AI-based localization and classification of skin disease with erythema (NALAIYA THIRAN)

on

PROFESSIONAL READINESS FOR INNOVATION,EMPLOYABILITY AND ENTREPRENEURSHIP

Submitted by

TEAM ID: PNT2022TMID38632

TEAM MEMBERS

EZHIL RAJAN R	**420419106009**
CHANDRUNATH K	**420419106008**
GOWTHAM B	**420419106011**
SIVA SHANMUGAM K	**420419106026**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONIC AND COMMUNICATION

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600

BONAFIDE CERTIFICATE

Certified that this project report **"AI-based localization and classification of skin disease with erythema"** is the bonafide work of EZHIL RAJAN R **(420419106009),** CHANDRUNATH K **(420419106008),** GOWTHAM B **(420419106011),** SIVA SHANMUGAM K**(4**20419106026**)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**	**SIGNATURE**
Mrs. M.MALATHI	**Mr B.ELANGO ,M.E**	**Mrs.SUGANYA M.E.,**
SPOC	**Faculty Mentor**	**Internal Evaluator**
Professor,	**Assistant professor,**	**Assistant professor,**
Department of Electronics and Communication,	Department of Electronics and Communication,	Department of Electronics and Communication ,
Adhiparasakthi Engineering college,	Adhiparasakthi Engineering college,	Adhiparasakthi Engineering college,
Melmaruvathur-603 319	Melmaruvathur-603 319	Melmaruvathur-603 319

CERTIFICATION OF EVALUATION

College Code/Name : 4204 / Adhiparasakthi Engineering College

Branch/ Semester : Electronics and Communication Engineering/VII

Team ID :PNT2022TMID38632

S.No	Name of the Student & Register Number	Title of the project	Name of the Faculty mentor with Designation
1.	EZHIL RAJAN R **(420419106009),** CHANDRUNATH K **(420419106008),** GOWTHAM B**(420419106011),** SIVA SHANMUGAM K **(**420419106026**)**	**AI-based localization and classification of skin disease with erythema**	Adhiparasakthi Engineering College, Melmaruvathur-603319

The report of the project works submitted by the above students in the partialfulfillment for the award of Bachelor of Engineering in Electronics and Communication of Anna University were evaluated and confirmed to be reports of workdone by the above students and then evaluated.

Submitted for the Project Work and Viva -voce examination held on……………

INTERNAL EVALUATOR **INDUSTRY EVALUATOR**

ACKNOWLEDGEMENT

It is indeed a great pleasure and proud privilege to acknowledgement the helpand support we received from the positive minds around us in making this Endeavour a successful one the spiritual blessings of His Holiness **ARULTHIRU AMMA** and the divine guidance of **THIRUMATHI AMMA** have undoubtedly taken us to the path of victory in completing this project.

The infrastructural support with all kinds of lab facilities have been a motivating factor in this completion of project work, all because of our **CORRESPONDENTSAKTHI THIRU Dr. G. B. SENTHILKUMAR** with great pleasure we take thisopportunity to thank him.

From the academic side the constant support from our honorable **PRINCIPAL Dr. J. RAJA, Ph.D.,** has encouraged us to work hard and attain this goal of completing the project. We sincerely thank our motivating and respected **HEAD OF THE DEPARTMENT Dr. J. RAJA, Ph.D.,**

Ph.D., who have given us both moral and technical support adding experience to the job we have undertaken.

We take enormous pleasure in thanking our respected **Mrs M.Malathi M.E SPOC Mr B.Elango M.E., Ph.D., Faculty mentor, Mrs.Suganya M.E.,Internal Evaluator** who helped us in crossing obstacles in the path to our glory. We also thank other Staff members, Non-teaching members of Main Block Computer Lab. Parents and Friends who have given their constant support and motivation in all our endeavors

TABLE OF CONTENTS

CHAPTER 1

1.INTRODUCTION

Skin cancer is one of the most common types of cancer worldwide. Over the past few years, different approaches have been proposed to deal with automated skin cancer detection. Nonetheless, most of them are based only on dermoscopic images and do not take into account the patient clinical information, an important clue towards clinical diagnosis. In this work, we present an approach to fill this gap. As per recent developments in medical science, the skin cancer is considered as one of the common type disease in human body. Although the presence of melanoma is viewed as a form of cancer, it is challenging to predict it. If melanoma or other skin diseases are identified in the early stages, prognosis can then be successfully achieved to cure them. For this, medical imaging science plays an essential role in detecting such types of skin lesions quickly and accurately. The application of our approaches is to Improve skin cancer detection accuracy in medical imaging and further, can be automated using electronic devices such as mobile phones etc.

1.1 Project Overview

Now a day's people are suffering from skin diseases, more than 125 million people suffering from Psoriasis also skin cancer rate is rapidly increasing over the last few decades especially Melanoma is most diversifying skin cancer. If skin diseases are not treated at an earlier stage, then it may lead to complications in the body including spreading of the infection from one individual to the other. To overcome the above problem, we are building a model which is used for the prevention and early detection of skin cancer, psoriasis. Basically, skin disease diagnosis depends on the different characteristics like colour, shape, texture etc. Here the person can capture the images of skin and then the image will be sent the trained model. The model analyses the image and detect whether the person is having skin disease or not.

1.2 PURPOSE

Classification of a disease is difficult due to the strong similarities between common skin disease symptoms. Therefore, it would be beneficial to exploit the strengths of CAD using artificial intelligence techniques, in order to improve the accuracy of dermatology diagnosis.

CHAPTER 2

2. LITERATURE SURVEY

Initially, we have done literature survey of various base papers and research publications to arrive at the idea of the project development. It is given below:

2.1 EXISTING PROBLEM:

A neglected public health problem Skin diseases are among the most common health problems in humans. Considering their significant impact on the individual, the family, the social life of patients, and their heavy economic burden, the public health importance of these diseases is underappreciated.

2.2 REFERENCE:

1) In 2021 a paper was published by Arturo I Hernandez-Serrano, Joseph Hardwicke, Emma Pickwell-MacPherson about Real time THz imaging—opportunities and challenges for skin cancer detection in this system the purpose is to find significant differences in the THz response of the skin on the underside of the feet of healthy and diabetic subjects

2) In 2020 a paper was published by Andre GC Pacheco, Renato A Krohling about The impact of patient clinical information on automated skin cancer detection in this system the purpose is to propose a straightforward method that includes an aggregation mechanism in well-known deep learning models to combine features from images and clinical data. Last, we carry out experiments to compare the models' performance with and without using this mechanism.

3) In 2020 a paper was published by Manoj Kumar, Mohammed Alshehri, Rayed AlGhamdi, Purushottam Sharma, Vikas Deep about A de-ann inspired skin cancer detection approach using fuzzy c-means clustering in this system the purpose is to improved strategy to detect three type of skin cancers in early stages are suggested. The considered Input is a skin lesion image which by using the proposed method, the system would classify it into cancerous or non-cancerous type of skin.

4) In 2020 a paper was published by Hardik Nahata, Satya P Singh about Deep learning solutions for skin cancer detection and diagnosis in this system the purpose is to to develop a skin cancer detection CNN model which can classify the skin cancer types and help in early detection .

5) In 2020 a paper was published by Rehan Ashraf, Sitara Afzal, Attiq Ur Rehman, Sarah Gul, Junaid Baber, Maheen Bakhtyar, Irfan Mehmood, Oh-Young Song, Muazzam Maqsood about Region-of-interest based transfer learning assisted framework for skin cancer detection in this paper an intelligent Region of Interest (ROI) based system to identify and discriminate melanoma with nevus cancer by using the transfer learning approach

2.3 PROBLEM STATEMENT DEFINITION:

We're trying to find a solution to identify Skin Disease but Developed model is under training because given an image of skin, we can decompose, segment, and classify in a sequential manner which takes to Early detection of skin cancer. our study shows that CAD may also be a viable option in dermatology by presenting a novel method to sequentially combine accurate segmentation and classification models. It was to create a screening test method to help healthcare workers make a more accurate diagnosis by preventing abnormal skin from being overlooked

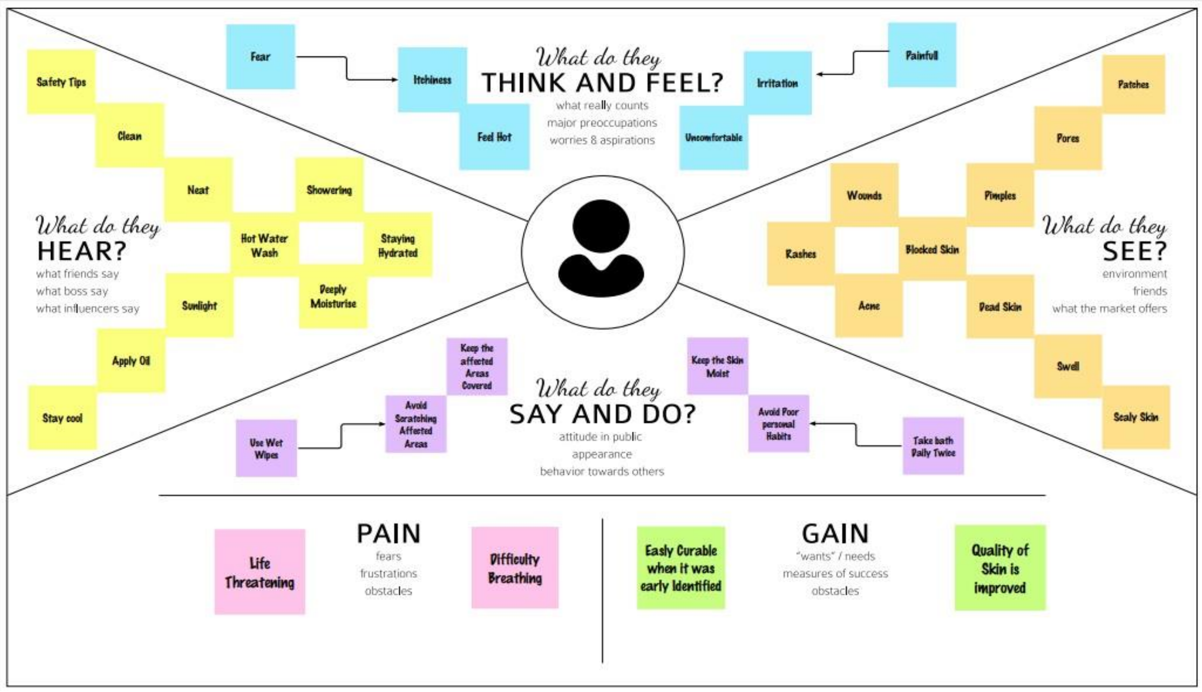
CHAPTER 3

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:

An empathy map canvas is here we say that,

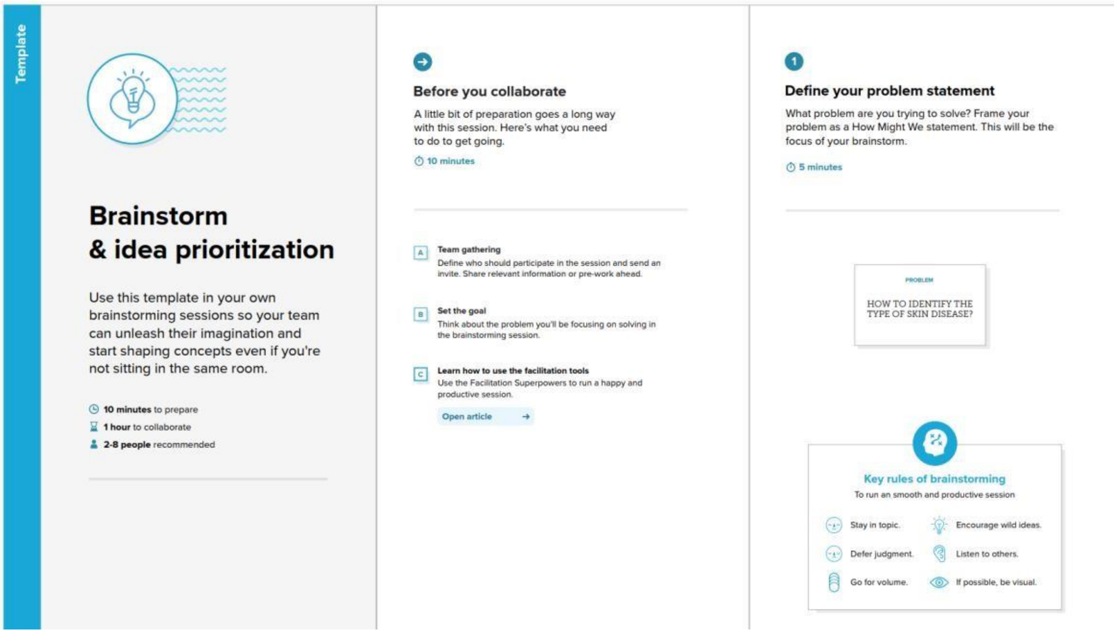
1) **What Do We Feel and think** – Uncomfortable, Irritation, Painfull, Itchiness
2) **What do we see** – Rashes, Acne, Pimples, Swell.
3) **What we say and do** - Avoid Scratching Affected Areas, Keep the affected Areas Covered, Keep the Skin Moist, Avoid Poor personal Habits , Take bath Daily Twice.
4) **What we hear** - Hot Water Wash, Sunlight ,Apply Oil, Stay cool ,Showering , Deeply Moisturise
5) **Pain** – Life Threatening, Difficulty Breathing.
6) **Gain** – Easly Curable when it was early Identified, Quality of Skin is improved.



3.2 IDEATION AND BRAINSTORMING

The Brainstorming Is the Process of Gather of The Ideas of The team members and gathering the ideas for the problem solution about the skin disease using the artificial intelligence.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

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 TO IDENTIFY THE TYPE OF SKIN DISEASE?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

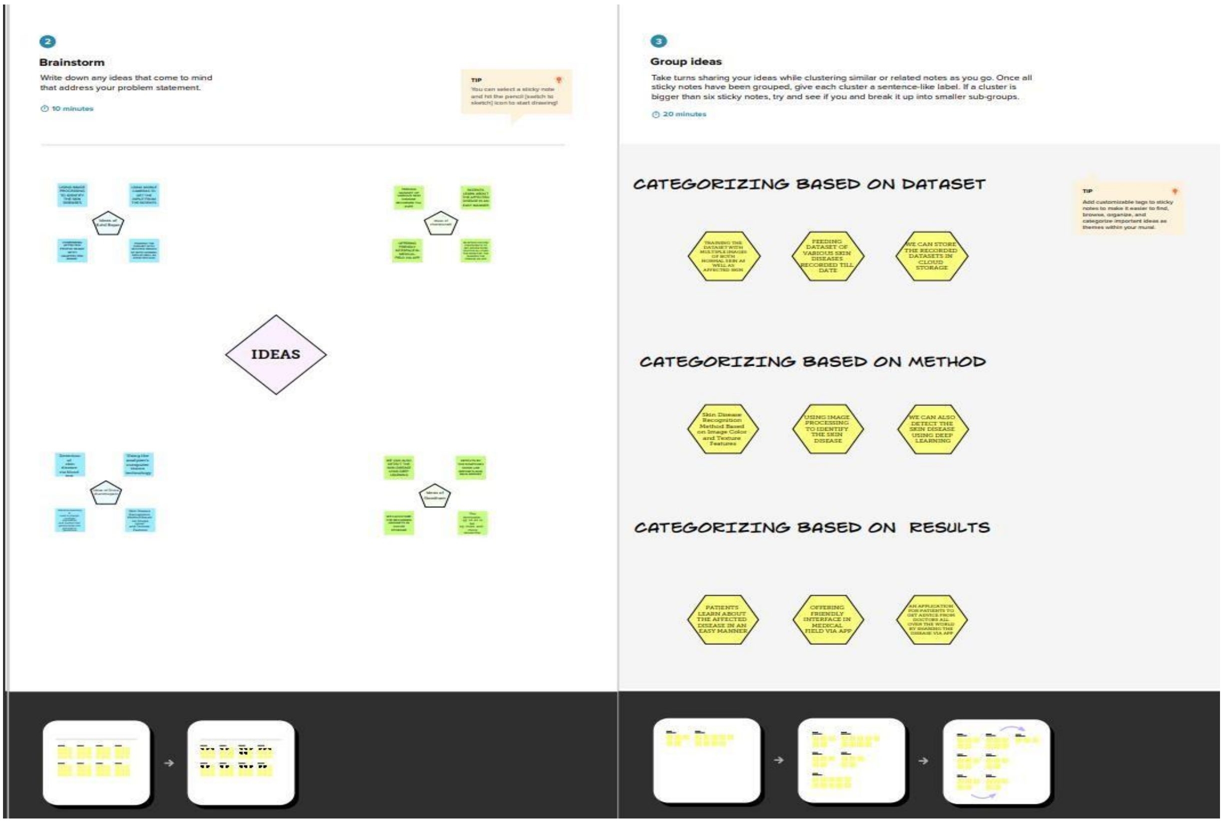
Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



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!

IDEAS

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

CATEGORIZING BASED ON DATASET

TIP

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

TRAINING THE DATASET WITH MULTIPLE IMAGES OF BOTH NORMAL SKIN AS WELL AS AFFECTED SKIN

FEEDING DATASET OF VARIOUS SKIN DISEASES RECORDED TILL DATE

WE CAN STORE THE RECORDED DATASETS IN CLOUD STORAGE

CATEGORIZING BASED ON METHOD

Skin Disease Recognition Method Based on Image Color and Texture Features

USING IMAGE PROCESSING TO IDENTIFY THE SKIN DISEASE

WE CAN ALSO DETECT THE SKIN DISEASE USING DEEP LEARNING

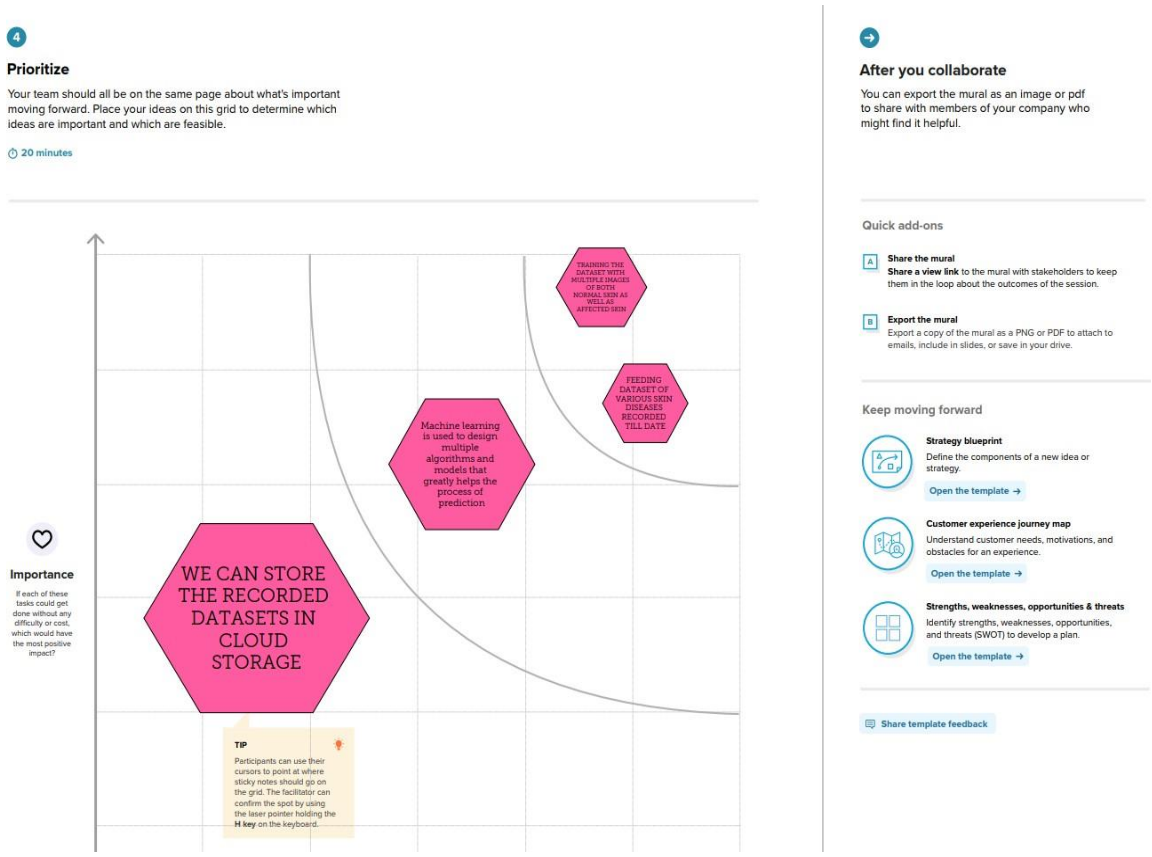
CATEGORIZING BASED ON RESULTS

PATIENTS LEARN ABOUT THE AFFECTED DISEASE IN AN EASY MANNER

OFFERING FRIENDLY INTERFACE IN MEDICAL FIELD VIA APP

AN APPLICATION FOR PATIENTS TO GET ADVICE FROM DOCTORS ALL OVER THE WORLD BY SHARING THE DISEASE VIA APP

Step-3: Idea Prioritization



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

TRAINING THE DATASET WITH MULTIPLE IMAGES OF BOTH NORMAL SKIN AS WELL AS AFFECTED SKIN

FEEDING DATASET OF VARIOUS SKIN DISEASES RECORDED TILL DATE

Machine learning is used to design multiple algorithms and models that greatly helps the process of prediction

WE CAN STORE THE RECORDED DATASETS IN CLOUD STORAGE

Importance

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

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.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

3.3 PROPOSED SOLUTION:

1) Problem Statement (Problem to be solved):

Although computer-aided diagnosis (CAD) is used to improve the quality of diagnosis in various medical fields such as mammography and colonography, it is not used in dermatology, where noninvasive screening tests are performed only with the naked eye, and avoidable inaccuracies may exist. our study shows that CAD may also be a viable option in dermatology by presenting a novel method to sequentially combine accurate segmentation and classification models. Given an image of the skin, we decompose the image to normalize and extract high-level features. Using a neural network-based segmentation model to create a segmented map of the image, we then cluster sections of abnormal skin and pass this information to a classification model. We classify each cluster into different common skin diseases using another neural network model. Our segmentation model achieves better performance compared to previous studies, and also achieves a near-perfect sensitivity score in unfavorable conditions. Our classification model is more accurate than a baseline model trained without segmentation, while also being able to classify multiple diseases within a single image. This improved performance may be sufficient to use CAD in the field of dermatology

2) Idea / Solution description:

The segmentation and classification of skin diseases has been gaining attention in the field of artificial intelligence because of its promising results. Two of the more prominent approaches for skin disease segmentation and classification are clustering algorithms and support vector machines (SVMs). Clustering algorithms generally have the advantage of being flexible, easy to implement, with the ability to generalize features that have a similar statistical variance. On experimenting with various clustering algorithms, such as fuzzy c-means, improved fuzzy c-means, and K-means, achieving approximately 83% true positive rates in segmenting a skin disease. implemented an ISODATA clustering algorithm to find the optimal threshold for the segmentation of skin lesions. An inherent disadvantage of clustering a skin disease is its lack of robustness against noise. Clustering algorithms rely on the identification of a centroid that can generalize a cluster of data. Noisy data, or the presence of outliers, can significantly degrade the performance of these algorithms. Therefore, with noisy datasets, caused by images with different types of lighting, non-clustering algorithms may be preferred; however, implemented an improved version of the fuzzy clustering algorithm using the RGB, HSV, and LAB color spaces to create a model that is more robust to noisy data. SVMs have gained attention for their effectiveness in high-dimensional data and their capability to decipher subtle patterns in noisy and complex

3) Novelty / Uniqueness:

we present a method to sequentially combine two separate models to solve a larger problem. In the past, skin disease models have been applied to either segmentation or classification. In this study, we sequentially combine both models by using the output of a segmentation model as input to a classification model. In addition, although past studies of non-CNN segmentation models used innovative preprocessing methods, recent CNN developments have focused more on the architecture of the model than on the preprocessing of data. As such, we apply an innovative preprocessing method to the data of our CNN segmentation model. The methods described above lack the ability to localize and classify multiple diseases within one image; however, we have developed a method to address this problem. Our objective is two-fold. First, we show that CAD can be used in the field of dermatology. Second, we show that state-of-the-art models can be used with current computing power to solve a wider range of complex problems than previously imagined. We begin by explaining the results of our experimentation, followed by a discussion of our findings, a more detailed description of our methodology, and finally, the conclusions that can be drawn from our study

4) Social Impact / Customer Satisfaction:

People may be affected with erythme and still will not know the seriousness they will not know that you have to see a doctor for consulting and treat it at the start .our project will use Computer-aided diagnosis (CAD) is a computer-based system that is used in the medical imaging field to aid healthcare workers in their diagnoses. CAD has become a mainstream tool in several medical fields such as mammography and colonography. However, in dermatology, although skin disease is a common disease, one in which early detection and classification is crucial for the successful treatment and recovery of patients, dermatologists perform most noninvasive screening tests only with the naked eye. So this may be a great idea that will help our society

5) **Business Model (Revenue Model):**

As our main objective was to demonstrate the viability of CAD, the performance was mostly determined using pixel-level sensitivity rather than the Intersection over Union or the Dice coefficient metrics that are often used to measure segmentation performance. Moreover, we mainly focused on the true positive rates of segmentation, represented by the sensitivity metric. This is because our aim was to create a screening test method to help healthcare workers make a more accurate diagnosis by preventing abnormal skin from being overlooked. Nevertheless, we also measured the performance of our model using the specificity, Dice coefficient, and Hausdorff distance to provide a more complete performance comparison. We measured these metrics by comparing the output from our U-Net model to an image that was masked by professional dermatologists. Going through each pixel, if a pixel of the U-Net output was black and the pixel of the dermatologist-masked image at the same location was black, this is seen as a true negative. If both were white, this was seen as a true positive. If the U-Net output was black but the dermatologist mask was white, this was seen as a false negative, and the converse was a false positive.

6) **Scalability of the Solution:**

We have shown that even without a large dataset and high-quality images, it is possible to achieve sufficient accuracy rates. In addition, we have shown that current state-of-the-art CNN models can outperform models created by previous research, through proper data preprocessing, self-supervised learning, transfer learning, and special CNN architecture techniques. Furthermore, with accurate segmentation, we gain knowledge of the location of the disease, which is useful in the preprocessing of data used in classification, as it allows the CNN model to focus on the area of interest. Lastly, unlike previous studies, our method provides a solution to classify multiple diseases within a single image. With higher quality and a larger quantity of data, it will be viable to use state-of-the-art models to enable the use of CAD in the field of dermatology.

3.4 PROBLEM SOLUTION FIT:

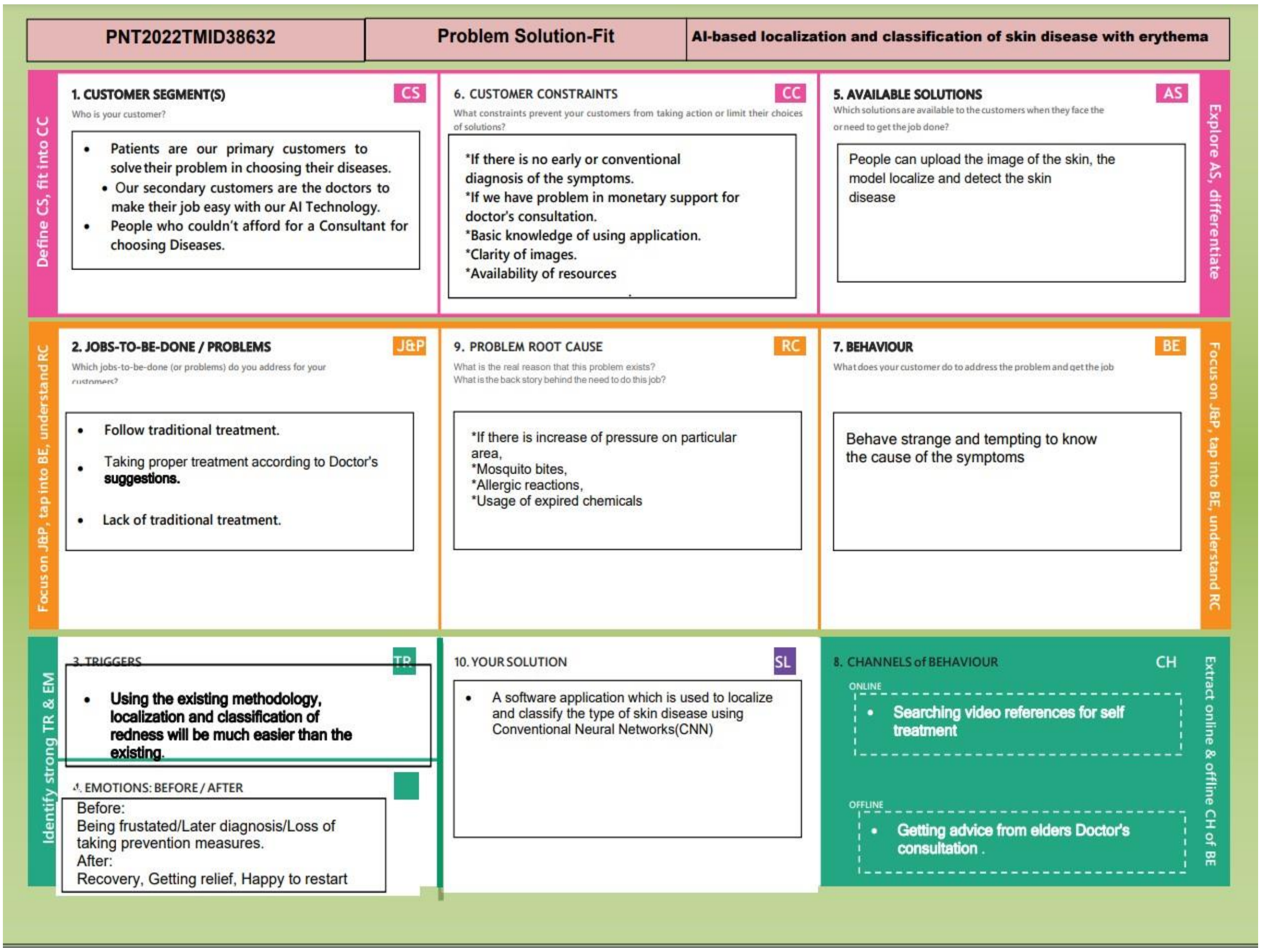
The Problem-Solution Fit simply means that you have found a problem with your patient and that the solution you have realized for it actually solves the patient's problem. It helps doctor and patient identify the skin disease and give an appropriate solution

3.4.1 PURPOSE:

Skin disease can appear in virtually any part of body and there is a lack of data required to form an association between the probability of a skin disease based on the body part. A Solution model used for the prevention and early detection of skin cancer and psoriasis by image analyses to detect whether the person is having skin disease or not. The location of the disease that is present in an image and improved performance by CNN model to focus on particular subsections of the images.

TEMPLATE:

This template defines the solution fit for the problem statement . they are various features that are used to define the solution fit like customer segments , jobs to be done , triggers , emotion before and after ,customer constraints ,problems root cause ,a variable solutions ,behaviour ,channel and what is our final solution



PNT2022TMID38632 | Problem Solution-Fit | AI-based localization and classification of skin disease with erythema

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

- Patients are our primary customers to solve their problem in choosing their diseases.
- Our secondary customers are the doctors to make their job easy with our AI Technology.
- People who couldn't afford for a Consultant for choosing Diseases.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

*If there is no early or conventional diagnosis of the symptoms.
*If we have problem in monetary support for doctor's consultation.
*Basic knowledge of using application.
*Clarity of images.
*Availability of resources

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the or need to get the job done?

People can upload the image of the skin, the model localize and detect the skin disease

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

- Follow traditional treatment.
- Taking proper treatment according to Doctor's **suggestions.**
- Lack of traditional treatment.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?

*If there is increase of pressure on particular area,
*Mosquito bites,
*Allergic reactions,
*Usage of expired chemicals

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job

Behave strange and tempting to know the cause of the symptoms

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- **Using the existing methodology, localization and classification of redness will be much easier than the existing.**

4. EMOTIONS: BEFORE / AFTER

Before:
Being frustated/Later diagnosis/Loss of taking prevention measures.
After:
Recovery, Getting relief, Happy to restart

10. YOUR SOLUTION SL

- A software application which is used to localize and classify the type of skin disease using Conventional Neural Networks(CNN)

8. CHANNELS of BEHAVIOUR CH

ONLINE

- **Searching video references for self treatment**

OFFLINE

- **Getting advice from elders Doctor's consultation .**

Extract online & offline CH of BE

CHAPTER 4

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

1) **User Registration :** Registration through Form,Gmail,LinkedIN.
2) **User Confirmation :** Confirmation via Email,OTP
3) **User Profile :** User will provide their medical details and save in the system.
4) **Input :** User capture the skin which is affected or upload the taken image as jpeg format
5) **Output Analysis :** Uploaded images is compared with the pre-defined Model and solution is generated.
6) **Provides Description :** Gives detailed description of type of the skin disease affected

4.2 NON-FUNCTIONAL REQUIREMENTS:

1) **Usability:** Used to classify skin disease with erythema

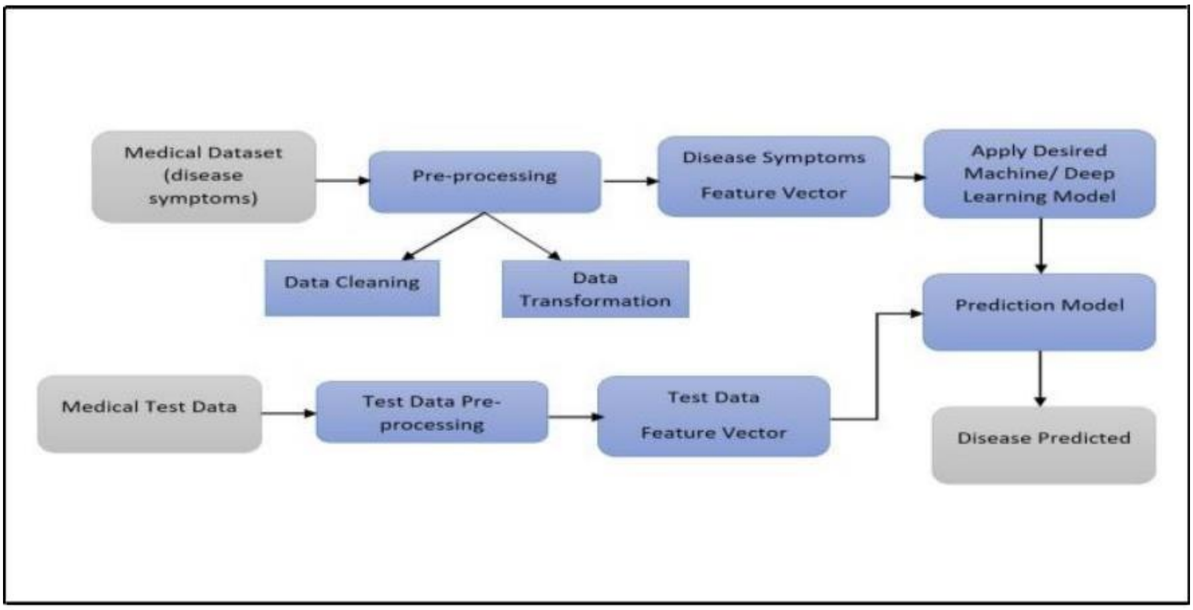
2) **Security:** It offers greater security and prevents unauthorized users to access the data
3) **Reliability:** Even with more users, there will be a good performance without failure.
4) **Performance:** Performance is very high and it provides result with high accuracy and precision.
5) **Availability:** With a good system, all authorised users can access and view the medical reports of patients.
6) **Scalability:** Performance will be good even with high user traffic.

CHAPTER 5

5. PROJECT DESIGN:

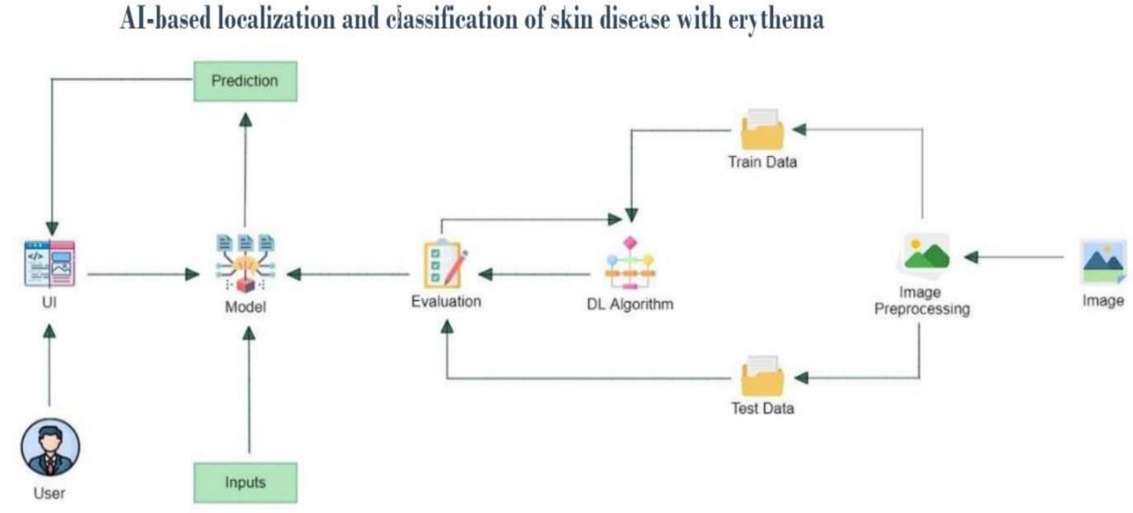
5.1 DATA FLOW DIAGRAM:

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.



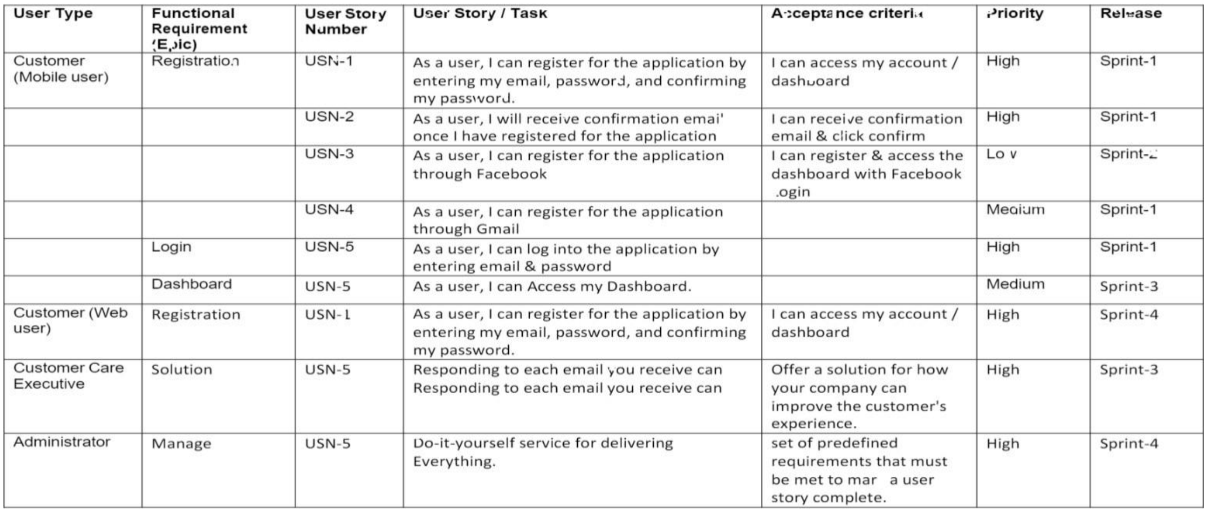
1) Medical dataset are collected from the patients and then it is pre-processing takes place.
2) There with the disease symptoms the machine learning/deep learning model with the use of prediction model it predicts the disease.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



User goes to the user interface ,then user gives the input to this model, using deep and machine learching the system evaluates the input and processes the image and it send back the test results to the user .

5.3 USER STORIES:

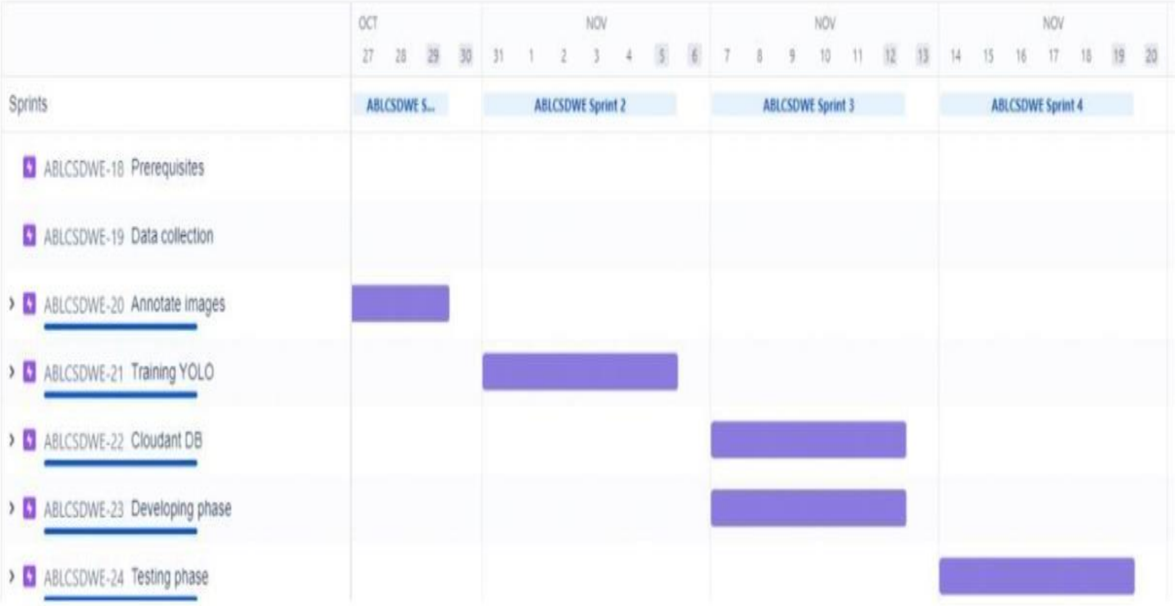


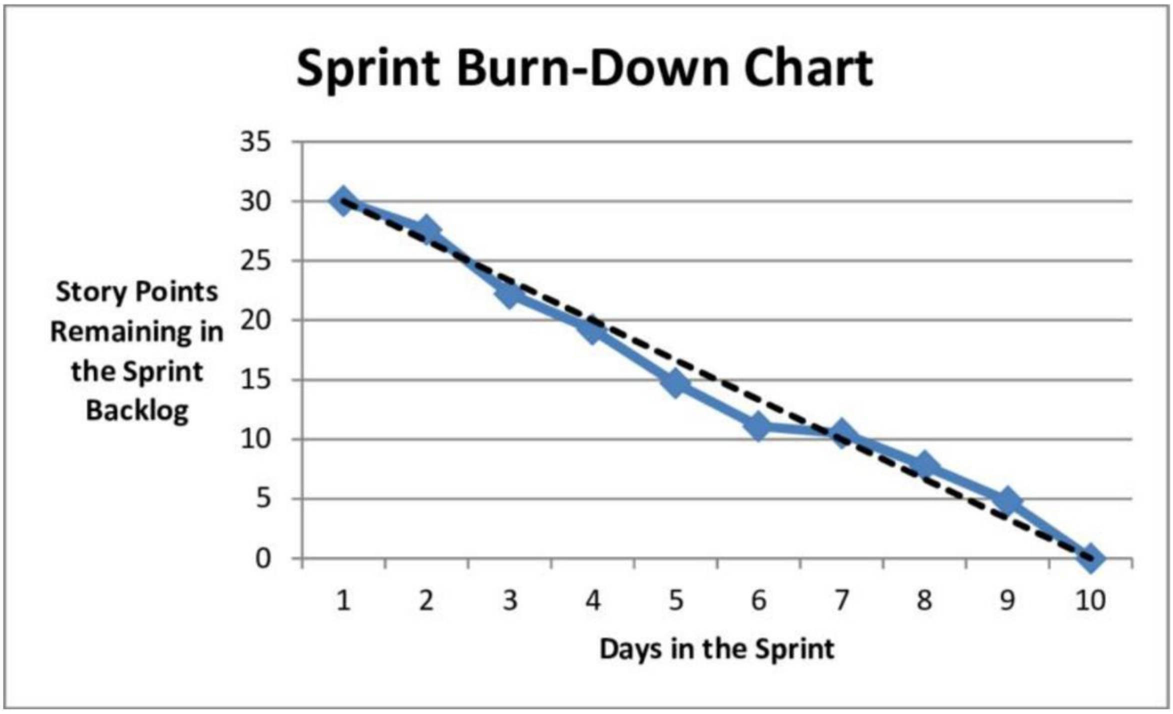
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can Access my Dashboard.		Medium	Sprint-3
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-4
Customer Care Executive	Solution	USN-5	Responding to each email you receive can Responding to each email you receive can	Offer a solution for how your company can improve the customer's experience.	High	Sprint-3
Administrator	Manage	USN-5	Do-it-yourself service for delivering Everything.	set of predefined requirements that must be met to mar a user story complete.	High	Sprint-4

CHAPTER 6

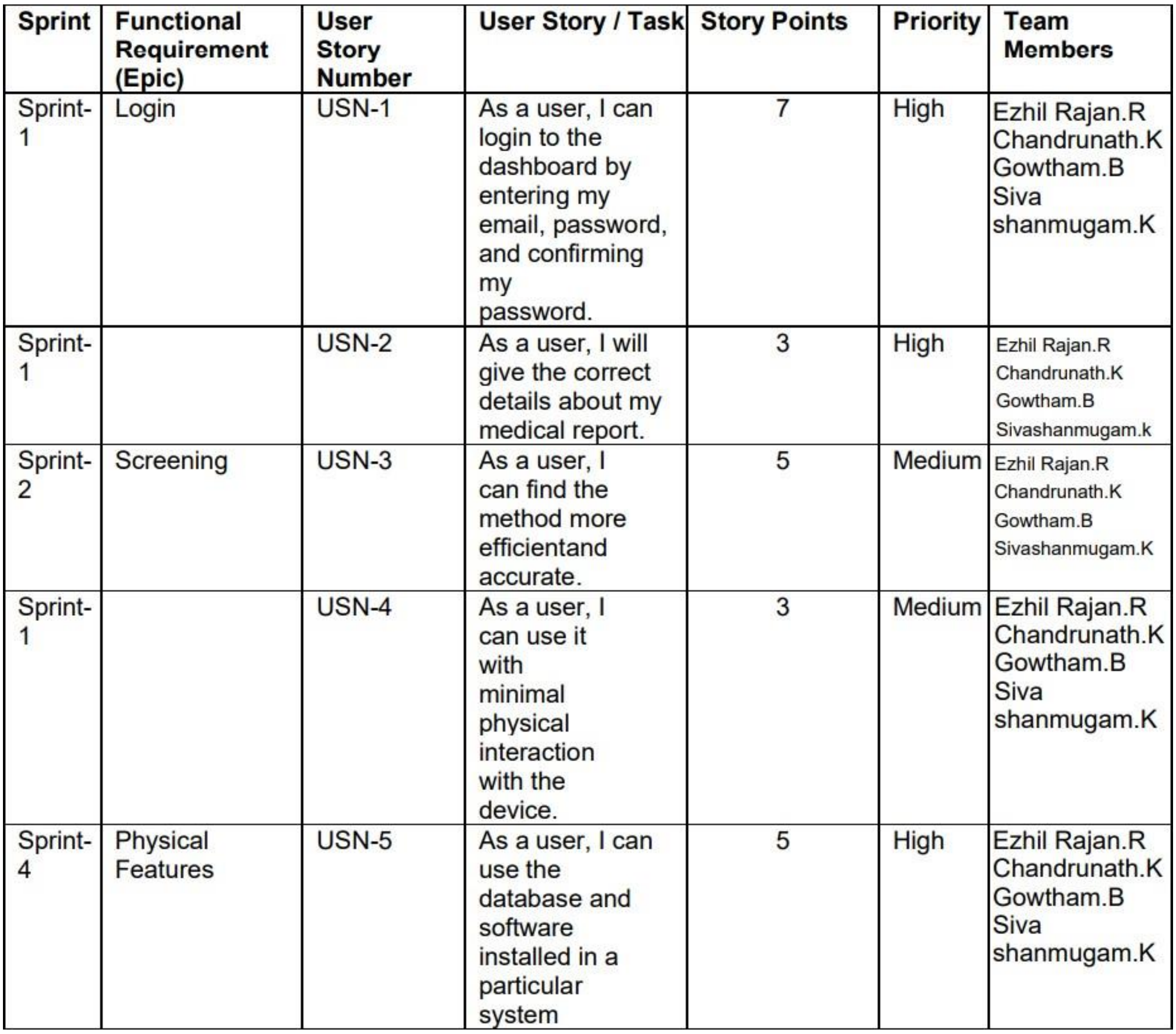
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

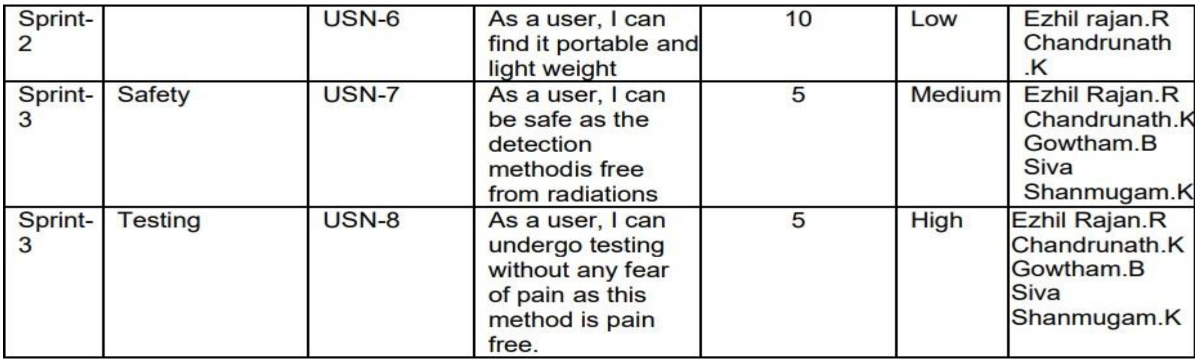




6.2 SPRINT DELIVERY SCHEDULE

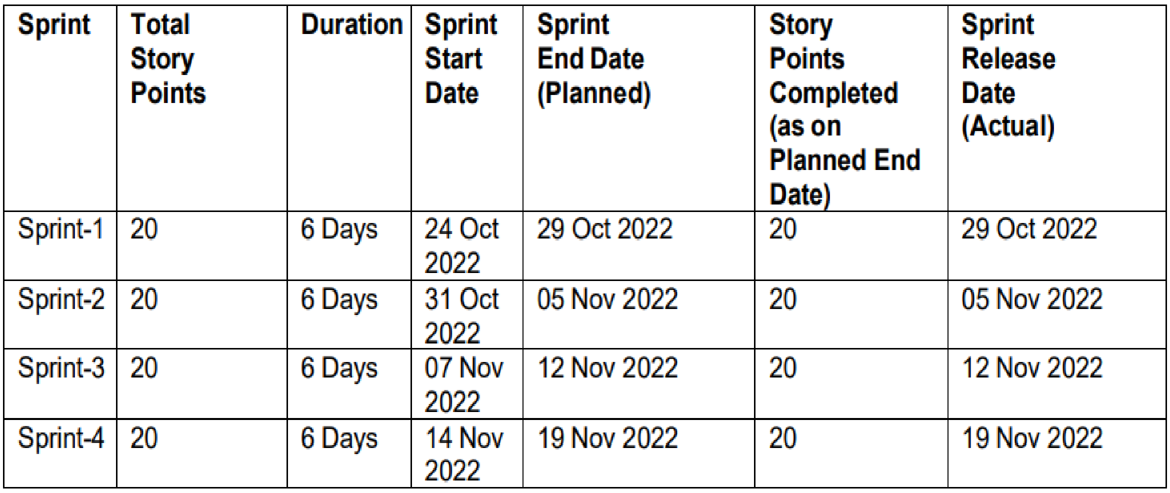


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a user, I can login to the dashboard by entering my email, password, and confirming my password.	7	High	Ezhil Rajan.R Chandrunath.K Gowtham.B Siva shanmugam.K
Sprint-1		USN-2	As a user, I will give the correct details about my medical report.	3	High	Ezhil Rajan.R Chandrunath.K Gowtham.B Sivashanmugam.k
Sprint-2	Screening	USN-3	As a user, I can find the method more efficientand accurate.	5	Medium	Ezhil Rajan.R Chandrunath.K Gowtham.B Sivashanmugam.K
Sprint-1		USN-4	As a user, I can use it with minimal physical interaction with the device.	3	Medium	Ezhil Rajan.R Chandrunath.K Gowtham.B Siva shanmugam.K
Sprint-4	Physical Features	USN-5	As a user, I can use the database and software installed in a particular system	5	High	Ezhil Rajan.R Chandrunath.K Gowtham.B Siva shanmugam.K



Sprint-2		USN-6	As a user, I can find it portable and light weight	10	Low	Ezhil rajan.R Chandrunath .K
Sprint-3	Safety	USN-7	As a user, I can be safe as the detection methodis free from radiations	5	Medium	Ezhil Rajan.R Chandrunath.K Gowtham.B Siva Shanmugam.K
Sprint-3	Testing	USN-8	As a user, I can undergo testing without any fear of pain as this method is pain free.	5	High	Ezhil Rajan.R Chandrunath.K Gowtham.B Siva Shanmugam.K

6.3 REPORTS FROM JIRA



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Project Tracker, Velocity & Burndown chart:

Velocity:

Imagine we have a 10-days sprint duration, and the velocity of the team is 20 (Points per sprint). Let’s calculate the team’s average velocity (AV)Per iteration unit (Story points per day).

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

CHAPTER 7

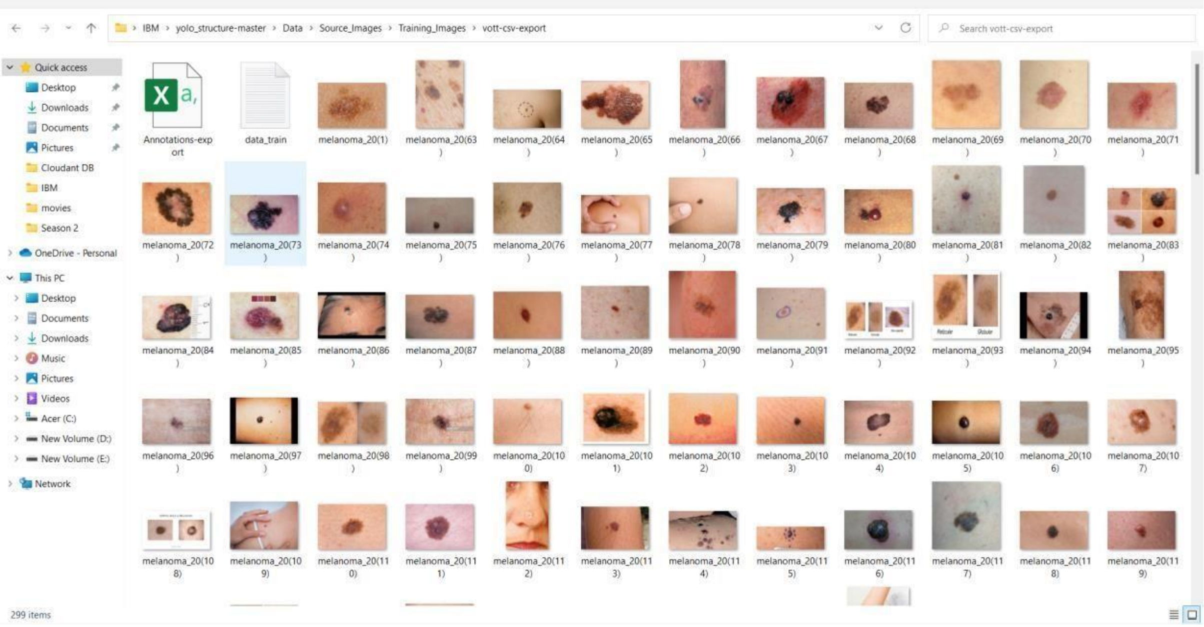
CODING AND SOLUTIONING

- **FEATURES 1 (TRAINING YOLO)**

Download and Convert Pre-Trained Weights

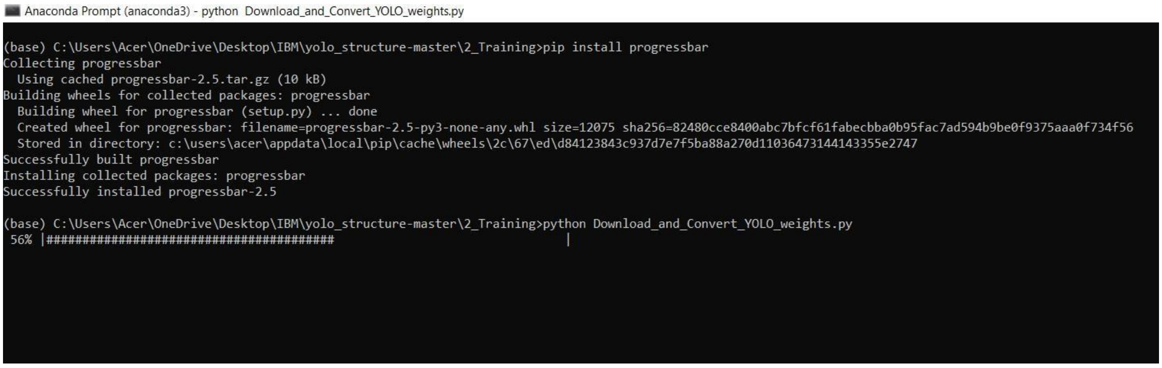
Step 1:

With the help of data_train.txt (Annotations) and the images we cantrain the model and before that we need to download and convert the weights.



Step 2:

The pre-trained weights are converted and being downloaded.



Anaconda Prompt (anaconda3) - python Download_and_Convert_YOLO_weights.py

```
(base) C:\Users\Acer\OneDrive\Desktop\IBM\yolo_structure-master\2_Training>pip install progressbar
Collecting progressbar
  Using cached progressbar-2.5.tar.gz (10 kB)
Building wheels for collected packages: progressbar
  Building wheel for progressbar (setup.py) ... done
  Created wheel for progressbar: filename=progressbar-2.5-py3-none-any.whl size=12075 sha256=82480cce8400abc7bfcf61fabecbba0b95fac7ad594b9be0f9375aaa0f734f56
  Stored in directory: c:\users\acer\appdata\local\pip\cache\wheels\2c\67\ed\d84123843c937d7e7f5ba88a270d11036473144143355e2747
Successfully built progressbar
Installing collected packages: progressbar
Successfully installed progressbar-2.5

(base) C:\Users\Acer\OneDrive\Desktop\IBM\yolo_structure-master\2_Training>python Download_and_Convert_YOLO_weights.py
 56% |########################################                                    |
```

TRAINING YOLOV3 DETECTOR

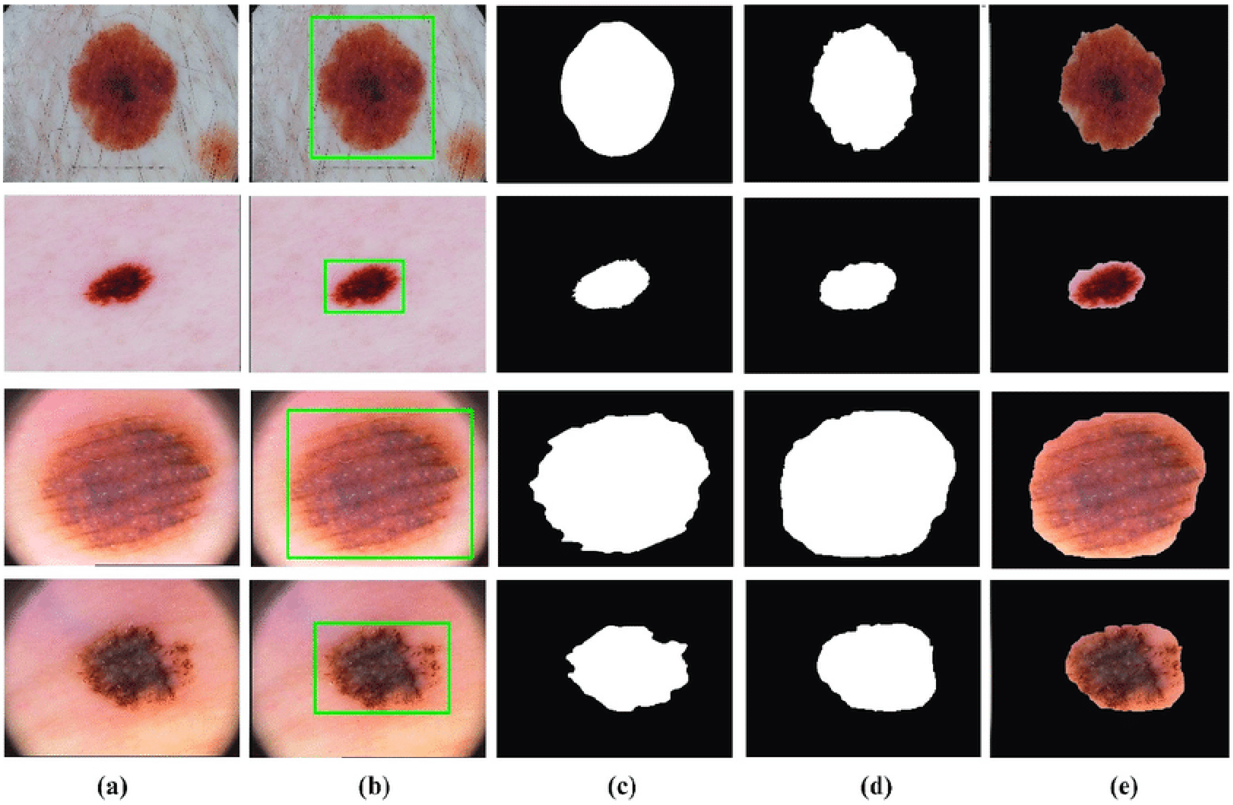


Image Processing

Histogram Manipulation

Import the required libraries.

```
import numpy as np
import matplotlib.pyplot as plt
importpandas as pdfrom skimage.ioimport imshow, imreadfrom
skimage.color import rgb2gray
fromskimage import img_as_ubyte, img_as_float
fromskimage.exposure importhistogram, cumulative_distribution
```

Convert the image to greyscale.

```
plt.figure(num=None, figsize=(8, 6), dpi=80) dark_image_grey =
img_as_ubyte(rgb2gray(image_dark))
imshow(dark_image_grey);
```

Extract the image’s value histogram.

```
freq, bins = histogram(dark_image_grey)plt.figure(num=None,figsize=(8,
6), dpi=100, facecolor='white')
freq, bins = histogram(dark_image_grey) plt.step(bins,
freq/freq.sum()) plt.xlabel('intensity value', fontsize = 12)
plt.ylabel('fraction of pixels', fontsize= 12);
```

Intensity Values of Image

It is very clear that this histogram does not resemble a normal distribution. You might be tempted to try and snap this distribution into a normal distribution. However there is a slightly more intuitiveway to handle this issue.

Remember that the theoretical Cumulative Distribution Function(CDF) for a normal distribution is a straight line. This being the case, it is better to snap the CDF of our image into a straight line.

Actual CDF of the Image

To do this, we can make use of the *interpolate* function in NumPy.

```
interpolation =np.interp(freq, target_freq, target_bins)
```

Use the interpolation to help us adjust the actual CDF.

```
dark_image_eq =
img_as_ubyte(interpolation[dark_image_grey].astype(int))
```

View the actual image.

```
imshow(dark_image_eq);
```

Create a function which will adjust the CDF of any image we feed it.

```
def histogram_adjuster(image):
      dark_image_grey    =    img_as_ubyte(rgb2gray(image))    freq,   bins   =
      cumulative_distribution(dark_image_grey)
target_bins = np.arange(255)
      target_freq    =    np.linspace(0,    1,    len(target_bins))    interpolation    =
np.interp(freq, target_freq, target_bins)
    dark_image_eq = img_as_ubyte(interpolation[dark_image_grey].astype(int))
    freq_adj, bins_adj = cumulative_distribution(dark_image_eq)


    fig, axes = plt.subplots(1, 2, figsize=(15,7));
    imshow(dark_image_grey, ax = axes[0]); imshow(dark_image_eq,
    ax = axes[1]);

    axes[0].axis('off')
    axes[1].axis('off')
    axes[0].set_title('Unadjusted Image', fontsize = 17)
    axes[1].set_title('Adjusted Image', fontsize = 17)

    fig, axes = plt.subplots(1, 1, figsize=(19,7)); plt.step(bins, freq, c='blue',
    label='Actual CDF') plt.step(bins_adj, freq_adj, c='purple', label='Adjusted
CDF')
    plt.plot(target_bins,
             target_freq, c='red', label='Target
             CDF',linestyle = '--')
    plt.legend(prop={'size': 14})
    plt.xlim(0, 255)
    plt.ylim(0, 1)
    plt.xlabel('Intensity values', fontsize = 15) plt.ylabel('Cumulativefraction
    of pixels', fontsize= 15);
```

Adjust the colored image directly.

```
defhistogram_adjuster_color(image):
    freq, bins = cumulative_distribution(image) target_bins =
np.arange(255)
    target_freq = np.linspace(0, 1, len(target_bins)) interpolation =
np.interp(freq, target_freq, target_bins)
    image_eq = img_as_ubyte(interpolation[image].astype(int)) freq_adj, bins_adj =
    cumulative_distribution(image_eq)


    fig, axes = plt.subplots(1, 2, figsize=(15,7)); imshow(image, ax =
    axes[0]);
    imshow(image_eq, ax = axes[1]);

    axes[0].axis('off')
    axes[1].axis('off')
    axes[0].set_title('Unadjusted Image', fontsize = 17)
    axes[1].set_title('Adjusted Image', fontsize = 17)
    fig, axes = plt.subplots(1, 1, figsize=(19,7)); plt.step(bins, freq, c='blue',
    label='Actual CDF') plt.step(bins_adj, freq_adj, c='purple', label='Adjusted
CDF')
    plt.plot(target_bins,
             target_freq, c='red', label='Target
             CDF',linestyle = '--')

    plt.legend(prop={'size': 15})
    plt.xlim(0, 255)
    plt.ylim(0, 1)
    plt.xlabel('Intensity values', fontsize = 17) plt.ylabel('Cumulative fraction of pixels',
    fontsize = 17);
```

- **FEATURE 2**

Training Yolo

To prepare for the training process, convert the YOLOv3 model to the Keras format. The YOLOv3 Detector can then be trained by Train_YOLO.py file.

Code:

```
     import  os
     import sys
     import
     argparse
     import
     warnings
     def get_parent_dir(n=1):
        """ returns the n-th parent directory of the
        currentworking directory """
        current_path = os.path.dirname(os.path.abspath(________file___))for k in range(n):
      current_path = os.path.dirname(current_path)return
        current_path
import tensorflow.compat.v1
as tfimport pickle
    from Train_Utils import (
    get_classes, get_anchors,
  create_model, create_tiny_model, data_generator,
  data_generator_wrapper,ChangeToOtherMachine,)

     keras_path = os.path.join(src_path,
        "keras_yolo3") Data_Folder =
        os.path.join(get_parent_dir(1), "Data")
        Image_Folder = os.path.join(Data_Folder, "Source_Images",
        "Training_Images")VoTT_Folder = os.path.join(Image_Folder, "vott-
        csv-export")
        YOLO_filename = os.path.join(VoTT_Folder, "data_train.txt")

        Model_Folder = os.path.join(Data_Folder,
        "Model_Weights") YOLO_classname =
        os.path.join(Model_Folder, "data_classes.txt")
        log_dir = Model_Folder

        anchors_path = os.path.join(keras_path, "model_data",
        "yolo_anchors.txt")weights_path = os.path.join(keras_path,
        "yolo.h5")
  type=str, default=log_dir,

 help="Folder to save training logs and trained weights to. Default is "

  + log_dir,

  )

parser.add_argument(
  "--anchors_path",
  type=str,
  default=anchors_path,

  help="Path to YOLO anchors. Default is " + anchors_path,

  )

parser.add_argument(
  "--weights_path",
  type=str,
  default=weights_path,

  help="Path to pre-trained YOLO weights. Default is " + weights_path,

  )
parser.add_argument("--val_split",
  type=float, default=0.1,
    help="Percentage of training set to be used for validation. Default is 10%.",

    )

parser.add_argument(
    "--is_tiny",
    default=False,


                    action="store_true",

                    help="Use the tiny Yolo version for better performance and less accuracy.

            Default is False.",

                )

              parser.add_argument("--random_seed",
              type=float, default=None,
              help="Random seed value to make script
              deterministic. Default is 'None', i.e.non-
              deterministic.",
              )

              parser.add_argument("--epochs",
              type=float, default=51,
                  help="Number of epochs for training last layers and number of
            epochs for fine-tuning layers. Default is 51.",
                )

               parser.add_a
                  rgument("-
                  -
                  warnings",
                  default=Fa
                  lse,
                  action="st
                  ore_true",
                   help="Display warning messages. Default is False.",

                )

               FLAGS = parser.parse_args()


               if not FLAGS.warnings:
```

```
        tf.logging.set_verbosity(tf.logging.E
        RROR)
        os.environ['TF_CPP_MIN_LOG_L
        EVEL']='3'
        warnings.filterwarnings("ignore")
    np.random.seed(FLAGS.random_seed)

    log_dir = FLAGS.log_dir

    class_names =
    get_classes(FLAGS.classes_file)
    num_classes = len(class_names)
    anchors =
    get_anchors(FLAGS.anchors_path)
    weights_path =
    FLAGS.weights_path
    input_shape = (416, 416) # multiple of 32, height,
    widthepoch1, epoch2 = FLAGS.epochs,
    FLAGS.epochs
    is_tiny_version = len(anchors) == 6 #
    default settingif FLAGS.is_tiny:
        model = create_tiny_model(

            input_shape, anchors, num_classes,
  freeze_body=2,weights_path=weights_path
        )

    else:

        model = create_model(

            input_shape, anchors, num_classes,
  freeze_body=2,weights_path=weights_path

) # make sure you know what you freeze



    log_dir_time = os.path.join(log_dir,
    "{}".format(int(time())))logging =
    TensorBoard(log_dir=log_dir_time)
    checkpoint =
        ModelCheckpoint(
        os.path.join(log_dir,
        "checkpoint.h5"),
        monitor="val_loss",
        save_weights_only=True,
        save_best_only=True,
        period=5,

    )

    reduce_lr = ReduceLROnPlateau(monitor="val_loss", factor=0.1,
patience=3,verbose=1)
    early_stopping = EarlyStopping(

        monitor="val_loss", min_delta=0, patience=10, verbose=1

    )


    val_split = FLAGS.val_split

    with
        open(FLAGS.annotation_fi
        le) as f:lines = f.readlines()


    # This step makes sure that the path names correspond to the local machine

    # This is important if annotation and training are done on different
machines (e.g.training on AWS)
    lines = ChangeToOtherMachine(lines,
    remote_machine="")np.random.shuffle(lines)
    num_val = int(len(lines) * val_split)


    num_train = len(lines) - num_val

    # Train with frozen layers first, to get a stable loss.

    # Adjust num epochs to your dataset. This step is enough to obtain a
    decent model.if True:
        model.compile(
            optimizer=Adam
            (lr=1e-3),loss={
                # use custom yolo_loss Lambda
                layer. "yolo_loss": lambda
```

```
                    y_true, y_pred: y_pred
                },

            )
        batch_size = 32print("Train on {} samples, val on {} samples, with batch size
{}.".format(num_train, num_val, batch_size
                )

            )

        history = model.fit_generator(
            data_generator_wrapper(
              lines[:num_train], batch_size, input_shape, anchors, num_classes

                ),

                steps_per_epoch=max(1, num_train //
                batch_size),
                validation_data=data_generator_wrappe
                r(
                    lines[num_train:], batch_size, input_shape, anchors, num_classes
),


         validation_steps=max(1, num_val //
           batch_size),


epochs=epoch1,
                initial_epoch=0,
               callbacks=[logging,
               checkpoint],
            )
            model.save_weights(os.path.join(log_dir,  "trained_weights_stage_1.h5"))


            step1_train_loss = history.history["loss"]


            file = open(os.path.join(log_dir_time,
            "step1_loss.npy"), "w") with
            open(os.path.join(log_dir_time, "step1_loss.npy"),
            "w") as f:
                for item in
                    step1_train_loss
                    :f.write("%s\n"
                    % item)
                file.close()


                step1_val_loss = np.array(history.history["val_loss"])
file = open(os.path.join(log_dir_time, "step1_val_loss.npy"), "w")
with open(os.path.join(log_dir_time, "step1_val_loss.npy"), "w") as f:
                  for item in
                    step1_val_loss
                    :
                    f.write("%s\n"
                    % item)
                file.close()


              # Unfreeze and continue training, to
              fine-tune.# Train longer if the result
              is unsatisfactory.
              if True:

                for i in range(len(model.layers)):
model.layers[i].trainable = True
model.compile(
                  optimizer=Adam(lr=1e-4), loss={"yolo_loss": lambda y_true, y_pred:
                  y_pred}

                ) # recompile to apply the change
                print("Unfreeze all layers.")


               batch_size = (

                  4 # note that more GPU memory is required after unfreezing the body

                )

                print(

                  "Train on {} samples, val on {} samples, with batch size
                    {}.".format(num_train, num_val, batch_size
                  )

                )
          history = model.fit_generator(
             data_generator_wrapper(
               lines[:num_train], batch_size, input_shape, anchors, num_classes

                  ),
                  steps_per_epoch=max(1, num_train //
                  batch_size),
                  validation_data=data_generator_wrappe
                  r(
                    lines[num_train:], batch_size, input_shape, anchors, num_classes

                  ),

                  validation_steps=max(1, num_val //
                  batch_size),epochs=epoch1 + epoch2,
                  initial_epoch=epoch1,
                  callbacks=[logging, checkpoint, reduce_lr, early_stopping],


                )
model.save_weights(os.path.join(log_dir, "trained_weights_final.h5"))
step2_train_loss = history.history["loss"]


                file = open(os.path.join(log_dir_time,
                "step2_loss.npy"), "w") with
                open(os.path.join(log_dir_time, "step2_loss.npy"),
                "w") as f:
                  for item in
                    step2_train_loss
                    :f.write("%s\n"
                    % item)
               file.close()


                step2_val_loss = np.array(history.history["val_loss"])


                file = open(os.path.join(log_dir_time,
                "step2_val_loss.npy"), "w") with
                open(os.path.join(log_dir_time, "step2_val_loss.npy"),
                "w") as f:
                  for item in
                    step2_val_loss
                    :
                    f.write("%s\n"
                    % item)
                file.close()
```

7.3 Database Schema

- Registration: When a new user registers, the backend connects to the IBM Cloudant and stores the user's credentials in the database.
- Login: To check if a user is already registered, the backend connects to Cloudant when they attempt to log in. They are an invalid user if they arenot already registered.
- IBM cloudant: Stores the data which is registered.
- app.py: Connects both Frontend and the cloudant for the verification of user credentials

CHAPTER 8

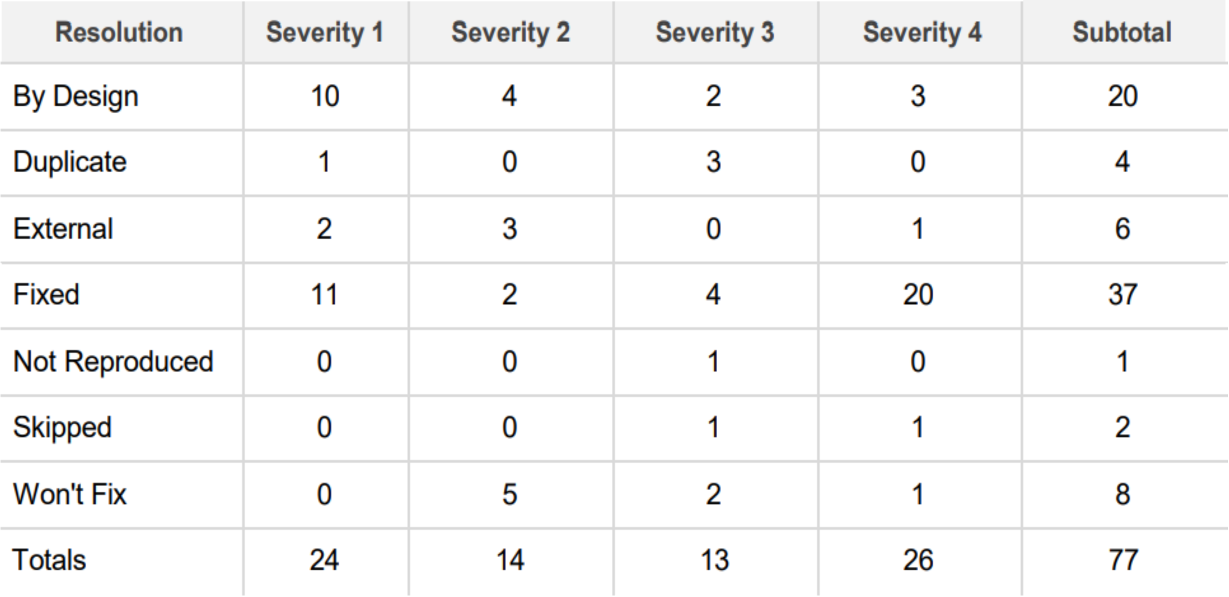
8. TESTING

8.1 USE CASE

The purpose of this document is to briefly explain the test coverage and open issues of the [Ai-Based Localization And Classification Of Skin Disease With Erythema] project at the time of the release to User Acceptance Testing (UAT).

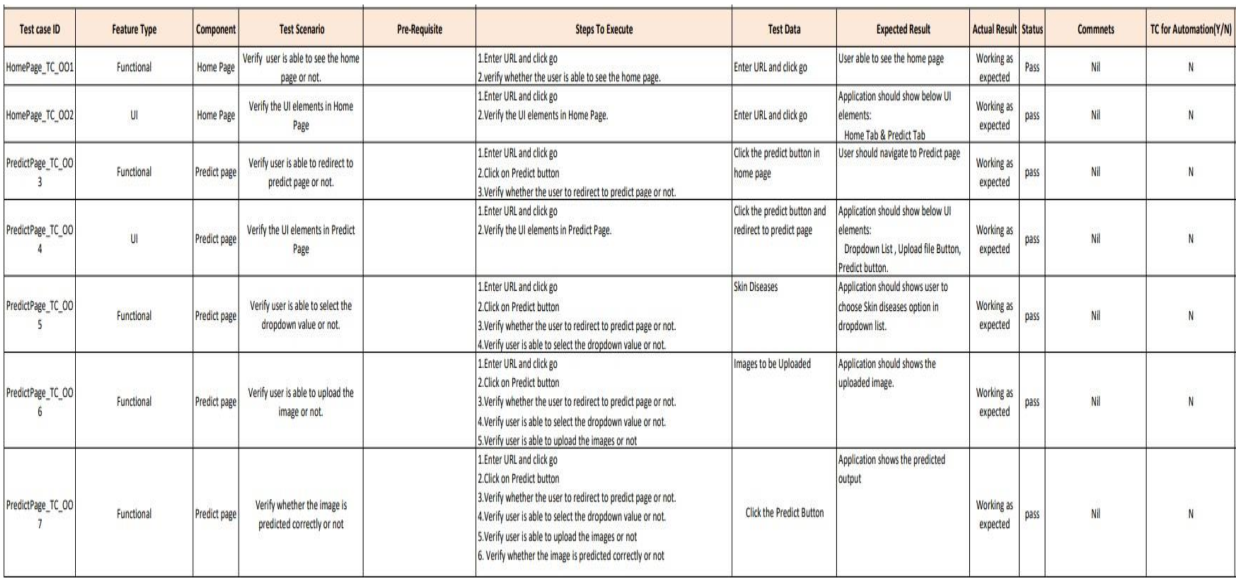
8.2 USER ACCEPTANCE TESTING

DEFECT ANALYSIS



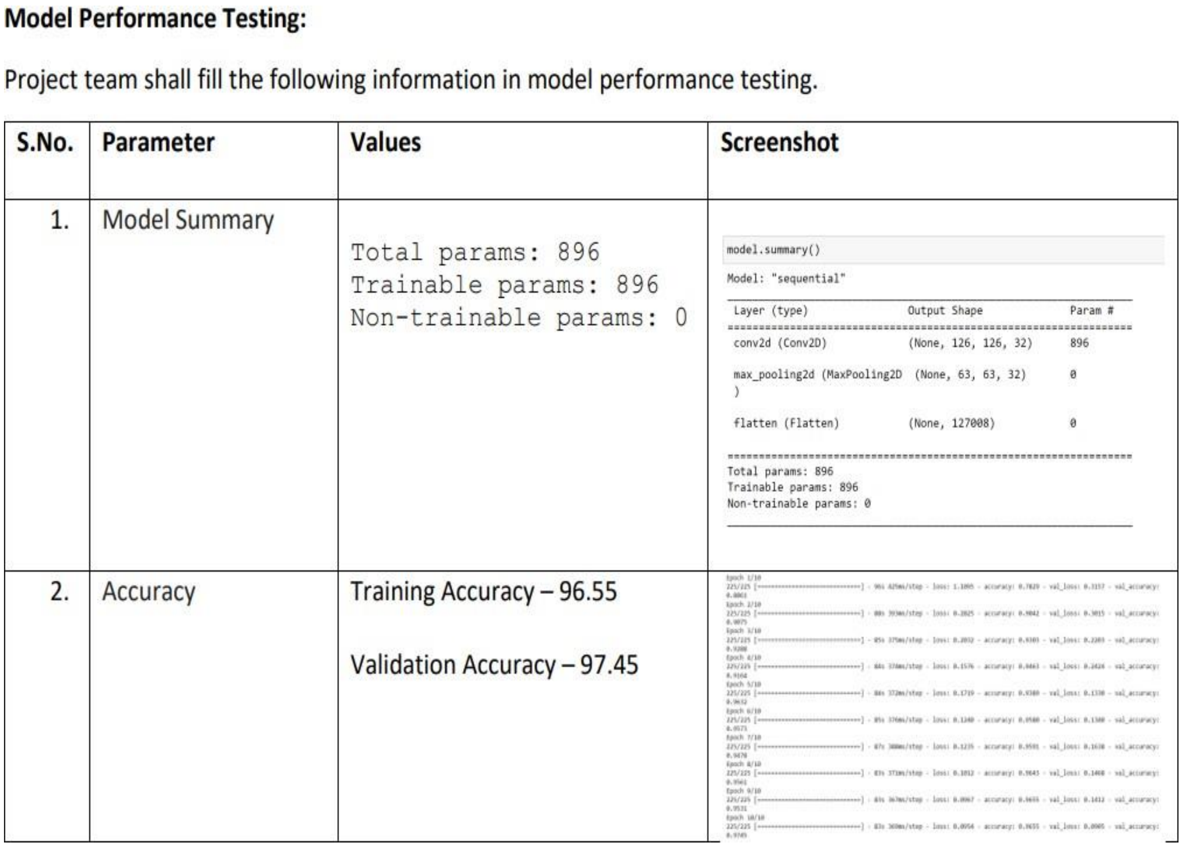
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

TEST CASE ANALYSIS



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the home page or not.		1.Enter URL and click go 2.verify whether the user is able to see the home page.	Enter URL and click go	User able to see the home page	Working as expected	Pass	Nil	N
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Home Page		1.Enter URL and click go 2.Verify the UI elements in Home Page.	Enter URL and click go	Application should show below UI elements: Home Tab & Predict Tab	Working as expected	pass	Nil	N
PredictPage_TC_OO3	Functional	Predict page	Verify user is able to redirect to predict page or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not.	Click the predict button in home page	User should navigate to Predict page	Working as expected	pass	Nil	N
PredictPage_TC_OO4	UI	Predict page	Verify the UI elements in Predict Page		1.Enter URL and click go 2.Verify the UI elements in Predict Page.	Click the predict button and redirect to predict page	Application should show below UI elements: Dropdown List , Upload file Button, Predict button.	Working as expected	pass	Nil	N
PredictPage_TC_OO5	Functional	Predict page	Verify user is able to select the dropdown value or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not.	Skin Diseases	Application should shows user to choose Skin diseases option in dropdown list.	Working as expected	pass	Nil	N
PredictPage_TC_OO6	Functional	Predict page	Verify user is able to upload the image or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5.Verify user is able to upload the images or not	Images to be Uploaded	Application should shows the uploaded image.	Working as expected	pass	Nil	N
PredictPage_TC_OO7	Functional	Predict page	Verify whether the image is predicted correctly or not		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5.Verify user is able to upload the images or not 6. Verify whether the image is predicted correctly or not	Click the Predict Button	Application shows the predicted output	Working as expected	pass	Nil	N

PERFORMANCE TESTING



Model Performance Testing:

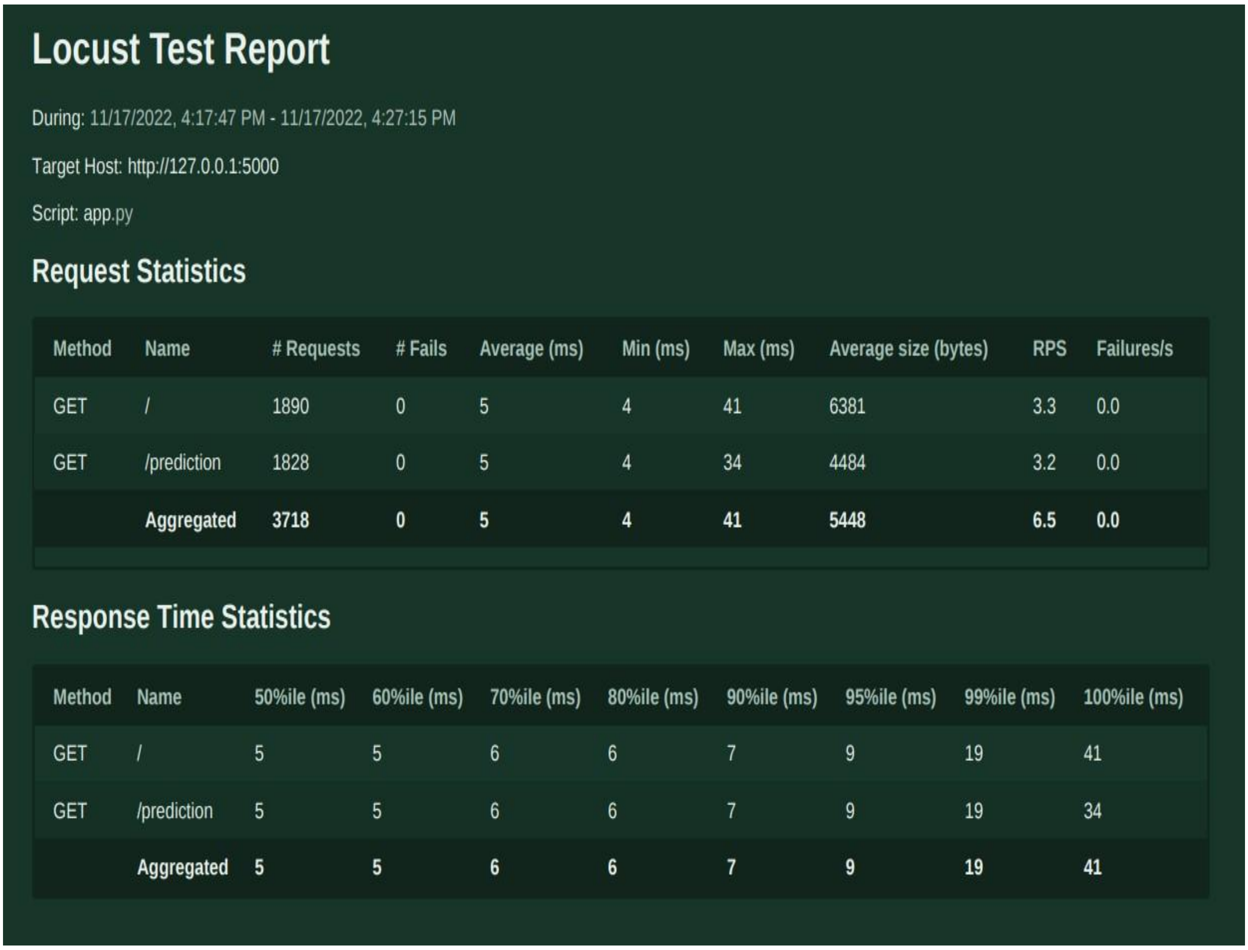
Project team shall fill the following information in model performance testing.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 896 Trainable params: 896 Non-trainable params: 0	model.summary()
2.	Accuracy	Training Accuracy – 96.55 Validation Accuracy – 97.45	

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

```
Epoch 1/10
225/225 [==============================] - 96s 425ms/step - loss: 1.1065 - accuracy: 0.7829 - val_loss: 0.3157 - val_accuracy:
0.8861
Epoch 2/10
225/225 [==============================] - 88s 393ms/step - loss: 0.2825 - accuracy: 0.9042 - val_loss: 0.3015 - val_accuracy:
0.9075
Epoch 3/10
225/225 [==============================] - 85s 375ms/step - loss: 0.2032 - accuracy: 0.9303 - val_loss: 0.2203 - val_accuracy:
0.9208
Epoch 4/10
225/225 [==============================] - 84s 374ms/step - loss: 0.1576 - accuracy: 0.9463 - val_loss: 0.2424 - val_accuracy:
0.9164
Epoch 5/10
225/225 [==============================] - 84s 372ms/step - loss: 0.1719 - accuracy: 0.9380 - val_loss: 0.1330 - val_accuracy:
0.9632
Epoch 6/10
225/225 [==============================] - 85s 376ms/step - loss: 0.1240 - accuracy: 0.9580 - val_loss: 0.1300 - val_accuracy:
0.9571
Epoch 7/10
225/225 [==============================] - 87s 388ms/step - loss: 0.1235 - accuracy: 0.9591 - val_loss: 0.1638 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 371ms/step - loss: 0.1012 - accuracy: 0.9643 - val_loss: 0.1468 - val_accuracy:
0.9561
Epoch 9/10
225/225 [==============================] - 83s 367ms/step - loss: 0.0967 - accuracy: 0.9655 - val_loss: 0.1412 - val_accuracy:
0.9531
Epoch 10/10
225/225 [==============================] - 83s 369ms/step - loss: 0.0954 - accuracy: 0.9655 - val_loss: 0.0905 - val_accuracy:
0.9745
```

LOCUST TEST REPORT



Locust Test Report

During: 11/17/2022, 4:17:47 PM - 11/17/2022, 4:27:15 PM

Target Host: http://127.0.0.1:5000

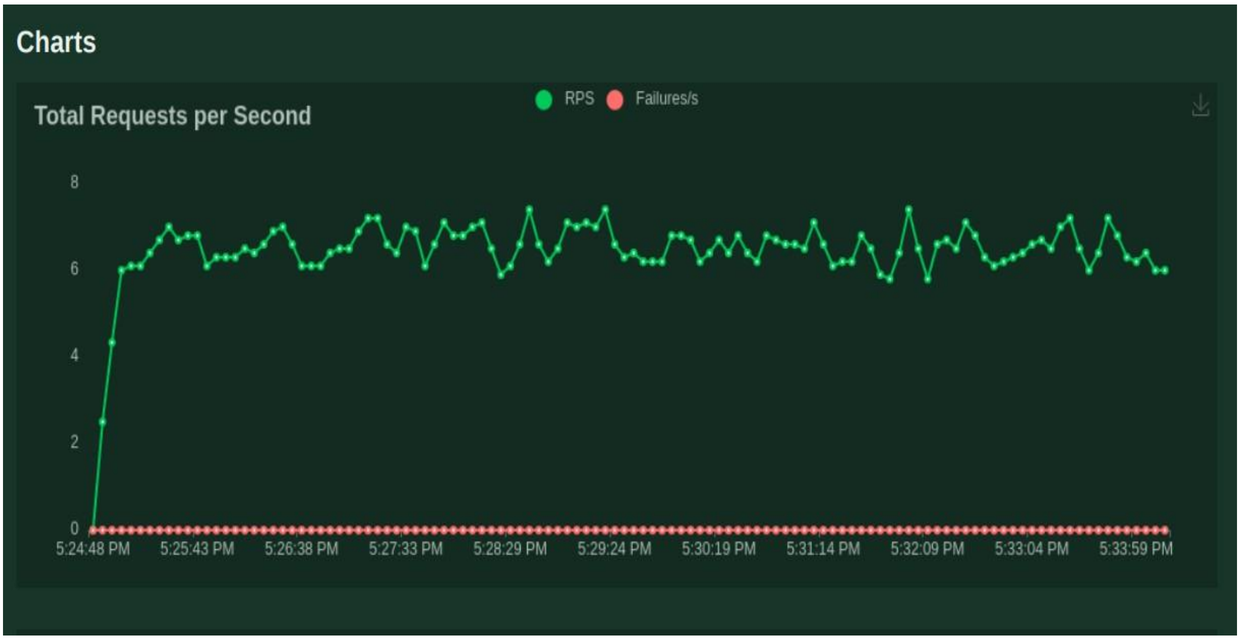
Script: app.py

Request Statistics

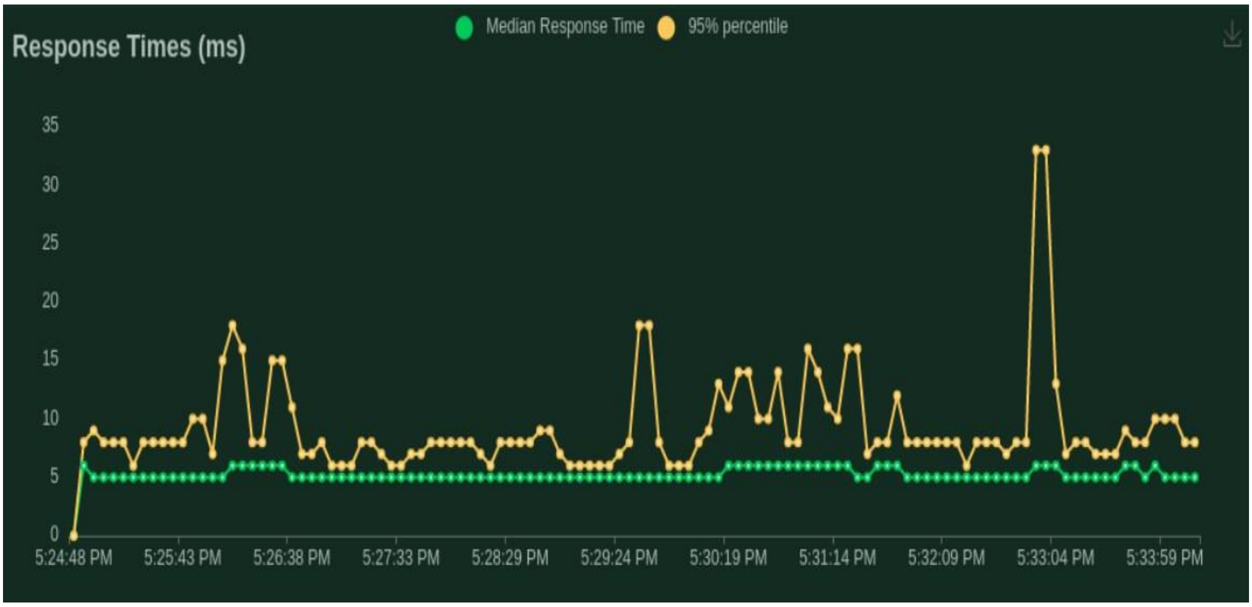
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1890	0	5	4	41	6381	3.3	0.0
GET	/prediction	1828	0	5	4	34	4484	3.2	0.0
	Aggregated	**3718**	**0**	**5**	**4**	**41**	**5448**	**6.5**	**0.0**

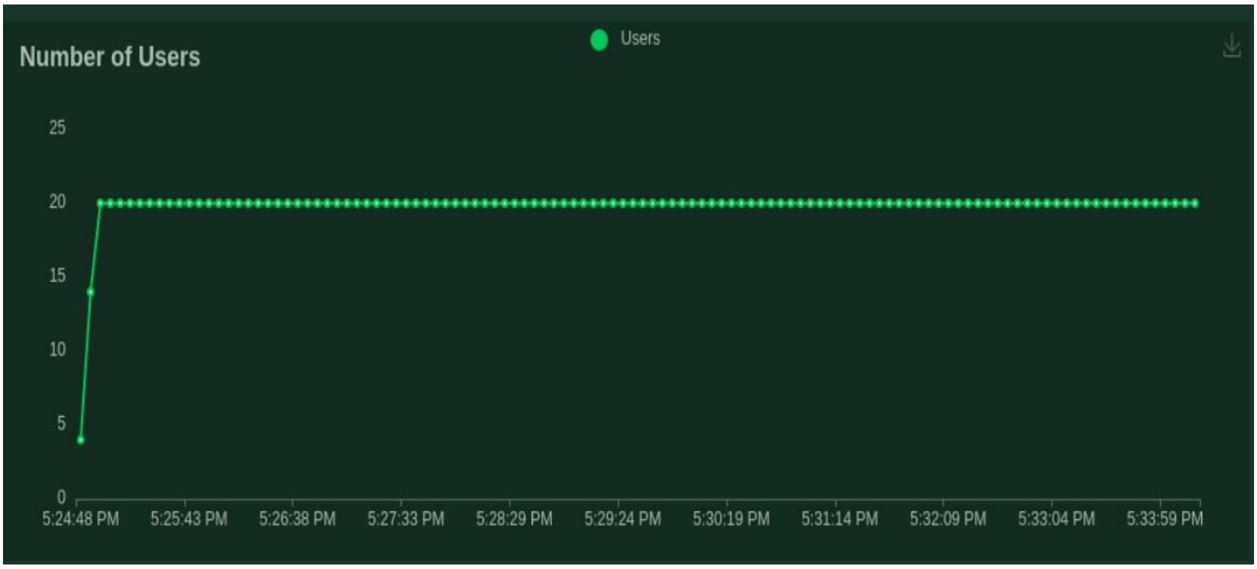
Response Time Statistics

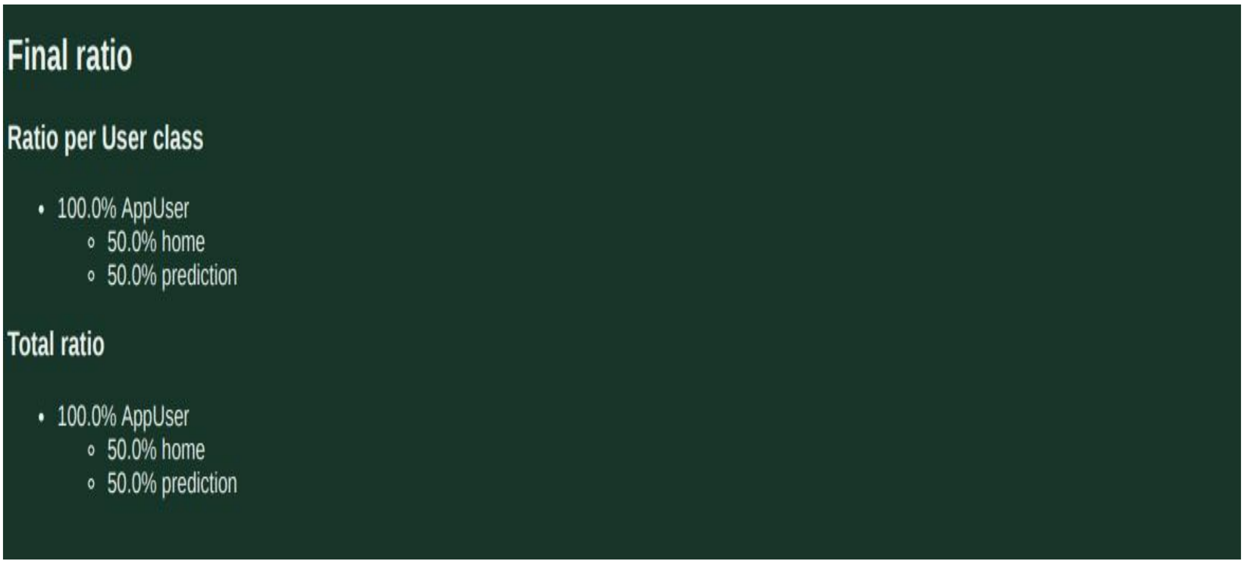
Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	5	5	6	6	7	9	19	41
GET	/prediction	5	5	6	6	7	9	19	34
	Aggregated	**5**	**5**	**6**	**6**	**7**	**9**	**19**	**41**



Charts







Final ratio

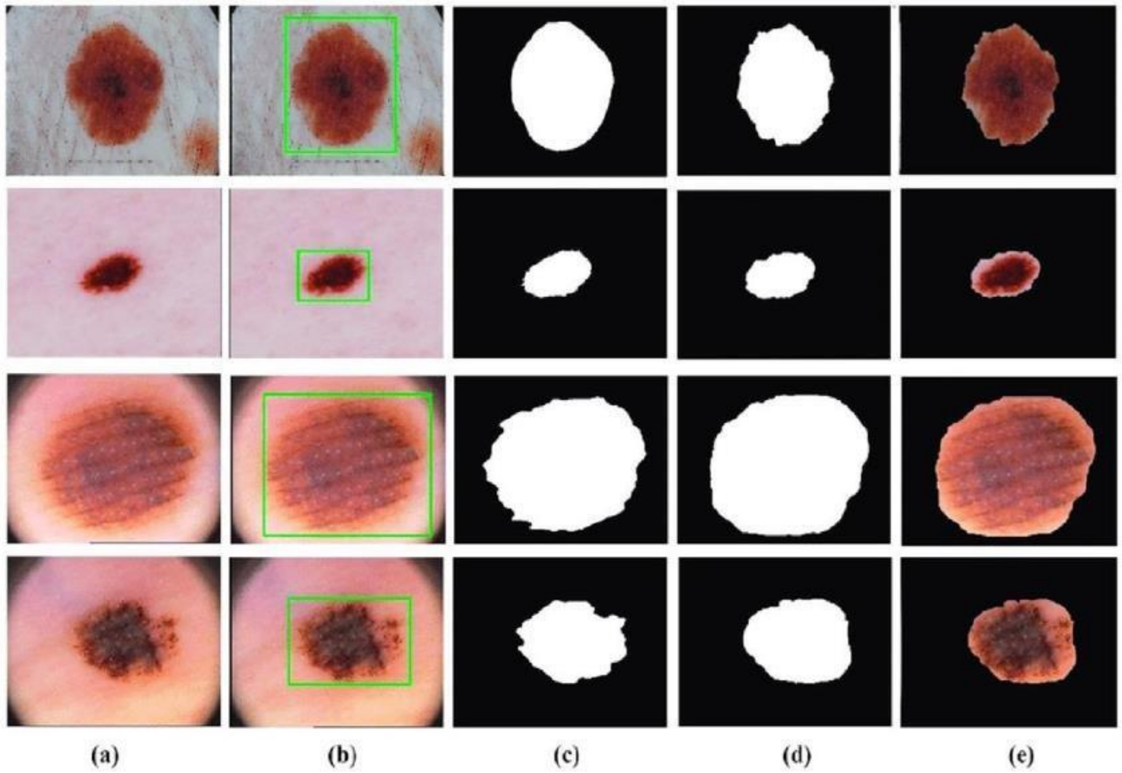
Ratio per User class

- 100.0% AppUser
  - 50.0% home
  - 50.0% prediction

Total ratio

- 100.0% AppUser
  - 50.0% home
  - 50.0% prediction

9 . RESULT



10 . ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

Instant Response, improves prediction of Skin Disease, no referral needed, Saves Money and Time, and Confidential Advice.

10.2 DISADVANTAGES

Network Connectivity and Accuracy

11 . Conclusion

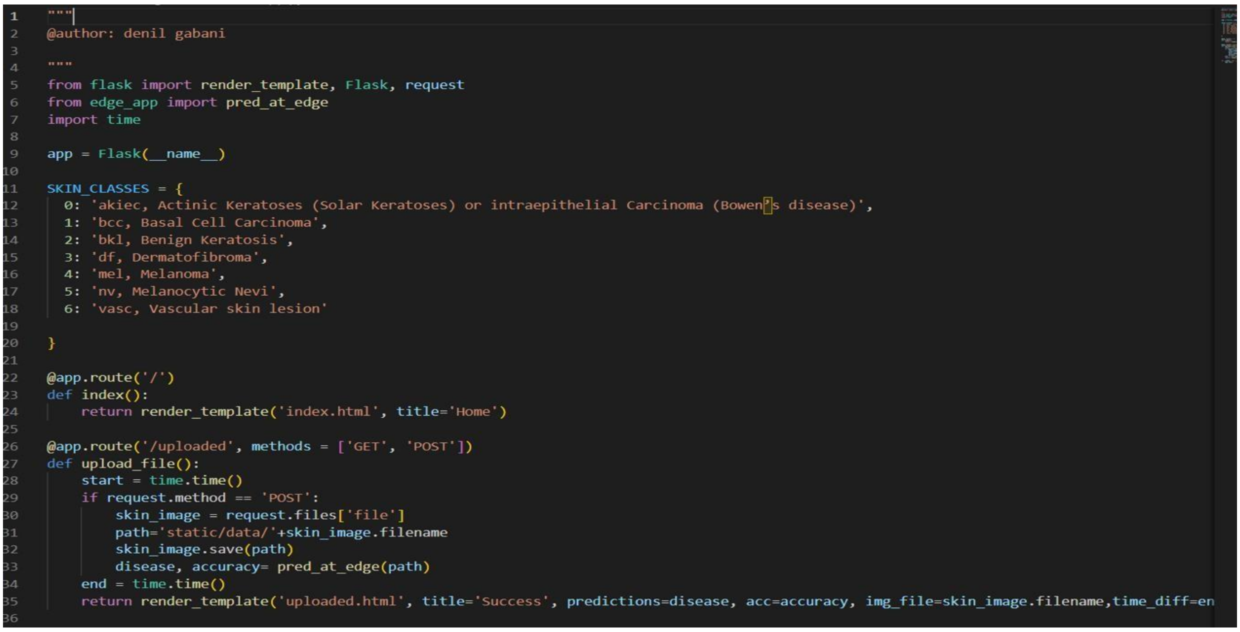
We have shown that even without a large dataset and high-quality images, it is possible to achieve sufficient accuracy rates. In addition, we have shown that current state-of-the-art CNN models can outperform models created by previous research, through proper data pre-processing, self-supervised learning, transfer learning, and special CNN architecture techniques. Furthermore, with accurate segmentation, we gain knowledge of the location of the disease, which is useful in the pre-processing of data used in classification, as it allows the CNN model to focus on the area of interest. Lastly, unlike previous studies, our method provides a solution to classify multiple diseases within a single image. With higher quality and a larger quantity of data, it will be viable to use state-of-the-art models to enable the use of CAD in the field of dermatology.

12 . FUTURE SCOPE

This implementation of the Structural Co-Occurrence matrices for feature extraction in the skin diseases classification and the pre-processing techniques are handled by using the Median filter, this filter helps toremove the salt and pepper noise in the image processing; thus, it enhances the quality of the images, and normally, the skin diseases are considered as the risk factor in all over the world. Our proposed

13. APPENDIX

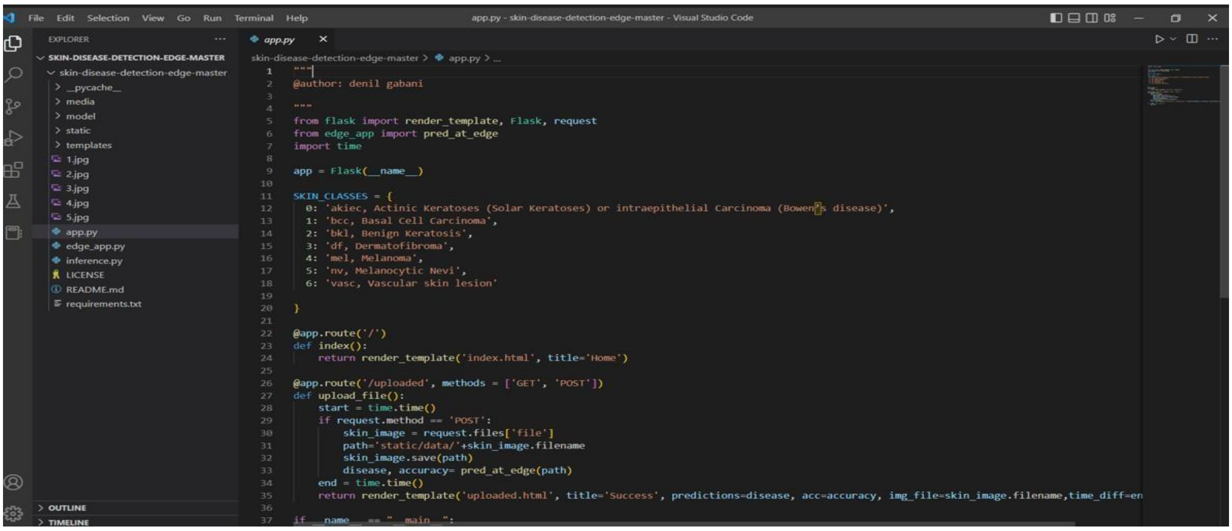
SOURCE CODE



```python
"""
@author: denil gabani

"""
from flask import render_template, Flask, request
from edge_app import pred_at_edge
import time

app = Flask(__name__)

SKIN_CLASSES = {
  0: 'akiec, Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen's disease)',
  1: 'bcc, Basal Cell Carcinoma',
  2: 'bkl, Benign Keratosis',
  3: 'df, Dermatofibroma',
  4: 'mel, Melanoma',
  5: 'nv, Melanocytic Nevi',
  6: 'vasc, Vascular skin lesion'

}

@app.route('/')
def index():
    return render_template('index.html', title='Home')

@app.route('/uploaded', methods = ['GET', 'POST'])
def upload_file():
    start = time.time()
    if request.method == 'POST':
        skin_image = request.files['file']
        path='static/data/'+skin_image.filename
        skin_image.save(path)
        disease, accuracy= pred_at_edge(path)
    end = time.time()
    return render_template('uploaded.html', title='Success', predictions=disease, acc=accuracy, img_file=skin_image.filename,time_diff=en
```

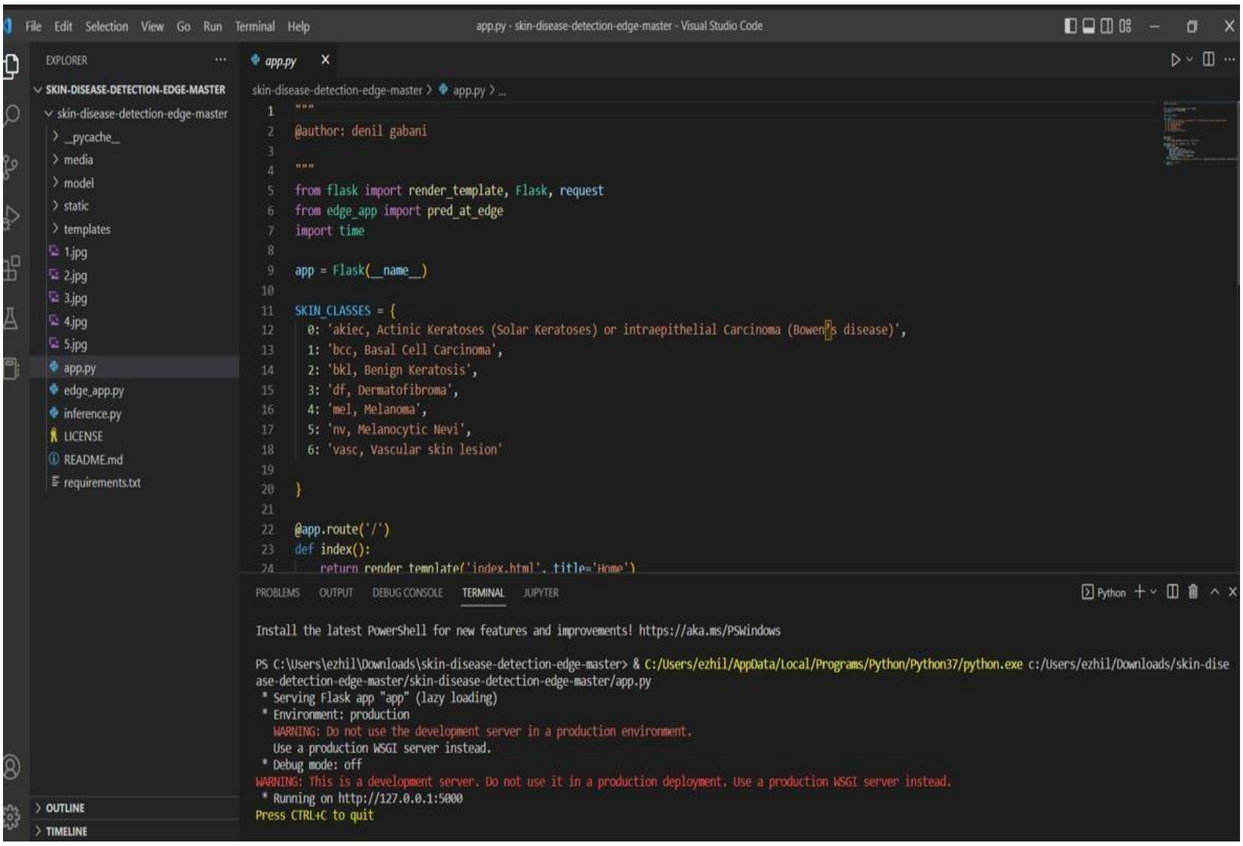
STEPS TO BE FOLLOWED

Step:1



```python
"""
@author: denil gabani

"""
from flask import render_template, Flask, request
from edge_app import pred_at_edge
import time

app = Flask(__name__)

SKIN_CLASSES = {
  0: 'akiec, Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen's disease)',
  1: 'bcc, Basal Cell Carcinoma',
  2: 'bkl, Benign Keratosis',
  3: 'df, Dermatofibroma',
  4: 'mel, Melanoma',
  5: 'nv, Melanocytic Nevi',
  6: 'vasc, Vascular skin lesion'

}

@app.route('/')
def index():
    return render_template('index.html', title='Home')

@app.route('/uploaded', methods = ['GET', 'POST'])
def upload_file():
    start = time.time()
    if request.method == 'POST':
        skin_image = request.files['file']
        path='static/data/'+skin_image.filename
        skin_image.save(path)
        disease, accuracy= pred_at_edge(path)
    end = time.time()
    return render_template('uploaded.html', title='Success', predictions=disease, acc=accuracy, img_file=skin_image.filename,time_diff=en

if __name__ == "__main__":
```

Step : 2

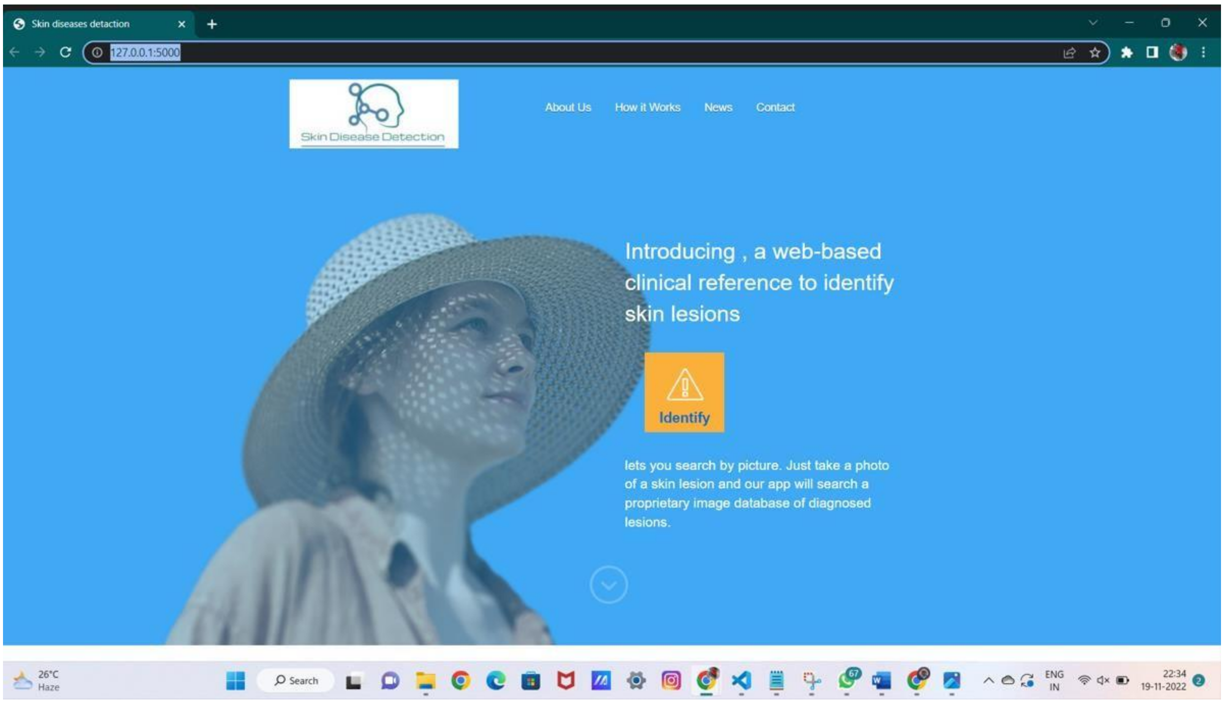


File Edit Selection View Go Run Terminal Help
app.py - skin-disease-detection-edge-master - Visual Studio Code
EXPLORER
SKIN-DISEASE-DETECTION-EDGE-MASTER
skin-disease-detection-edge-master
pycache
media
model
static
templates
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
app.py
edge_app.py
inference.py
LICENSE
README.md
requirements.txt
OUTLINE
TIMELINE
app.py
skin-disease-detection-edge-master > app.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
Python

```python
"""
@author: denil gabani

"""
from flask import render_template, Flask, request
from edge_app import pred_at_edge
import time

app = Flask(__name__)

SKIN_CLASSES = {
  0: 'akiec, Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen's disease)',
  1: 'bcc, Basal Cell Carcinoma',
  2: 'bkl, Benign Keratosis',
  3: 'df, Dermatofibroma',
  4: 'mel, Melanoma',
  5: 'nv, Melanocytic Nevi',
  6: 'vasc, Vascular skin lesion'

}

@app.route('/')
def index():
    return render_template('index.html', title='Home')
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ezhil\Downloads\skin-disease-detection-edge-master> & C:/Users/ezhil/AppData/Local/Programs/Python/Python37/python.exe c:/Users/ezhil/Downloads/skin-dise
ase-detection-edge-master/skin-disease-detection-edge-master/app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

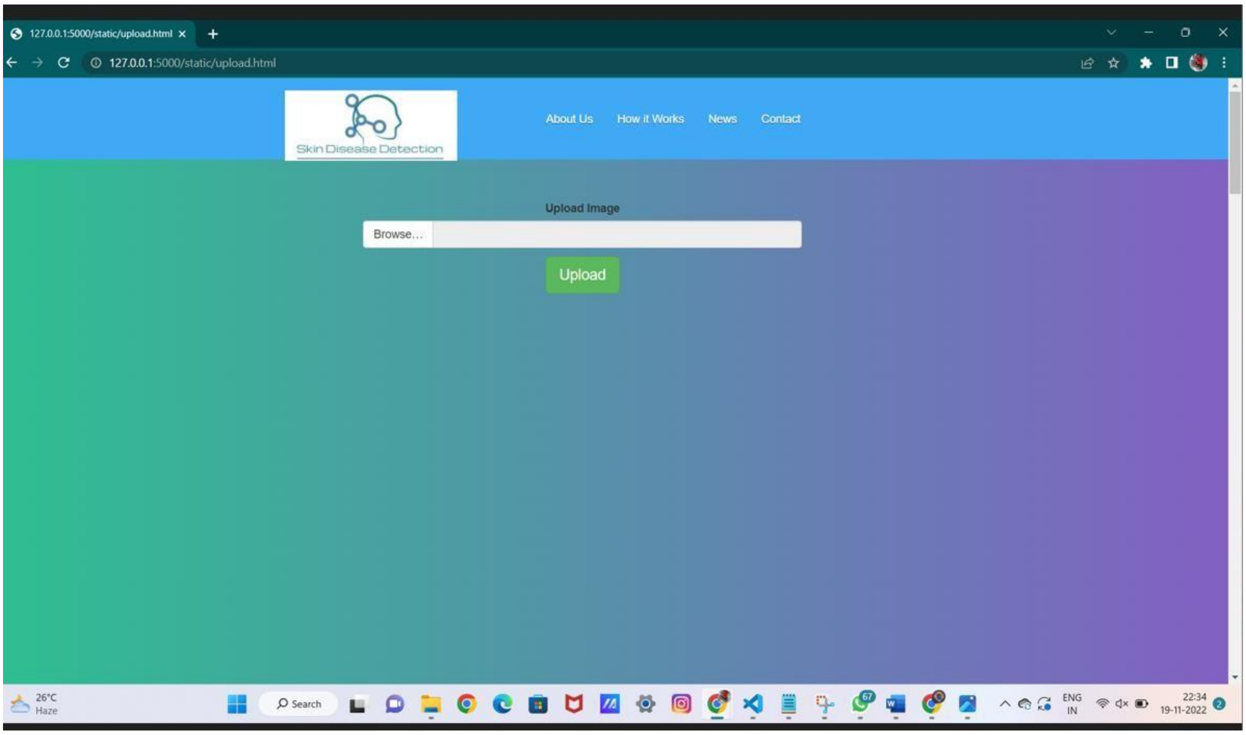
Step : 3



Introducing , a web-based clinical reference to identify skin lesions

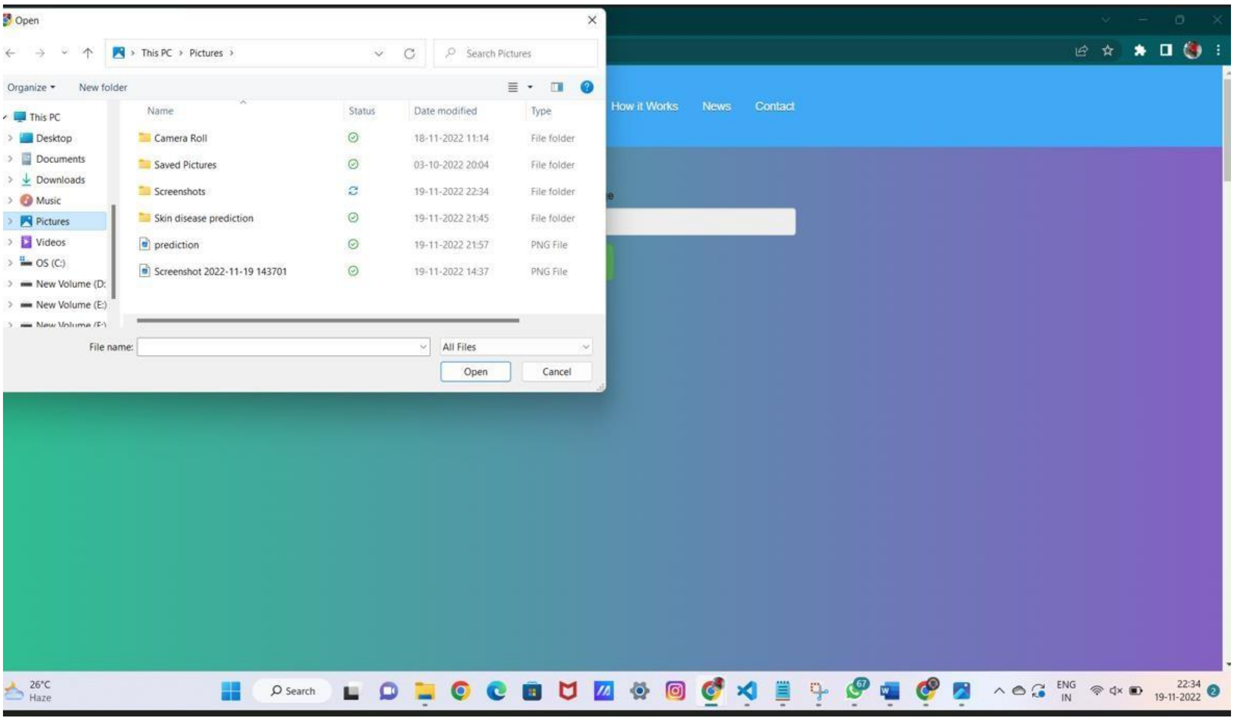
lets you search by picture. Just take a photo of a skin lesion and our app will search a proprietary image database of diagnosed lesions.

Step :4



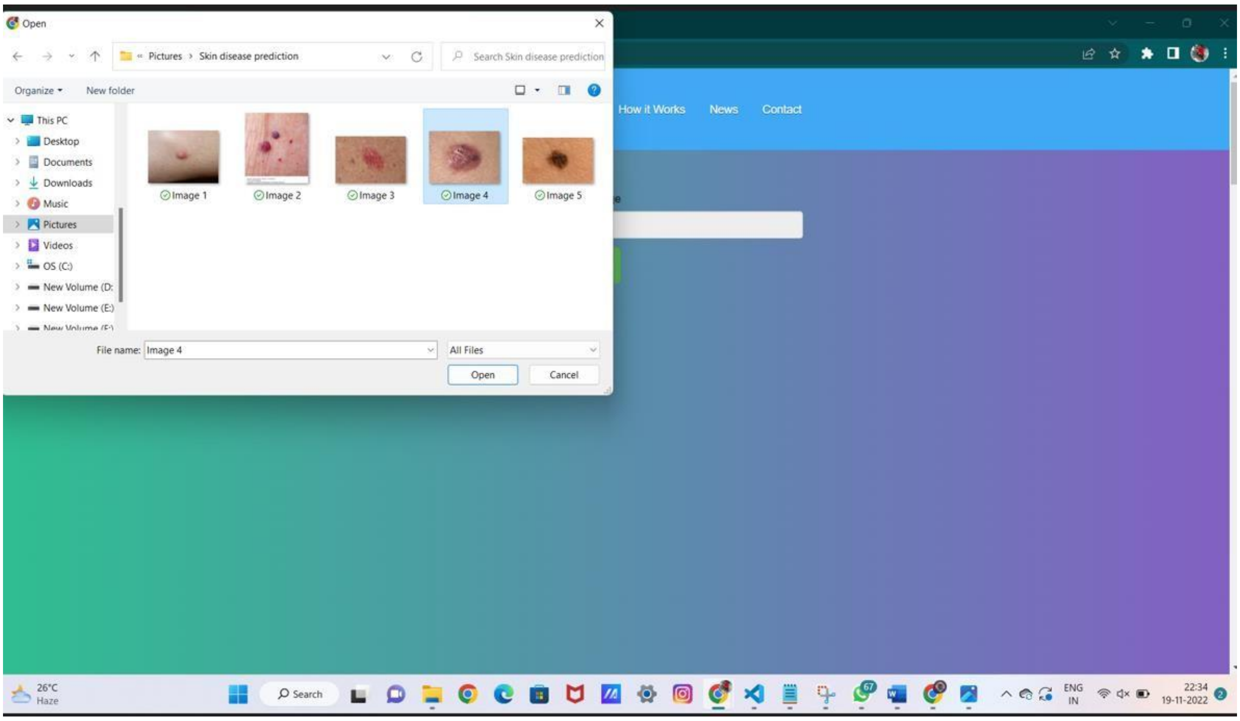
127.0.0.1:5000/static/upload.html

Step :5



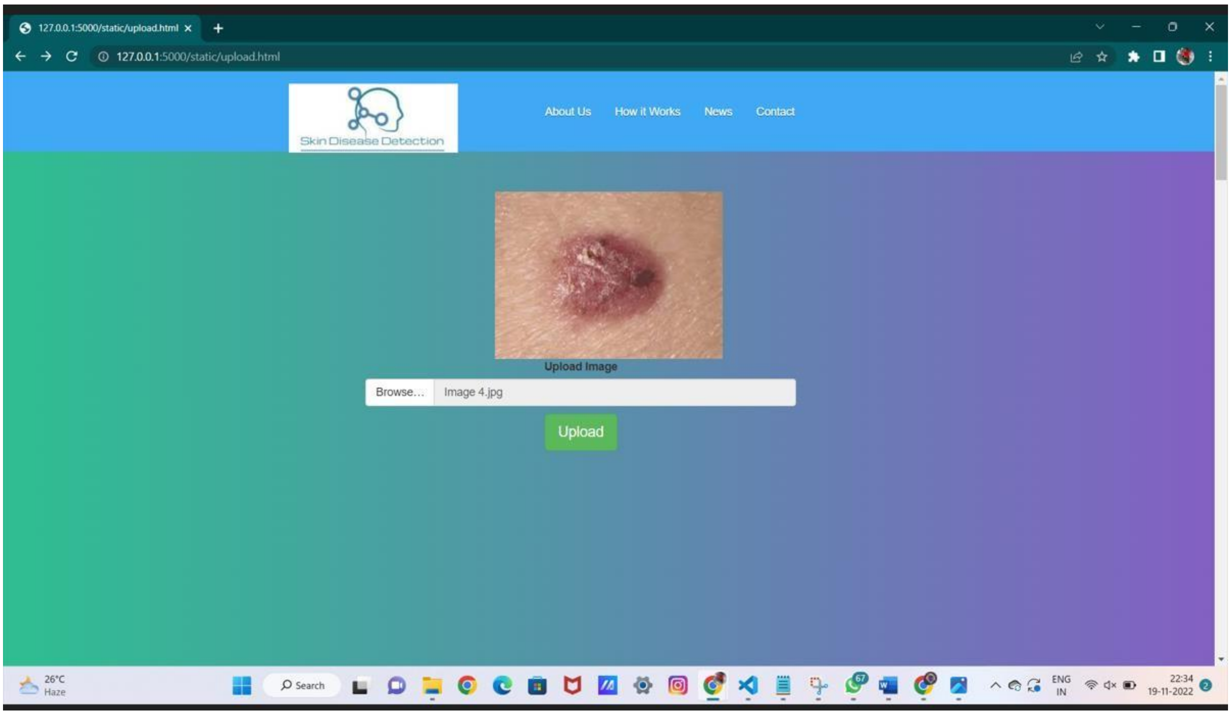
Name	Status	Date modified	Type
Camera Roll	✓	18-11-2022 11:14	File folder
Saved Pictures	✓	03-10-2022 20:04	File folder
Screenshots	⟳	19-11-2022 22:34	File folder
Skin disease prediction	✓	19-11-2022 21:45	File folder
prediction	✓	19-11-2022 21:57	PNG File
Screenshot 2022-11-19 143701	✓	19-11-2022 14:37	PNG File

Step : 6



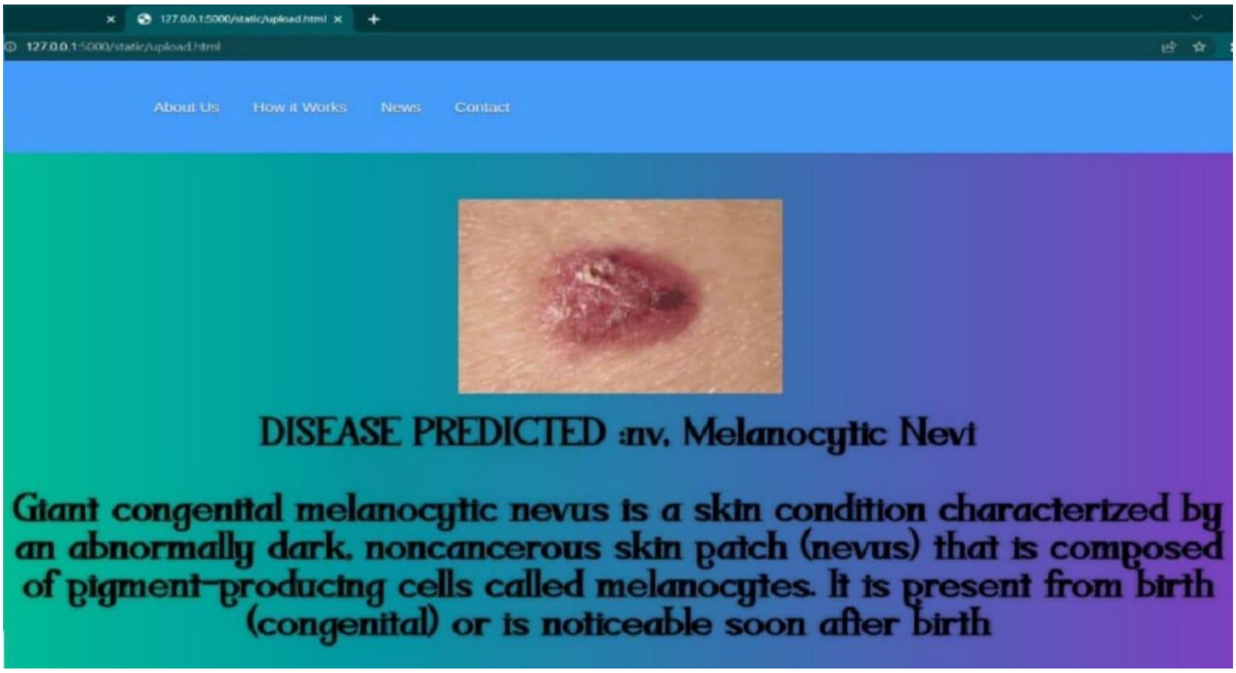
Open
Pictures › Skin disease prediction
Search Skin disease prediction
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
New Volume (D:
New Volume (E:)
Image 1
Image 2
Image 3
Image 4
Image 5
File name: Image 4
All Files
Open
Cancel
How it Works
News
Contact

Step : 7



127.0.0.1:5000/static/upload.html
Skin Disease Detection
About Us
How it Works
News
Contact
Upload Image
Browse...
Image 4.jpg
Upload

Step:8


127.0.0.1:5000/static/upload.html
127.0.0.1:5000/static/upload.html
About Us
How it Works
News
Contact

DISEASE PREDICTED :nv, Melanocytic Nevi

Giant congenital melanocytic nevus is a skin condition characterized by an abnormally dark, noncancerous skin patch (nevus) that is composed of pigment-producing cells called melanocytes. It is present from birth (congenital) or is noticeable soon after birth

Demo Link : https://youtu.be/NeXLIEg3-uQs

Github Link: https://github.com/IBM-EPBL/IBM-Project-23818-1659931638