: 0.9688
Epoch 7/11
3/3 [==============================] - 6s 2s/step - loss: 0.4663 - accuracy: 0.9323 - val_loss: 0.3629 - val_accuracy: 0.9375
Epoch 8/11
3/3 [==============================] - 6s 2s/step - loss: 0.3462 - accuracy: 0.9688 - val_loss: 0.3258 - val_accuracy: 0.9688
Epoch 9/11
3/3 [==============================] - 5s 2s/step - loss: 0.4886 - accuracy: 0.9091 - val_loss: 0.3702 - val_accuracy: 0.9375
Epoch 10/11
3/3 [==============================] - 4s 2s/step - loss: 0.4202 - accuracy: 0.9371 - val_loss: 0.3700 - val_accuracy: 0.9375
Epoch 11/11
3/3 [==============================] - 6s 2s/step - loss: 0.3769 - accuracy: 0.9323 - val_loss: 0.3139 - val_accuracy: 0.9531
```

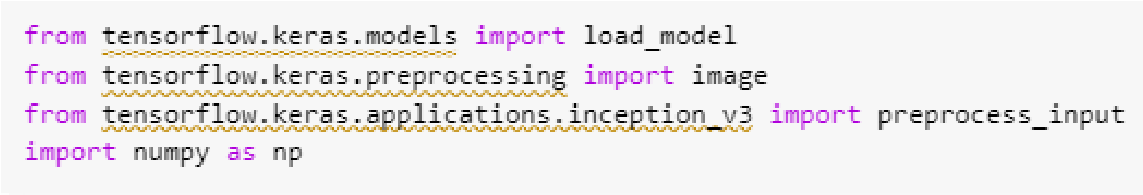
Model Saving



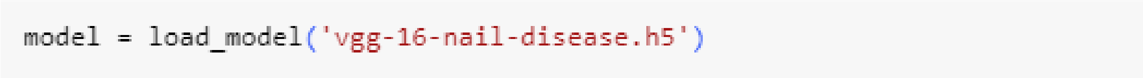
```
model.save('vgg-16-nail-disease.h5')
```

The model is saved with .h5 extension.

Evaluation is a process during the development of the model to check whether the model isthe best fit for the given problem and corresponding data. Load the saved model using load_model



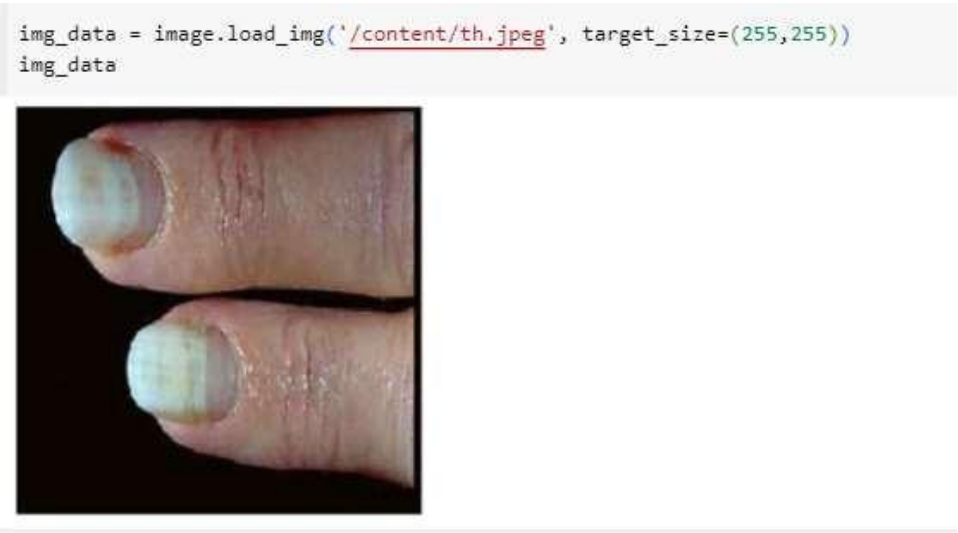
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
```



```
model = load_model('vgg-16-nail-disease.h5')
```

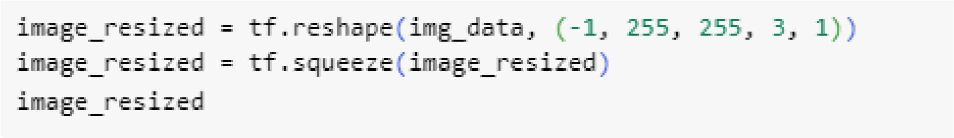
Model Testing

Taking an image as input and checking the results

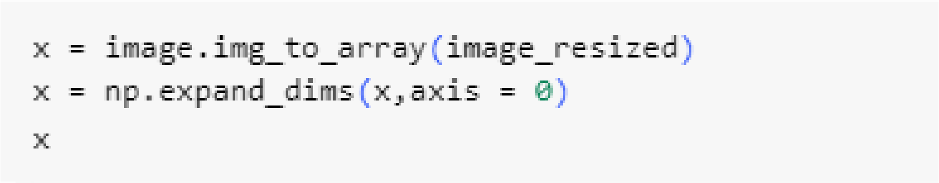


```
img_data = image.load_img('/content/th.jpeg', target_size=(255,255))
img_data
```

Resizing the image

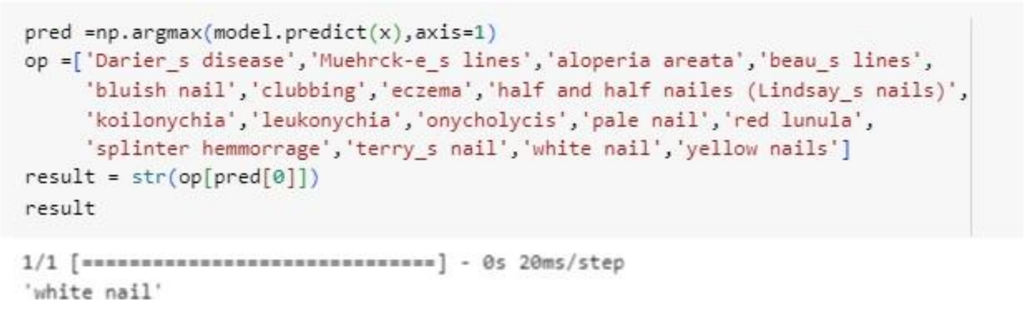


```
image_resized = tf.reshape(img_data, (-1, 255, 255, 3, 1))
image_resized = tf.squeeze(image_resized)
image_resized
```



```
x = image.img_to_array(image_resized)
x = np.expand_dims(x,axis = 0)
x
```

Predicting:



```
pred =np.argmax(model.predict(x),axis=1)
op =['Darier_s disease','Muehrck-e_s lines','aloperia areata','beau_s lines',
     'bluish nail','clubbing','eczema','half and half nailes (Lindsay_s nails)',
     'koilonychia','leukonychia','onycholycis','pale nail','red lunula',
     'splinter hemmorrage','terry_s nail','white nail','yellow nails']
result = str(op[pred[0]])
result
```

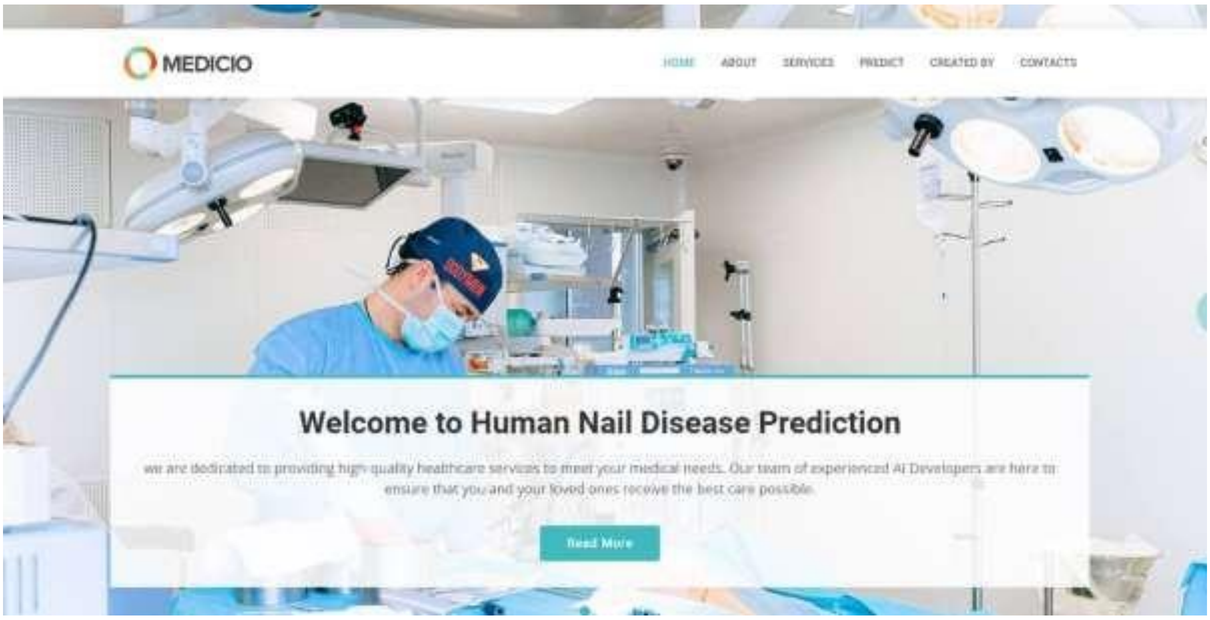
```
1/1 [==============================] - 0s 20ms/step
'white nail'
```

7.2 Feature-2 Web Interface

HTML File Creation

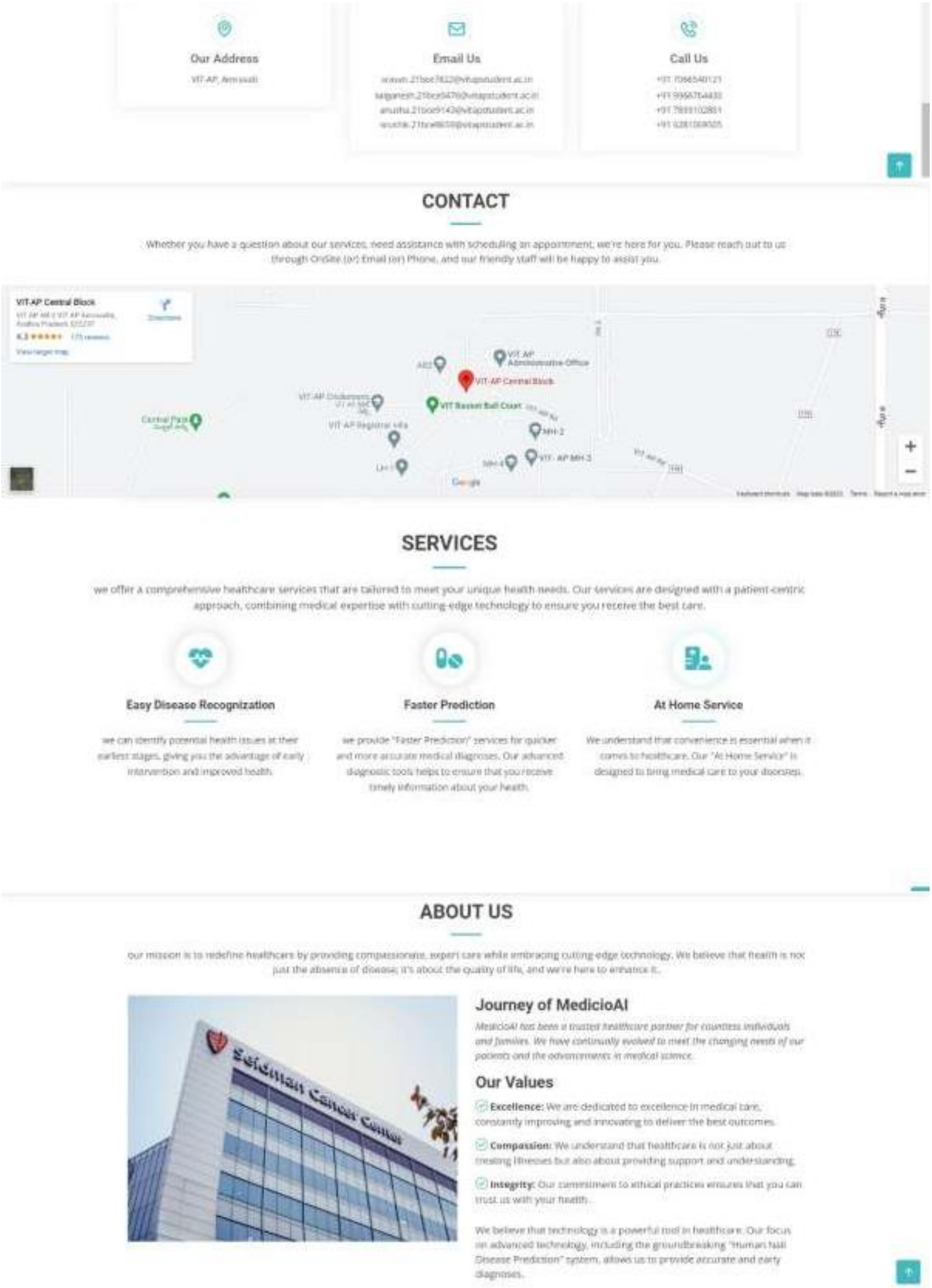
Having completed the model training, our next step involves constructing a Flask applicationthat will operate within the confines of our local browser, offering a user interface for interaction. Within the Flask application, the input parameters are extracted from the HTML page. These parameters are subsequently utilized by the model to predict the estimated cost for the incurred damage, with the results promptly displayed on the HTML page, thereby informingthe user. Upon user interaction with the UI and selection of the "Image" button, a subsequentpage is presented, enabling the user to choose the desired image and acquire the corresponding prediction output.

We created the index.html file using html



Welcome to Human Nail Disease Prediction

we are dedicated to providing high-quality healthcare services to meet your medical needs. Our team of experienced AI Developers are here to ensure that you and your loved ones receive the best care possible.



Our Address

VIT-AP, Amravati

Email Us

sravan.21bce7822@vitapstudent.ac.in
saiganesh.21bce9476@vitapstudent.ac.in
anusha.21bce9143@vitapstudent.ac.in
sruchik.21bce8659@vitapstudent.ac.in

Call Us

+91 7066540121
+91 9966764430
+91 7893102881
+91 6281069505

CONTACT

Whether you have a question about our services, need assistance with scheduling an appointment, we're here for you. Please reach out to us through OnSite (or) Email (or) Phone, and our friendly staff will be happy to assist you.

SERVICES

we offer a comprehensive healthcare services that are tailored to meet your unique health needs. Our services are designed with a patient-centric approach, combining medical expertise with cutting-edge technology to ensure you receive the best care.

Easy Disease Recognization

we can identify potential health issues at their earliest stages, giving you the advantage of early intervention and improved health.

Faster Prediction

we provide "Faster Prediction" services for quicker and more accurate medical diagnoses. Our advanced diagnostic tools helps to ensure that you receive timely information about your health.

At Home Service

We understand that convenience is essential when it comes to healthcare. Our "At Home Service" is designed to bring medical care to your doorstep.

ABOUT US

our mission is to redefine healthcare by providing compassionate, expert care while embracing cutting-edge technology. We believe that health is not just the absence of disease; it's about the quality of life, and we're here to enhance it.

Journey of MedicioAI

MedicioAI has been a trusted healthcare partner for countless individuals and families. We have continually evolved to meet the changing needs of our patients and the advancements in medical science.

Our Values

✓ **Excellence:** We are dedicated to excellence in medical care, constantly improving and innovating to deliver the best outcomes.

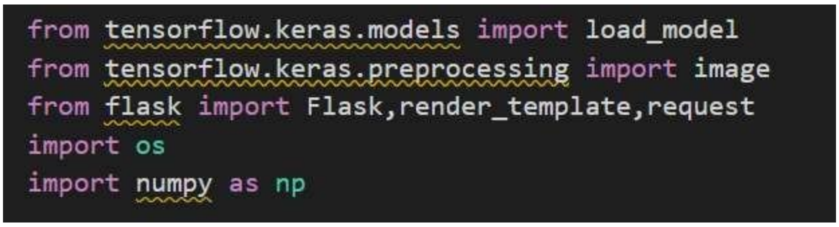
✓ **Compassion:** We understand that healthcare is not just about treating illnesses but also about providing support and understanding.

✓ **Integrity:** Our commitment to ethical practices ensures that you can trust us with your health.

We believe that technology is a powerful tool in healthcare. Our focus on advanced technology, including the groundbreaking "Human Nail Disease Prediction" system, allows us to provide accurate and early diagnoses.

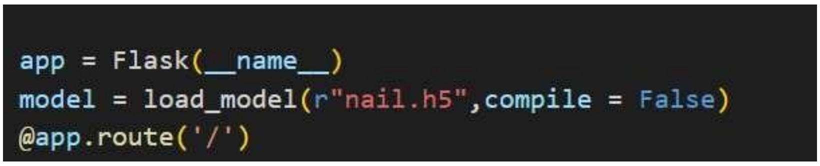
Python Code Building

Step 1: Import libraries



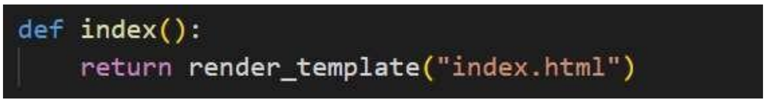
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request
import os
import numpy as np
```

Step 2 : Load our model to create flask application



```
app = Flask(__name__)
model = load_model(r"nail.h5",compile = False)
@app.route('/')
```

Step 3 : Redirect to index page

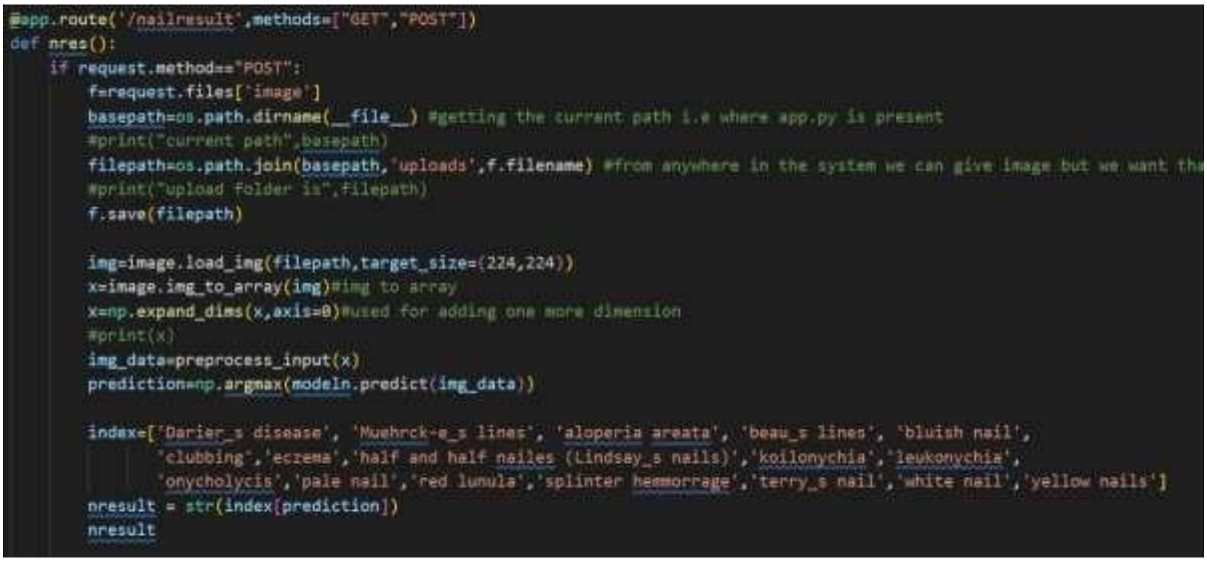


```
def index():
    return render_template("index.html")
```

Step 4 : Showcasing prediction on UI

In this section, we define a function that requests the selected file from the HTML page viathe post method. The received image file is then saved to the "uploads" folder within the same directory, utilizing the OS library. Subsequently, we employ the "load image" class from the Keras library to retrieve the saved image from the specified path. Various image processing techniques are applied to the retrieved image, which is then forwarded to the model for class prediction.

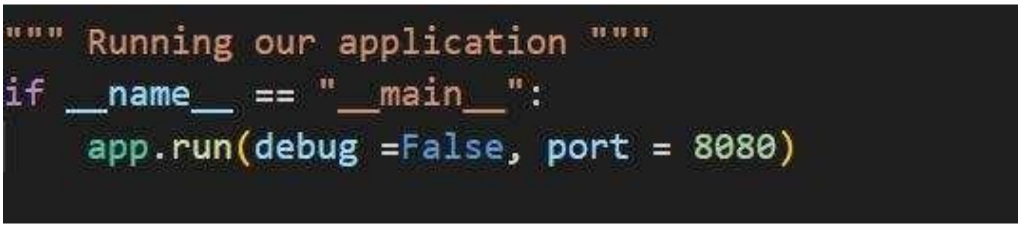
The outcome is a numerical value representing a specific class (e.g., 0, 1, 2, etc.), which resides at the 0th index of the variable "preds." This numerical value is assigned to the declared index variable. The corresponding class name is then derived and assigned to the"predict" variable, which is subsequently rendered on the HTML page for user reference.



```
@app.route('/nailresult',methods=["GET","POST"])
def nres():
    if request.method=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__) #getting the current path i.e where app.py is present
        #print("current path",basepath)
        filepath=os.path.join(basepath,'uploads',f.filename) #from anywhere in the system we can give image but we want the
        #print("upload folder is",filepath)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(224,224))
        x=image.img_to_array(img)#img to array
        x=np.expand_dims(x,axis=0)#used for adding one more dimension
        #print(x)
        img_data=preprocess_input(x)
        prediction=np.argmax(modeln.predict(img_data))

        index=['Darier_s disease', 'Muehrck-e_s lines', 'aloperia areata', 'beau_s lines', 'bluish nail',
               'clubbing','eczema','half and half nailes (Lindsay_s nails)','koilonychia','leukonychia',
               'onycholycis','pale nail','red lunula','splinter hemmorrage','terry_s nail','white nail','yellow nails']
        nresult = str(index[prediction])
        nresult
```

Application Execution

Finally, we will run the application.



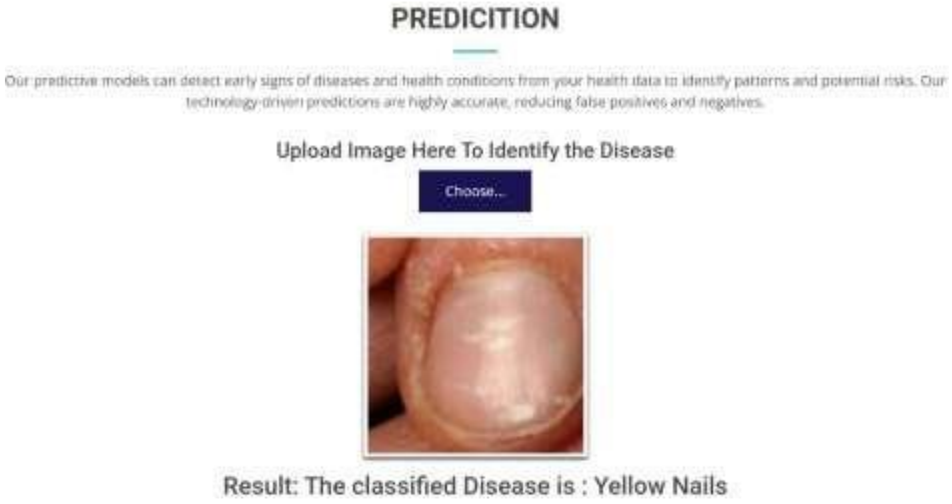
```
""" Running our application """
if __name__ == "__main__":
    app.run(debug =False, port = 8080)
```

Follow the steps outlined below to execute your Flask application:

1. Open the Anaconda prompt from the Start menu.
2. Navigate to the directory where your "app.py" file is located.
3. Type the command "python app.py" in the Anaconda prompt.
4. The local host where your application is running, typically indicated ashttp://127.0.0.1:8080/, will be displayed.
5. Copy the aforementioned local host URL and paste it into your preferred webbrowser. This action will direct you to the web page interface.
6. Proceed to input the necessary values, and subsequently click the "Predict" button toinitiate the prediction process.
7. The resulting prediction will be displayed on the web page for your observation andanalysis.

Final Output Display

Prediction 1 :

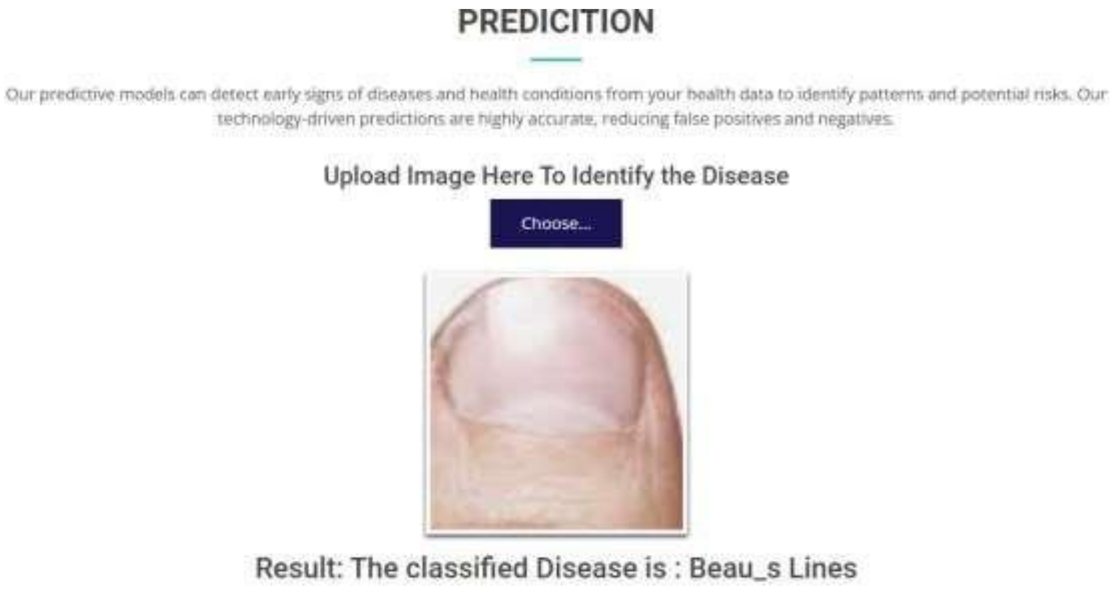

PREDICITION

Our predictive models can detect early signs of diseases and health conditions from your health data to identify patterns and potential risks. Our technology-driven predictions are highly accurate, reducing false positives and negatives.

Upload Image Here To Identify the Disease

Result: The classified Disease is : Yellow Nails

Prediction 2 :


PREDICITION

Our predictive models can detect early signs of diseases and health conditions from your health data to identify patterns and potential risks. Our technology-driven predictions are highly accurate, reducing false positives and negatives.

Upload Image Here To Identify the Disease

Result: The classified Disease is : Beau_s Lines

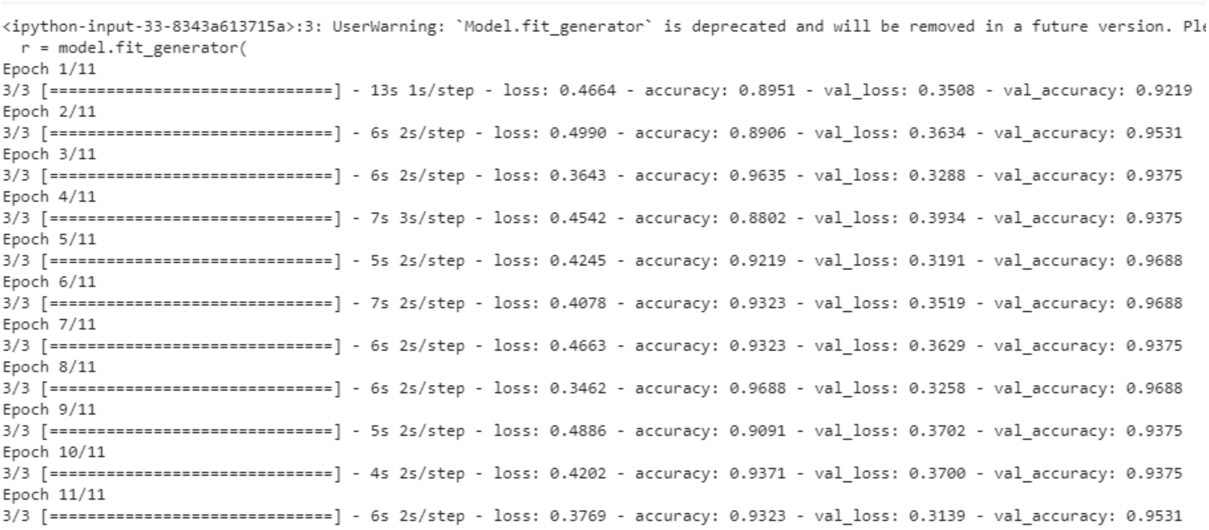
Thus, the project is ended.

PERFORMANCE TESTING

Model Performance Testing:

Accuracy : 93%

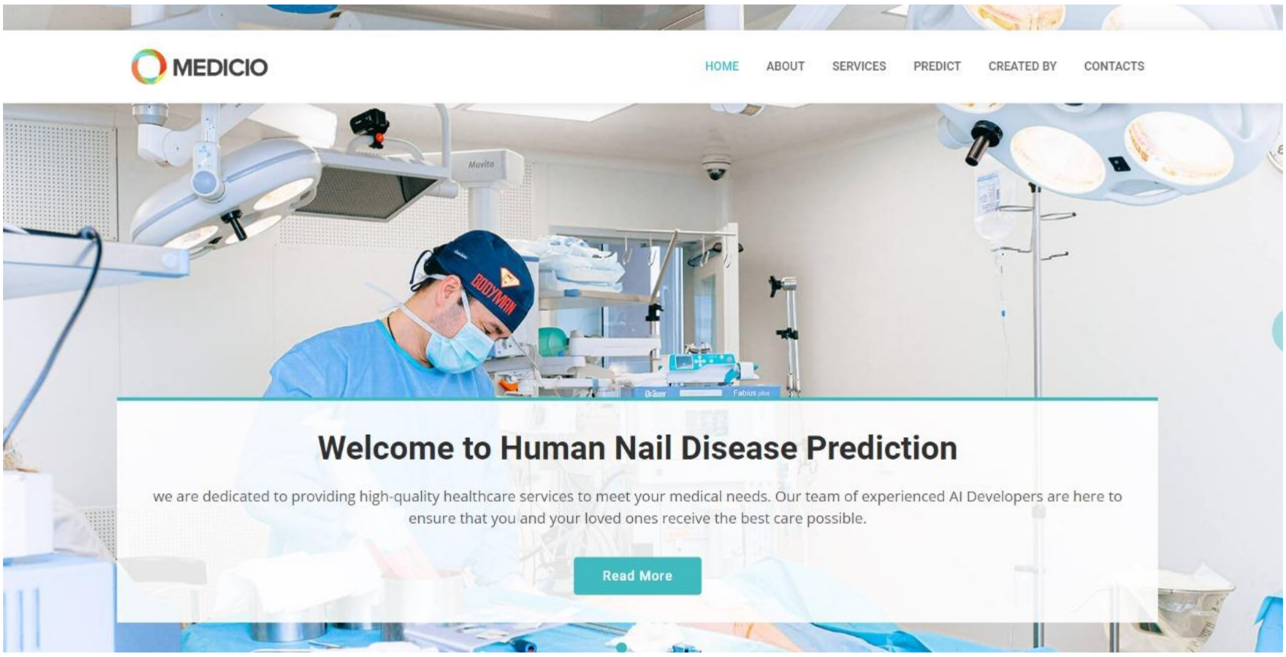
Validation accuracy : 95%



```
<ipython-input-33-8343a613715a>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Ple
  r = model.fit_generator(
Epoch 1/11
3/3 [==============================] - 13s 1s/step - loss: 0.4664 - accuracy: 0.8951 - val_loss: 0.3508 - val_accuracy: 0.9219
Epoch 2/11
3/3 [==============================] - 6s 2s/step - loss: 0.4990 - accuracy: 0.8906 - val_loss: 0.3634 - val_accuracy: 0.9531
Epoch 3/11
3/3 [==============================] - 6s 2s/step - loss: 0.3643 - accuracy: 0.9635 - val_loss: 0.3288 - val_accuracy: 0.9375
Epoch 4/11
3/3 [==============================] - 7s 3s/step - loss: 0.4542 - accuracy: 0.8802 - val_loss: 0.3934 - val_accuracy: 0.9375
Epoch 5/11
3/3 [==============================] - 5s 2s/step - loss: 0.4245 - accuracy: 0.9219 - val_loss: 0.3191 - val_accuracy: 0.9688
Epoch 6/11
3/3 [==============================] - 7s 2s/step - loss: 0.4078 - accuracy: 0.9323 - val_loss: 0.3519 - val_accuracy: 0.9688
Epoch 7/11
3/3 [==============================] - 6s 2s/step - loss: 0.4663 - accuracy: 0.9323 - val_loss: 0.3629 - val_accuracy: 0.9375
Epoch 8/11
3/3 [==============================] - 6s 2s/step - loss: 0.3462 - accuracy: 0.9688 - val_loss: 0.3258 - val_accuracy: 0.9688
Epoch 9/11
3/3 [==============================] - 5s 2s/step - loss: 0.4886 - accuracy: 0.9091 - val_loss: 0.3702 - val_accuracy: 0.9375
Epoch 10/11
3/3 [==============================] - 4s 2s/step - loss: 0.4202 - accuracy: 0.9371 - val_loss: 0.3700 - val_accuracy: 0.9375
Epoch 11/11
3/3 [==============================] - 6s 2s/step - loss: 0.3769 - accuracy: 0.9323 - val_loss: 0.3139 - val_accuracy: 0.9531
```

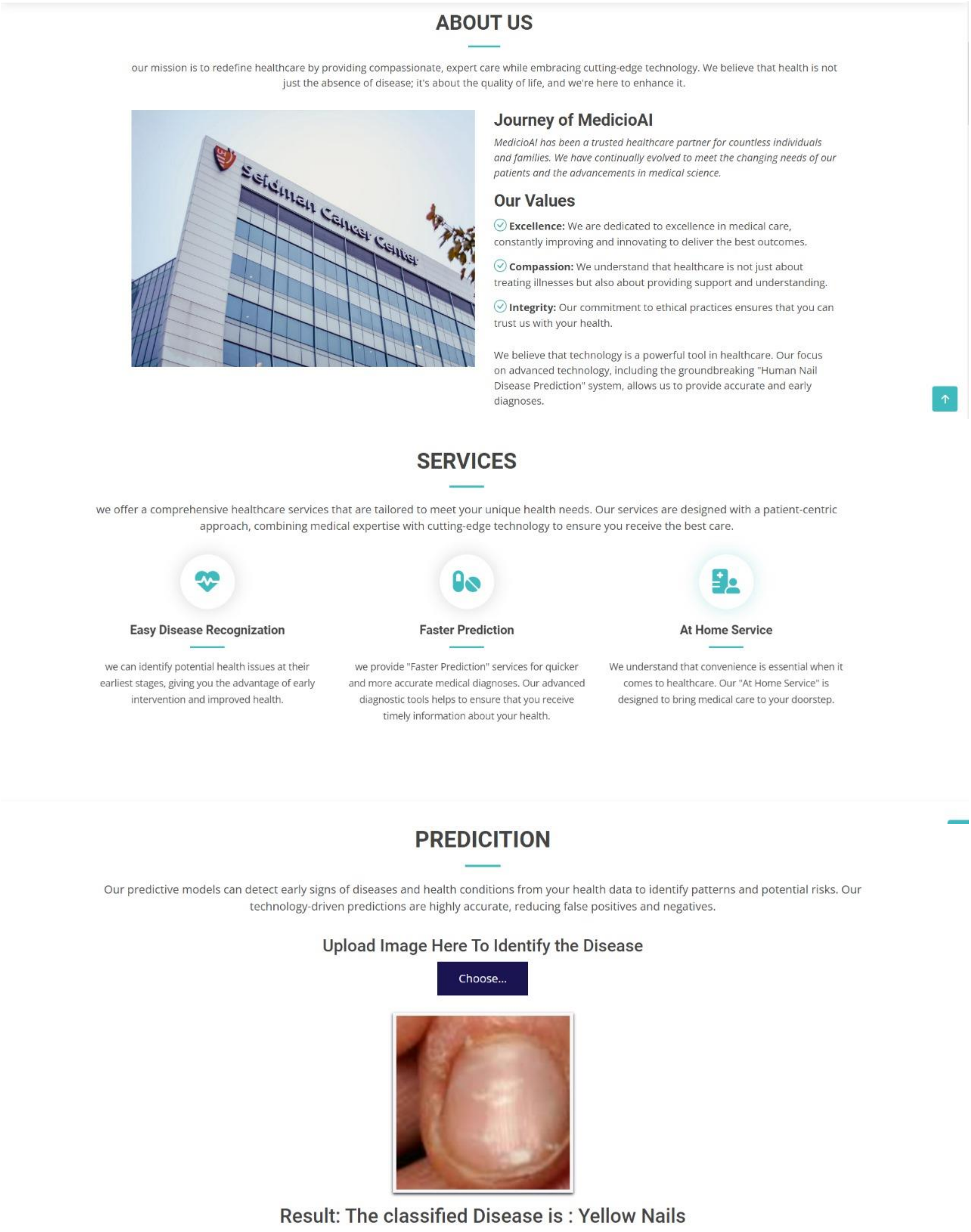
RESULTS

9.1 Output Screenshots



Welcome to Human Nail Disease Prediction

we are dedicated to providing high-quality healthcare services to meet your medical needs. Our team of experienced AI Developers are here to ensure that you and your loved ones receive the best care possible.



ABOUT US

our mission is to redefine healthcare by providing compassionate, expert care while embracing cutting-edge technology. We believe that health is not just the absence of disease; it's about the quality of life, and we're here to enhance it.

Journey of MedicioAI

MedicioAI has been a trusted healthcare partner for countless individuals and families. We have continually evolved to meet the changing needs of our patients and the advancements in medical science.

Our Values

✓ **Excellence:** We are dedicated to excellence in medical care, constantly improving and innovating to deliver the best outcomes.

✓ **Compassion:** We understand that healthcare is not just about treating illnesses but also about providing support and understanding.

✓ **Integrity:** Our commitment to ethical practices ensures that you can trust us with your health.

We believe that technology is a powerful tool in healthcare. Our focus on advanced technology, including the groundbreaking "Human Nail Disease Prediction" system, allows us to provide accurate and early diagnoses.

SERVICES

we offer a comprehensive healthcare services that are tailored to meet your unique health needs. Our services are designed with a patient-centric approach, combining medical expertise with cutting-edge technology to ensure you receive the best care.

Easy Disease Recognization

we can identify potential health issues at their earliest stages, giving you the advantage of early intervention and improved health.

Faster Prediction

we provide "Faster Prediction" services for quicker and more accurate medical diagnoses. Our advanced diagnostic tools helps to ensure that you receive timely information about your health.

At Home Service

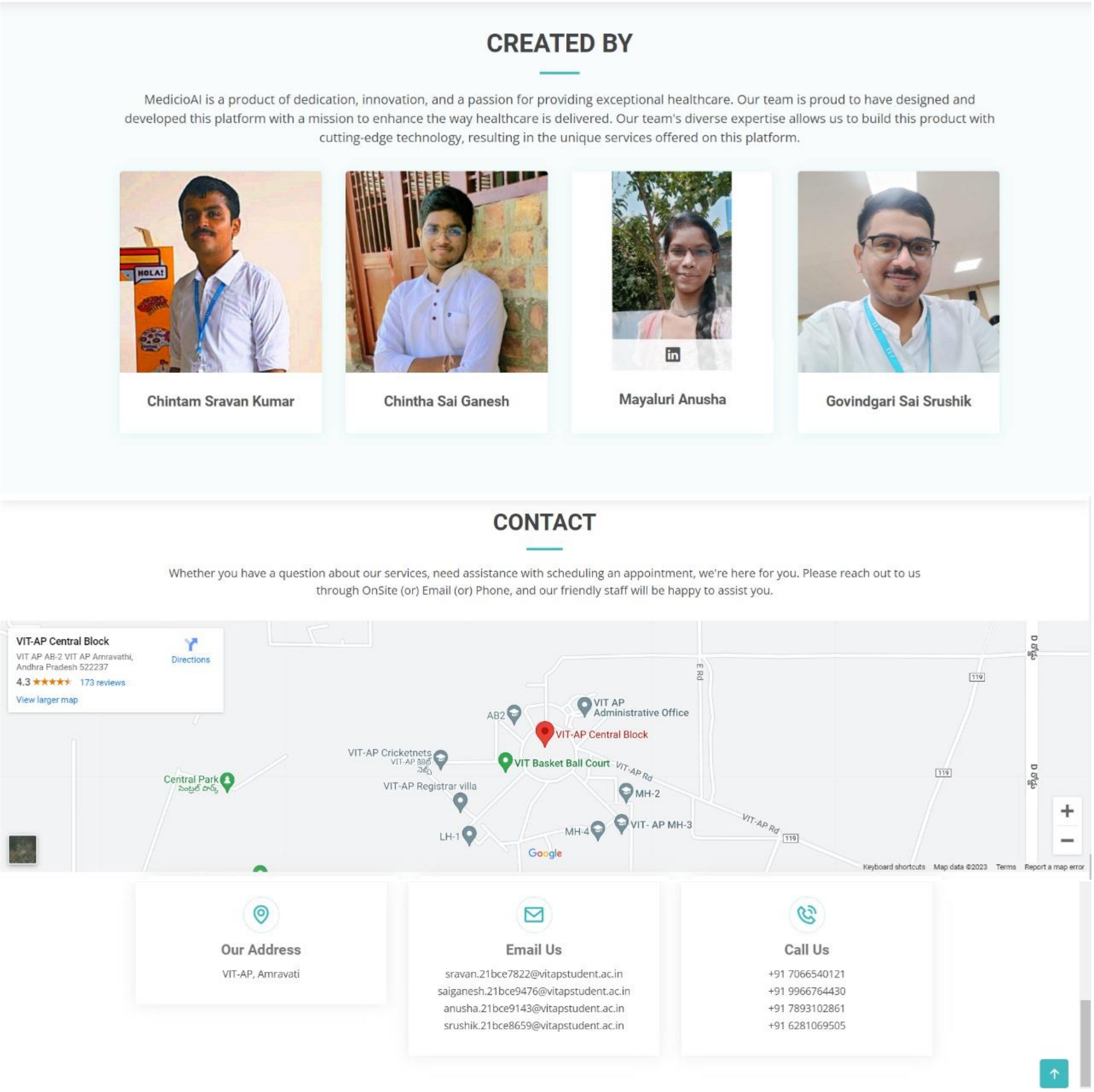
We understand that convenience is essential when it comes to healthcare. Our "At Home Service" is designed to bring medical care to your doorstep.

PREDICITION

Our predictive models can detect early signs of diseases and health conditions from your health data to identify patterns and potential risks. Our technology-driven predictions are highly accurate, reducing false positives and negatives.

Upload Image Here To Identify the Disease

Result: The classified Disease is : Yellow Nails



CREATED BY

MedicioAI is a product of dedication, innovation, and a passion for providing exceptional healthcare. Our team is proud to have designed and developed this platform with a mission to enhance the way healthcare is delivered. Our team's diverse expertise allows us to build this product with cutting-edge technology, resulting in the unique services offered on this platform.

Chintam Sravan Kumar

Chintha Sai Ganesh

Mayaluri Anusha

Govindgari Sai Srushik

CONTACT

Whether you have a question about our services, need assistance with scheduling an appointment, we're here for you. Please reach out to us through OnSite (or) Email (or) Phone, and our friendly staff will be happy to assist you.

VIT-AP Central Block
VIT AP AB-2 VIT AP Amravathi, Andhra Pradesh 522237
4.3 ★★★★★ 173 reviews
View larger map

Our Address
VIT-AP, Amravati

Email Us
sravan.21bce7822@vitapstudent.ac.in
saiganesh.21bce9476@vitapstudent.ac.in
anusha.21bce9143@vitapstudent.ac.in
srushik.21bce8659@vitapstudent.ac.in

Call Us
+91 7066540121
+91 9966764430
+91 7893102861
+91 6281069505

ADVANTAGES & DISADVANTAGES

Advantages:

1. Early Diagnosis: Using image processing of human nails allows for the early detection of diseases. This can lead to more effective treatment and better patient outcomes.
2. Non-Invasive: Unlike traditional diagnostic methods, this approach is non-invasive, making it more comfortable for patients.
3. Cost-Effective: Image processing techniques are generally cost-effective, which can make healthcare more affordable.

4. Large-Scale Screening: The method can be applied for large-scale population screening, aiding in the early identification of diseases in communities.
5. Reduced Human Error: Automation through image processing reduces the risk of human error in diagnosis.

Disadvantages:

1. Dependency of Image Quality: The accuracy of diagnosis heavily depends on the quality of the nail images. Poor quality images can lead to inaccurate results.
3. Limited to Certain Diseases: This method may not be suitable for the diagnosis of all diseases, and its application is currently limited to 17 Diseases.
4. Ethical and Privacy Concerns: Collecting and processing medical images raises privacy concerns, and ethical considerations regarding data usage must be addressed.

CONCLUSION

In conclusion, the "Early Diagnosis of Diseases Using Image Processing of Human Nails", the model developed is utilizing the VGG19 architecture, has demonstrated significant promise in the field of medical diagnostics. The advantages of early disease detection, non-invasiveness, cost-effectiveness, and reduced human error highlight the potential impact of this approach on healthcare. However, it is crucial to acknowledge the challenges and limitations associated with image quality, disease applicability, and ethical concerns.

This project not only represents a technological breakthrough but also underscores the importance of interdisciplinary collaboration between healthcare and computer science. The successful implementation of VGG19 for nail image analysis paves the way for future research and innovation in the realm of medical image processing. As we move forward, it is essential to continue refining our methodology, expanding the scope of diseases that can be diagnosed through this approach, and addressing ethical and privacy considerations to ensure the responsible use of patient data.

FUTURE SCOPE

The "Early Diagnosis of Diseases Using Image Processing of Human Nails" project opens the door to various future research avenues and enhancements. Some of the key areas of future scope include:

1. Disease Expansion: The project can be extended to include the diagnosis of a broader range of diseases, allowing for a more comprehensive and versatile diagnostic tool.
2. Enhanced Image Quality: Research on improving image quality and standardization techniques can enhance the reliability and accuracy of diagnosis.
3. Machine Learning Algorithms: Exploring other advanced machine learning algorithms and deep learning architectures can further refine the diagnostic capabilities.

4. Clinical Validation: Collaborating with healthcare professionals for clinical validation and testing of the system in real-world healthcare settings.

The future scope of this project not only promises to refine and expand its applications but also to contribute to the broader landscape of healthcare and medical imaging. By addressing these areas, the project can have a more significant and lasting impact on the early diagnosis of diseases using image processing of human nails.

APPENDIX

Demo Link:-

https://drive.google.com/file/d/15U4m9uRsBVDrQad9N7CGF8_mxAWKobFf/view?usp=sharing

Github Repo Link:-

https://github.com/smartinternz02/SI-GuidedProject-591054-1697539511

Code:

```
import tensorflow as tf
tf.__version__
!pip install -q kaggle
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle
!kaggle datasets download -d saisrushikgovindgari/nail-disease-dataset
!unzip /content/nail-disease-dataset.zip
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
train_path="/content/Nail Disease DataSet/Train"
test_path="/content/Nail Disease DataSet/Test"
x_train=[]
for folder in os.listdir(train_path):
    sub_path=train_path+"/"+folder
    for img in os.listdir(sub_path):
        image_path=sub_path+"/"+img
        img_arr=cv2.imread(image_path)
        img_arr=cv2.resize(img_arr,(255,255))
        x_train.append(img_arr)
```

```
x_test=[]

for folder in os.listdir(test_path):
    sub_path=test_path+"/"+folder
    for img in os.listdir(sub_path):
        image_path=sub_path+"/"+img
        img_arr=cv2.imread(image_path)
        img_arr=cv2.resize(img_arr,(255,255))
        x_test.append(img_arr)
train_x=np.array(x_train)
test_x=np.array(x_test)

train_x=train_x/255.0
test_x=test_x/255.0
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale
=1./255,zoom_range=0.2,shear_range= 0.2,horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale =1./255)
#Applying data augmentation for training data
training_set = train_datagen.flow_from_directory('/content/Nail Disease
DataSet/Train',
                                            target_size=(255,255),
                                            class_mode = 'categorical',
                                            batch_size = 10)

#Applying data augmentation for testing data

test_set = test_datagen.flow_from_directory('/content/Nail Disease DataSet/Test',
                                            target_size=(255,255),
                                            class_mode = 'categorical',
                                            batch_size = 10)
train_y=training_set.classes
test_y=test_set.classes
training_set.class_indices
train_y.shape,test_y.shape
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
vgg19=VGG19(input_shape=[255,255]+[3],weights='imagenet',include_top=False)
for layer in vgg19.layers:
  layer.trainable = False
x = Flatten()(vgg19.output)
output = Dense(17,activation='softmax')(x)
vgg=Model(vgg19.input,output)
vgg.summary()
vgg.compile(loss = 'sparse_categorical_crossentropy',
              optimizer = 'adam',
              metrics =['accuracy','mse'],
              run_eagerly=True)
#Early stopping to avoid overfitting of model
```

```
from tensorflow.keras.callbacks import EarlyStopping

early_stop=EarlyStopping(monitor='val_loss',mode='min',verbose=1,patience=5)
history = vgg.fit(train_x,train_y,
                validation_data=(test_x,test_y),
                epochs=17,
                batch_size=7,
                steps_per_epoch=(len(training_set))
            )
# accuracies

plt.plot(history.history['accuracy'], label='train acc')
plt.plot(history.history['val_accuracy'], label='test acc')
plt.legend()
plt.savefig('vgg-acc-rps-1.png')
plt.show()
# loss

plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label='test loss')
plt.legend()
plt.savefig('vgg-loss-rps-1.png')
plt.show()
vgg.evaluate(test_x,test_y)
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
import numpy as np



#predict
y_pred=vgg.predict(test_x)
y_pred=np.argmax(y_pred,axis=1)
#get accuracy score
print(accuracy_score(y_pred,test_y))
#get classification report
print(classification_report(y_pred,test_y))
import seaborn as sns
#get confusion matrix
print(sns.heatmap(confusion_matrix(y_pred,test_y),annot=True))
vgg.save('vgg-16-nail-disease.h5')
from tensorflow.keras.preprocessing import image
import numpy as np
img=image.load_img('/content/Nail Disease DataSet/Train/Muehrck-e_s
lines/10.PNG', target_size=(255,255))
img
resized_image = tf.reshape(img, (255, 255, 3))
resized_image = tf.squeeze(resized_image)
resized_image
x = image.img_to_array(resized_image)
x = np.expand_dims(x,axis = 0)
```

```
x
pred =np.argmax(vgg.predict(x),axis=1)
op =['Darier_s disease','Muehrck-e_s lines','aloperia areata','beau_s lines',
    'bluish nail','clubbing','eczema','half and half nailes (Lindsay_s nails)',
    'koilonychia','leukonychia','onycholycis','pale nail','red lunula',
    'splinter hemmorrage','terry_s nail','white nail','yellow nails']
result = str(op[pred[0]])
result
```