AI-BASED LOCALIZATION AND CLASSIFICATION OF SKIN DISEASE WITH ERYTHEMA

NALAIYA THIRAN PROJECT

Team id: PNT2022TMID43452

Submitted by:

Vikram C.M.
Akilesh J.S.
Vikas Shrinivas Naik
Puvarasan.V.
Pagadala Davidlivingston

AI-based localization and classification of skin disease with erythema.

1. INTRODUCTION

AI-based localization and classification of skin disease with erythema in this project, We will be identifying skin diseases like Psoriasis, Melanoma, Rosacea and Eczema in early stages without the intervention of a doctor..

1.1 PROJECT OVERVIEW:

Skin diseases are more common than other diseases. Skin diseases may be caused by fungal infection, bacteria, allergy, or viruses, etc. A skin disease may change the texture or color of the skin. In general, skin diseases are chronic, infectious and sometimes may develop into skin cancer. Therefore, skin diseases must be diagnosed early to reduce their development and spread. The diagnosis and treatment of a skin disease takes a longer time and causes financial and physical cost to the patient. In general, most of the common people do not know the type and stage of a skin disease. Some of the skin diseases show symptoms several months later, causing the disease to develop and grow further. This is due to the lack of medical knowledge in the public. Sometimes, a dermatologist (skin specialist doctor) may also find it difficult to diagnose the skin disease and may require expensive laboratory tests to correctly identify the type and stage of the skin disease. We propose an image processing-based approach to diagnose skin diseases. This method takes the digital image or video frames of disease effect skin area then uses image analysis to identify the type of disease. Our proposed approach is simple, fast and does not require expensive equipment other than a camera and a computer.

1.2 PURPOSE:

The purpose of the project is design and implementation of deep learning model deployed for detection of image processing based skin disease

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM:

[1] This Proposed system describes , 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 non-invasive screening tests only with the naked eye. This may result in avoidable diagnostic inaccuracies as a result of human error, as the detection of the disease can be easily overlooked. Therefore, it would be beneficial to exploit the strengths of CAD using artificial intelligence techniques, in order to improve the accuracy of dermatology diagnosis. 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).
Advantages: Clustering algorithms generally have the advantage of being flexible, easy to implement, with the ability to generalize features that have a similar statistical variance.
Disadvantages: 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.

[2] This Proposed system describes an improved hybrid feature selection method, named improved F-score and Sequential Forward Search (IFSFS), which is a combination of filter and wrapper methods to select the optimal feature subset from the original feature set. The news IFSFS method improved the original Fscore from measuring the discrimination of two sets of real numbers to measuring the discrimination between more than two sets of real numbers. The improved Fscore and Sequential Forward Search (SFS) are combined to find the optimal feature subset in the process of feature selection. The best parameters of kernel function of SVM are found by grid search technique. Xie and Wang then conducted experiments on different training-test partitions of the Erythemato – Squamous diseases dataset taken from UCI (University of California Irvine) machine learning database. Their experimental results show that the proposed SVM-based model with IFSFS achieves 98.61% classification accuracy and contains 21 features. The authors conclude that their method is very promising compared to the previously reported results.

Advantages: Experimental results showed that their proposed hybrid methods construct efficient diagnosis classifiers with high average accuracy when compared with traditional algorithms.
Disadvantages: Although the CART classifier achieved a better accuracy than some methods such as ANN and C4.5, it is strongly recommended that ensemble methods should be used to classify differential diagnosis of ESD. Further experimental investigations are needed to conduct a prediction model of ESD, using the CART classifier in combination with other methods.

[3] Initially, skin images are filtered to remove unwanted hairs and noise and then the segmentation process is carried out to extract lesion areas. For segmentation, a region growing method is applied by automatic initialization of seed points. The segmentation performance is measured with different well known measures and the results are appreciable. Subsequently, the extracted lesion areas are represented by color and texture features. SVM and k-NN classifiers are used along with their fusion for the classification using the extracted features. The performance of the system is tested on our own dataset of 726 samples from 141 images consisting of 5 different classes of diseases. The results are very promising with 46.71% and 34% of F-measures using SVM and k-NN classifiers respectively and with 61% of F-measures for fusion of SVM and k-NN.
Advantages: We consider color and texture features in this paper. One more point that needs to be observed is the learning algorithms. Although many classifiers are used, fusion of decisions from multiple classifiers is recently gaining importance due to the fact that improvement in classification can be achieved.
Disadvantages: Based on the performance of the proposed model we have observed the performance of the system has decreased quite considerably for some classes and because of which the overall performance is also affected. This is due to a collection of dataset from internet resources.

[4] A skin lesion segmentation (SLS) method based on the separable-Unet with stochastic weight averaging is proposed in this work. Specifically, the proposed Separable-Unet framework takes advantage of the separable convolution block and U-Net architectures, which can extremely capture the context feature channel correlation and higher semantic feature information to enhance the pixel-level discriminative representation capability of fully convolutional networks (FCN). Further, considering that the over-fitting is a local optimum (or sub-optimum) problem, a scheme based on stochastic weight averaging is introduced, which can obtain a much broader optimum and better generalization.
Advantages: The proposed approach is compared with other state-of-the-art methods, and the results demonstrate that the proposed approach outperforms them for SLS in both melanoma and non-melanoma cases.

Disadvantages: Moreover, the considerably decreased computation time suggests that the proposed approach has potential for practical computer-aided diagnosis systems, besides providing a segmentation for the specific analysis with improved segmentation performance.

[5] In this paper, we have presented a survey of more than 100 papers and comparative analysis of state of the art techniques, models and methodologies. Malignant melanoma is one of the most threatening and deadliest cancers. Since the last few decades, researchers are putting extra attention and effort in accurate diagnosis of melanoma. The main challenges of dermoscopic skin lesion images are: low contrasts, multiple lesions, irregular and fuzzy borders, blood vessels, regression, hairs, bubbles, variegated coloring and other kinds of distortions. The lack of a large training dataset makes these problems even more challenging. Due to recent advancement in the paradigm of deep learning, and especially the outstanding performance in medical imaging, it has become important to review the deep learning algorithms performance in skin lesion segmentation.
Advantages: It is expected to improve results by utilizing the capabilities of deep learning frameworks with other pre and post processing techniques so reliable and accurate diagnostic systems can be built.
Disadvantages: It also leaves the programmers clueless when they try to understand why certain aspects fail. Generally, deep learning algorithms sift through millions of data points to find patterns and correlations that often go unnoticed by human experts.

[6] Dermoscopy is one of the major imaging modalities used in the diagnosis of melanoma and other pigmented skin lesions. Due to the difficulty and subjectivity of human interpretation, computerized analysis of dermoscopy images has become an important research area. One of the most important steps in dermoscopy image analysis is the automated detection of lesion borders. In this article, we present a systematic overview of the recent border detection methods in the literature paying particular attention to computational issues and evaluation aspects.
Advantages: Border determination by dermatologists appears to depend upon higher-level knowledge, therefore it is likely that the incorporation of domain knowledge in automated methods will enable them to perform better, especially in sets of images with a variety of diagnoses.
Disadvantages: Common problems with the existing approaches include the acquisition, size, and diagnostic distribution of the test image set, the evaluation of the results, and the inadequate description of the employed methods.

[7] This proposed system describes skin disease recognition by using a neural network which is based on image analysis. In general, these diseases have similarities in pattern of infection and symptoms such as redness and rash. Diagnosis and recognition of skin disease take a very long term process because it requires a patient's history, physical examination and proper laboratory diagnostic tests. Computer algorithms which contain a few steps that involve image processing, image feature extraction and classification of data have been implemented with the help of classifiers such as artificial neural networks (ANN).

Advantages: The ANN can learn patterns of symptoms of particular diseases and provides faster diagnosis and recognition than a human physician. Thus, the patients can do the treatment for the skin disease faced immediately based on the symptoms detected.

Disadvantages: Not only do artificial neural networks, but also the statistical models can be trained with only numeric data, so it makes it very difficult for ANN to understand the problem statement.

[8] This proposed system examines the use of AI methods for detecting erythema against the most clinically relevant skin conditions that may be “confusers”. Early detection of erythema , and diagnosis and treatment of Lyme disease, avoids potential neurologic, rheumatologic, and cardiac complications. So they develop the most extensively curated dataset thus far for this challenging problem. We evaluate several deep learning models against various problems of growing complexity and on public domain and clinical images. Results suggest that AI can help in prescreening and referring individuals to physicians for earlier diagnosis and treatment.

Advantages : These results suggest that a DL system can help in prescreening and referring individuals to physicians for earlier diagnosis and treatment, in the presence of clinically relevant confusers, thereby reducing further complications and morbidity.

Disadvantages : It requires a very large amount of data in order to perform better than other techniques. It is extremely expensive to train due to complex data models.

[9] A method of skin disease detection using Image Processing and machine learning” has proposed an early detection method on image processing based on Convolutional neural network (CNN) to feature extraction and then using color to identify the features.

Advantages: CNNs can expand the advantages of SVMs, such as robustness in noisy datasets without the need for optimal preprocessing, by capturing image context and extracting high-level features through down-sampling. CNNs can interpret the pixels of an image within its own image-level context, as opposed to viewing each pixel in a dataset-level context.

Disadvantages: CNNs can have trouble generalizing to new domains or by learning unwanted correlations (like the background of an image for example) rather than the desired classes (the foreground).

[10] There are several techniques for image mining that can classify and predict various types of skin disorders based on their classifications. This paper presents image mining techniques. The augmentation, feature extraction and classification. On three different forms of skin disorders (acne, cold sore, hives), all these methods are applied and classification-based identification is performed. The skin analyzing scenario can be organized and performed for early detection of skin disorders with the development of technology. The image and pattern based exploration of different skin disorders, numerous technologies are available. Skin disorders can be categorized by image classification. Image classification is a supervised learning technique that characterizes a lot of objective classes and trains a model to interpret the class.

Advantages: Different image processing techniques were used to achieve a fast and highly precise classifier in the proposed work. Multiple modules for handling different steps are included in the overall process: noise removal, contrast improvement, segmentation, extraction of features and classification (diagnosis).

Disadvantages: Data analytics is a complicated process and often requires people with training to use the tools. The barrier to entry for data analytics can discourage small businesses from using this technology. It can also be difficult to find adequate data that isn't already private or proprietary in nature.

2.2 REFERENCES:

1. Son, H.M., Jeon, W., Kim, J. et al. AI-based localization and classification of skin disease with erythema. Sci Rep 11, 5350 (2021). https://doi.org/10.1038/s41598-021-84593-z

2. J. Xie and C. Wang, "Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases", Expert Systems with Applications, Vol. 38, No. 5, pp. 5809-5815, 2011.

3. Sumithra R, Suhil M, Guru DS. Segmentation and classification of skin lesions for disease diagnosis. Proced. Comput. Sci. 2015;45:76–85. doi: 10.1016/j.procs.2015.03.090. [CrossRef] [Google Scholar].

4. Tang P, Liang Q, Yan X, Xiang S, Sun W, Zhang D, Coppola G. Comput Methods Programs Biomed. 2019 Sep;178:289-301. doi: 10.1016/j.cmpb.2019.07.005. Epub 2019 Jul 8.PMID: 3141655.

5. Baig R, Bibi M, Hamid A, Kausar S, Khalid S. Deep Learning Approaches Towards Skin Lesion Segmentation and Classification from Dermoscopic Images - A Review.Curr Med Imaging. 2020;16(5):513-533. doi: 10.2174/1573405615666190129120449.PMID: 32484086 Review.

6. Lesion border detection in dermoscopy images.Celebi ME, Iyatomi H, Schaefer G, Stoecker WV.Comput Med Imaging Graph. 2009 Mar;33(2):148-53. doi: 10.1016/j.compmedimag.2008.11.002. Epub 2009 Jan 3.PMID: 19121917 Free PMC article.

7. Tan, M. & Le, Q., E cientnet: Rethinking model scaling for convolutional neural networks, in ICML, 6105–6114

8. Lu, J., Manton, J. H., Kazmierczak E. & Sinclair, R., Erythema detection in digital skin images. In 2010 IEEE International Conference on Image Processing, Hong Kong, 2545–2548.

9. Trabelsi, O., Tlig, L., Sayadi, M. & Fnaiech, F., Skin disease analysis and tracking based on image segmentation. 2013 International Conference on Electrical Engineering and So ware Applications, Hammamet, 1–7.

10. Effective diagnosis mechanism for skin disorders using image mining techniques Galiveeti Poornima, Deepak S. Sakkari Published 15 August 2022 Medicine, Computer Science International journal of health sciences

2.3 PROBLEM STATEMENT DEFINITION:

Mr. Narasimma Rao is a 55 year old man. He has been suffering from severe erythema for the past 3years.In these 3 years he faced severe itching , allergy, rashes , Dry and Cracked skin.

- He wants to get better knowledge about the disease he is suffering from.
- He never cared about the infection which became more severe.
- He decided to find the reason behind this infection.
- So,Narasimma Rao needs an immediate remedy from the disease. He wants to know which specialist to get consulted .

Who does the problem affect?

Person who is suffering from erythema.

What are the boundaries of the problem?

Person who might have had a bite of insects and some allergenic plants and sunburn etc..

What is the issue?

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.

When does the issue occur?

It is predicted that it may be due to some allergic plants ,insects and sunburn.

Where does the issue occur?

Redness of the skin caused by injury or another inflammation causing condition.

Why is it important that we fix the problem?

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. So the skin diseases can be prevented by investigating the infected region at an early stage.

What solution to solve this issue?

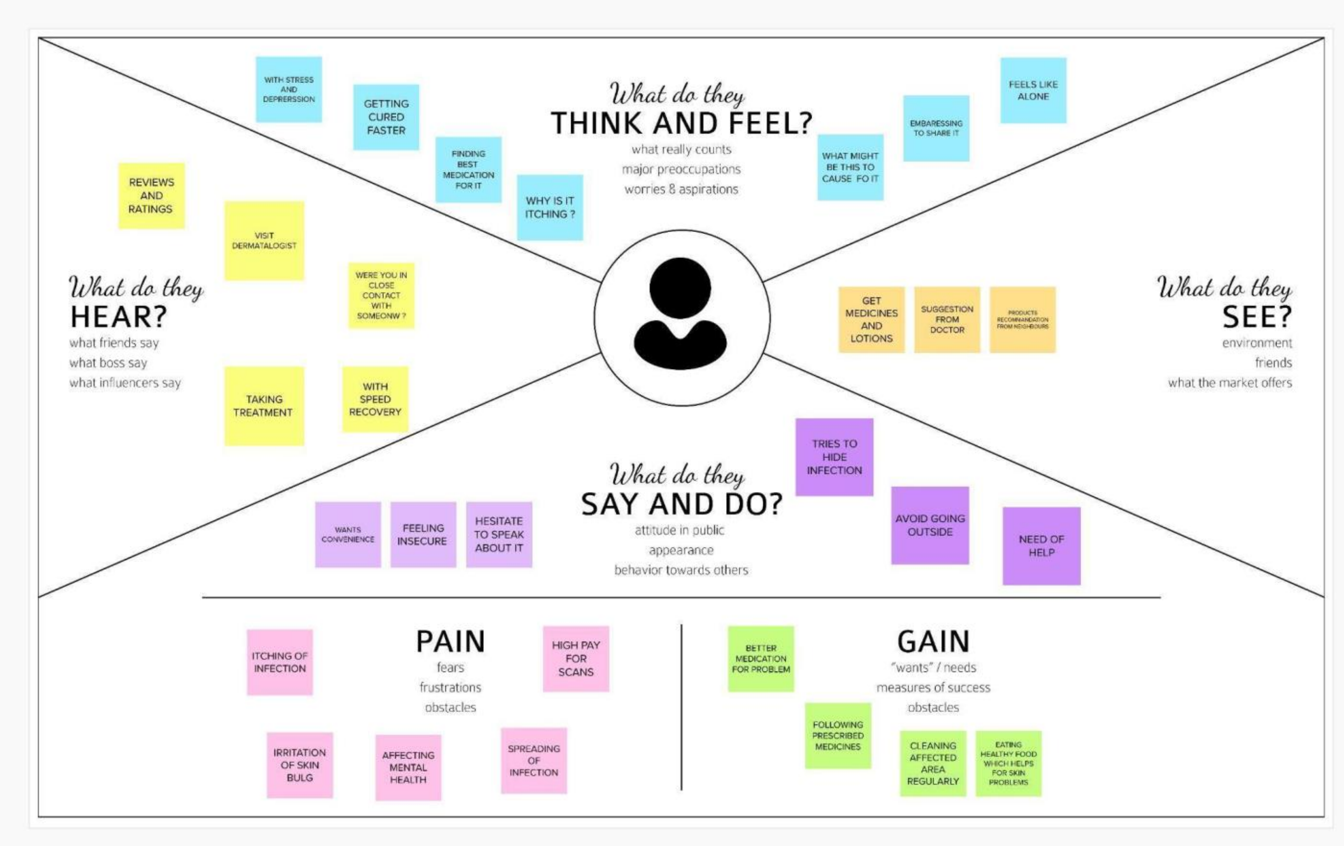
The characteristic of the skin images is diversified so that it is a challenging job to devise an efficient and robust algorithm for automatic detection of skin disease and its severity.

What methodology was used to solve the issue?

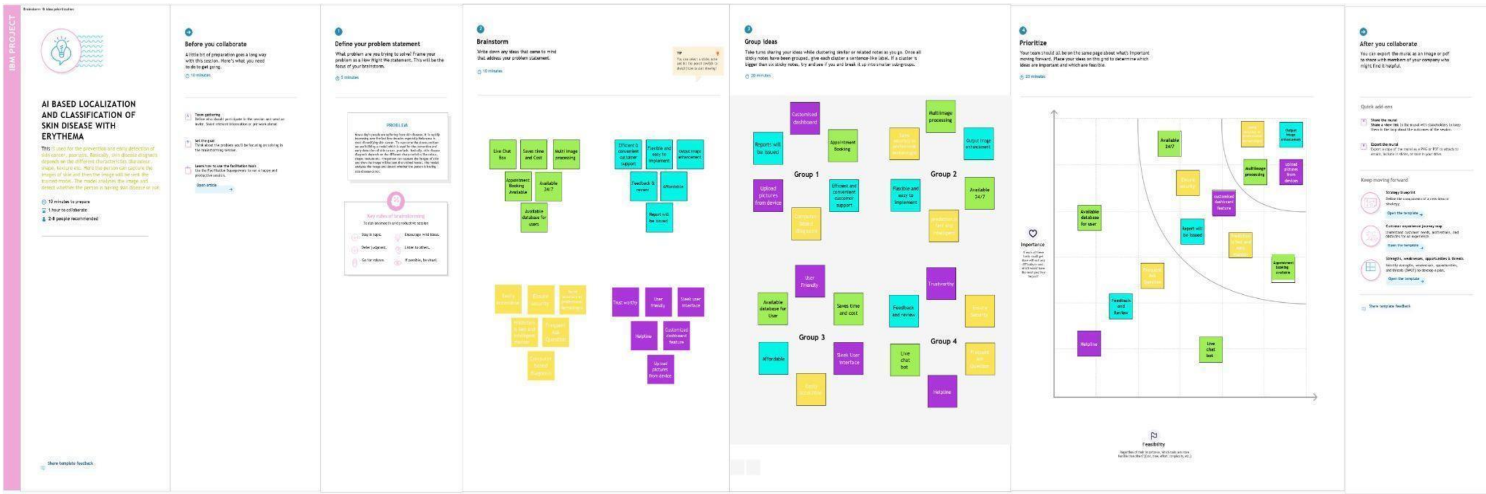
- We are building a model which is used for the prevention and early detection of skin cancer, psoriasis. Basically, skin disease diagnosis depends on the different characteristics like color, shape, texture etc…
- The person can capture the images of skin and then the image will be sent to the trained model. The model analyzes the image and detects whether the person is having skin disease or not.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION & BRAINSTORMING:



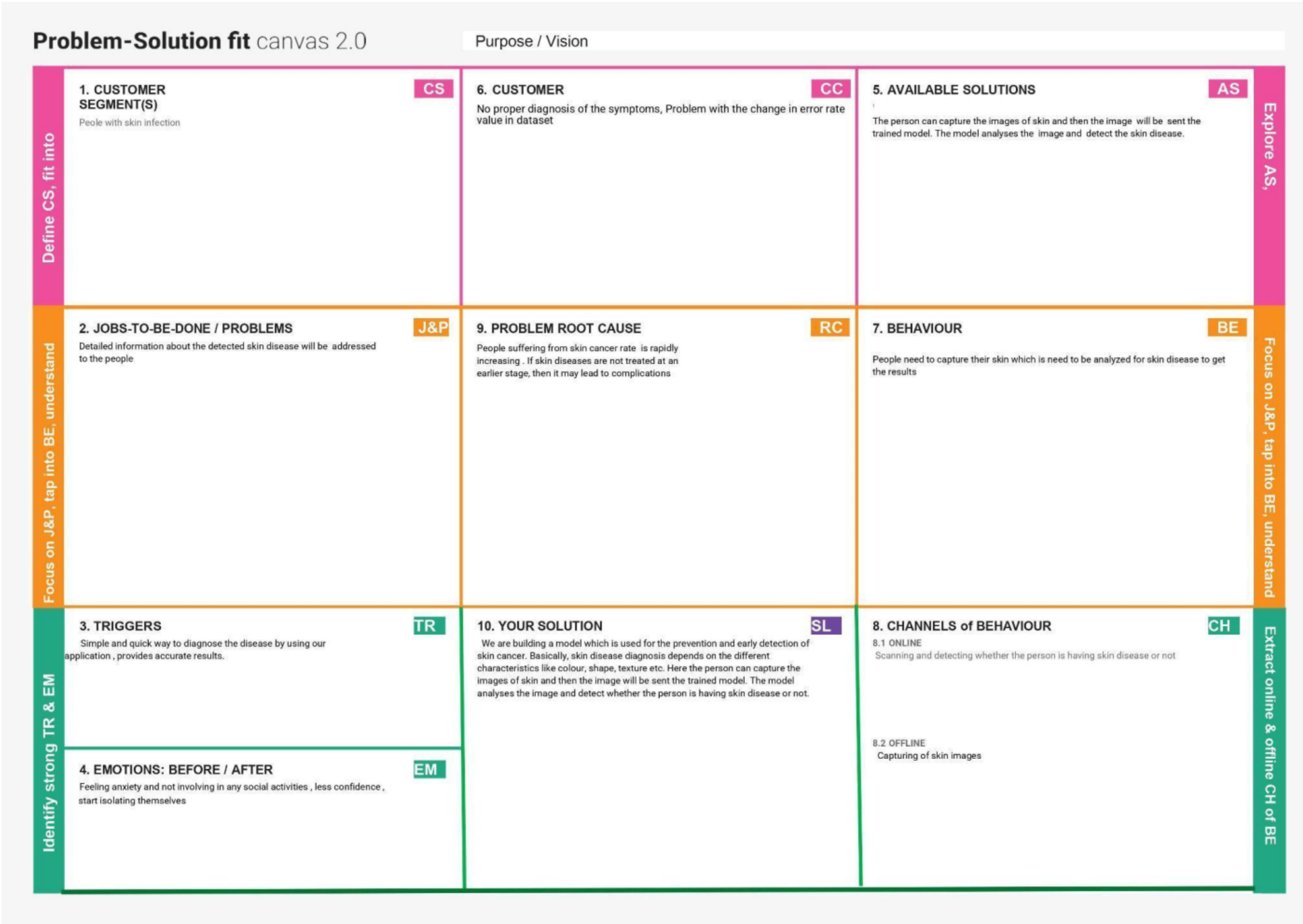
3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Basically, child cannot complain about problems which they face in their day to day life to their parents. They can’t even realize what actually happens to them in their early age. It is also difficult for parents to identify their children are in trouble. Parents can’t monitor their children 24/7

2.	Idea / Solution description	The solution for this problem is to create a child tracker device(Gadget) through which parents can monitor their child’s location anytime. An alert message will be sent to parents or guardian when the child crosses the geofence. An emergency button is given in the device to notify parents, when the child is in trouble. All the data are stored in the database
3.	Novelty / Uniqueness	The novelty of the work is that the system automatically alerts the parent/caretaker by sending notification, when immediate attention is required for the child during emergency.
4.	Social Impact / Customer Satisfaction	The parents need not worry about their child’s location and safety as they will get alert messages in case of any trouble.
5.	Business Model (Revenue Model)	The model of the gadget is wearable device. like watch. That consist the GPS to track the location of the person .The device is cost efficient, and easily wearable. Because the device was used by the person everyday.

6.	Scalability of the Solution	The scalability of the solution is that we can use the gadget 24 hours that can sense and send the information to the parents and guardians in the right way and at the right time.

3.4 PROBLEM SOLUTION FIT:



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile Number Registration through Google Account
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Patient Image Capturing Process	Providing Access to Capture Image Through Camera Provide Access to Upload Image Through Gallery.
FR-4	Patient Medicine Reminder	Remind the patients to take their Medicine/ointments At the right time through the remaining alarm.
FR-5	Suggestion Box	Patients can take suggestions from the doctors through chats.
FR-6	Flareup Cycles	Patients can know their medicine level from doctors through messages.

4.2 NON-FUNCTIONAL REQUIREMENTS:

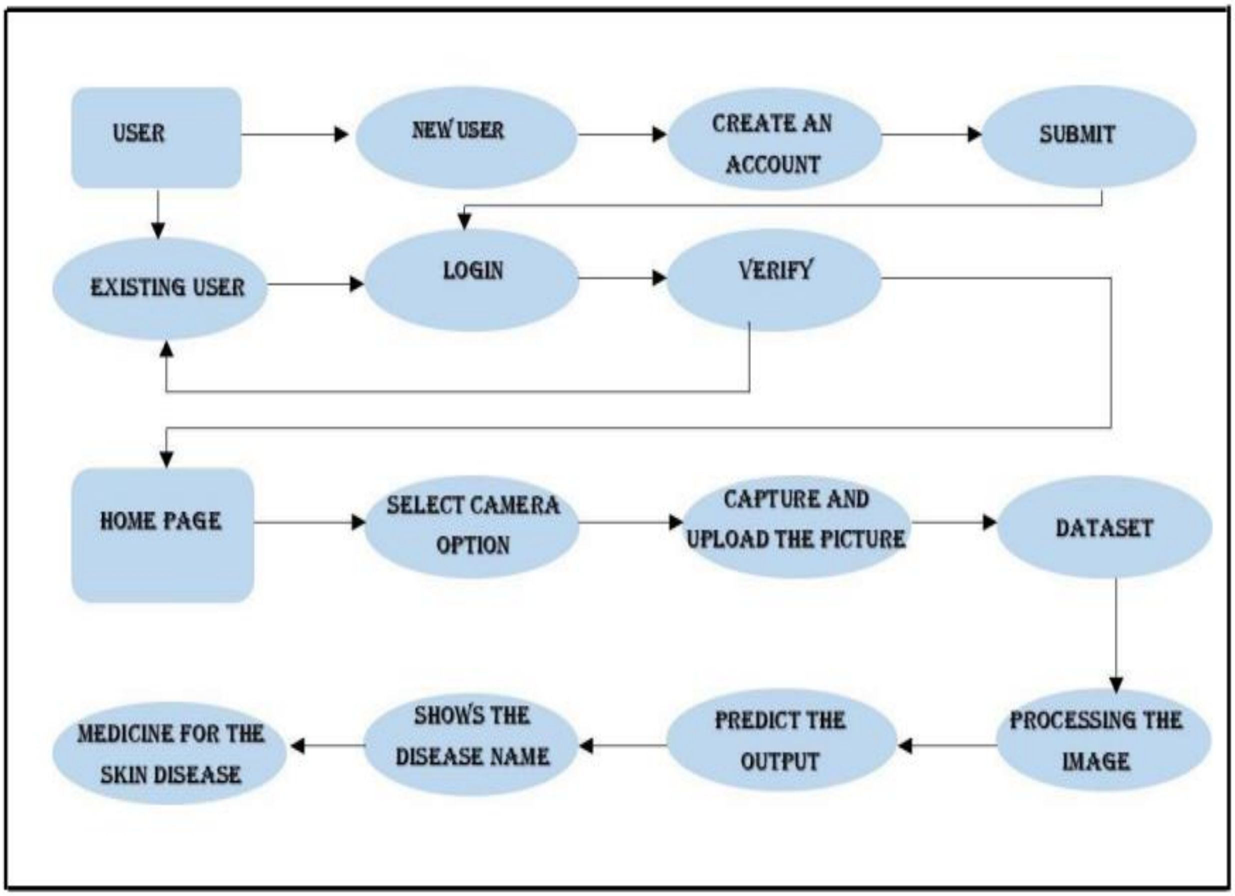
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Our Mobile phone application designed to improve the quality of patient-held photos, and was developed to generate and hold their own skin images to help guide their skin care.
NFR-2	**Security**	Data privacy and security practices may vary based on users and their age
NFR-3	**Reliability**	Easy to use app to get personalized answers to your skin conditions questions
NFR-4	**Performance**	Good treatments are available for a variety of skin conditions including rash, itchy skin, skin fungus etc.
NFR-5	**Availability**	Our app helps you to screen your skin symptoms and prepare for your practitioner visit.

NFR-6	**Scalability**	The app gives users evidence-based dermatologist approved health information insights on diseases affecting various parts of our body.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:

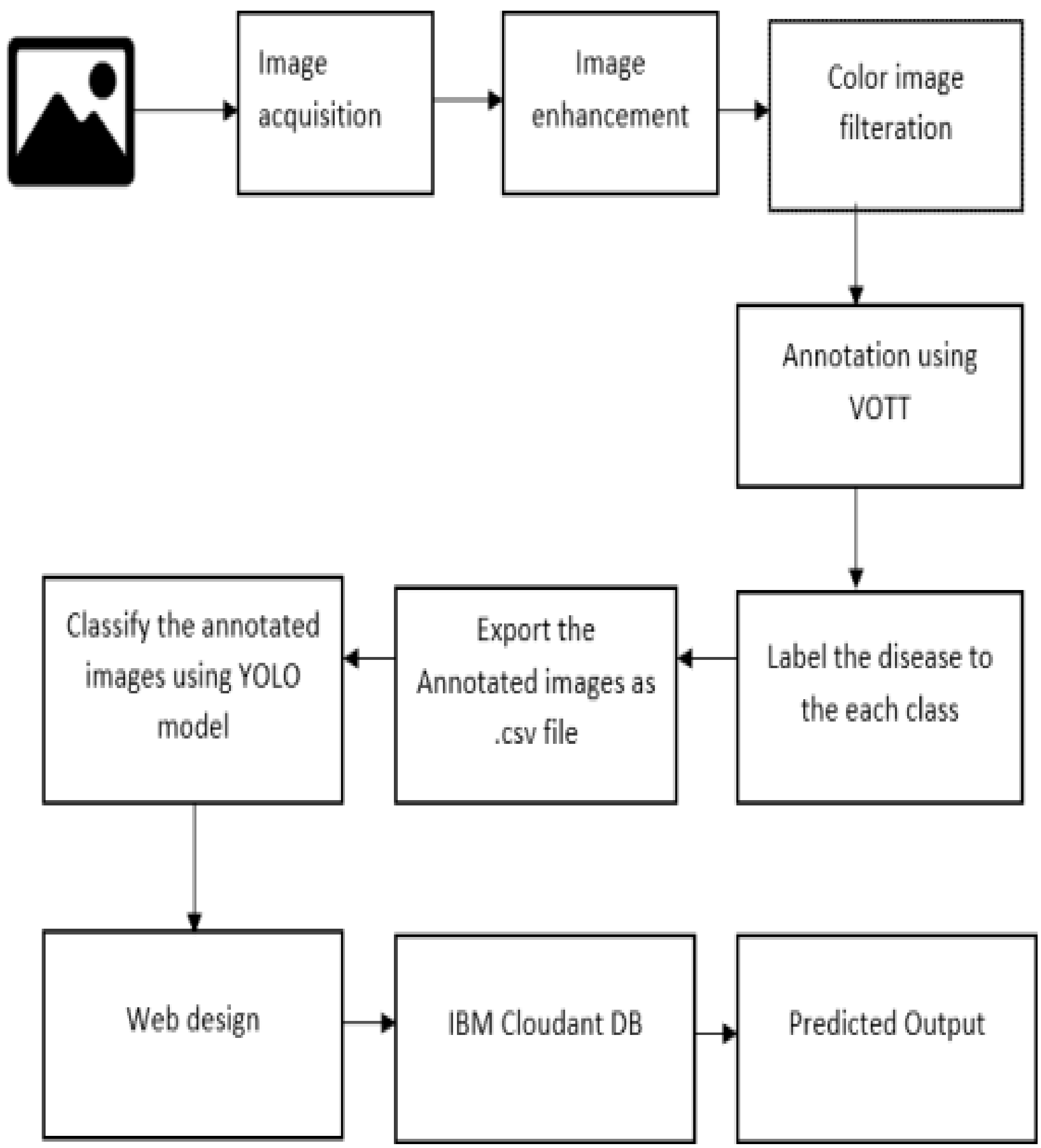
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE:

Child tracker helps the parents in continuously monitoring the child's location. They can simply leave their children in school or parks and create a geofence around the particular location. By continuously checking the child's location notifications will be generated if the child crosses the geofence. Notifications will be sent according to the child's location to their parents or caretakers. The entire location data will be stored in the database.

SOLUTION ARCHITECTURE



5.3 USER STORIES:

Use the below template to list all the user stories for the product.

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 mark a user story complete.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	Akilesh.J.S.
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application.	2	Medium	Vikas Shrinivas Naik.
Sprint-2		USN-3	As a user, I can register for the application through Mobile number.	3	High	Vikram.C.M.
Sprint-2		USN-4	As a user, I will receive a conformation SMS.	3	High	Pagadala Davidlivingston.
Sprint-2	Login	USN-5	As a user, I can log into the application by entering login credentials.	1	High	Puvarasan.V
Sprint-3	Dashboard	USN-6	As a user, I can upload my images and get my details of skin diseases.	3	High	Vikram.C.M. Akilesh.J.S.

6.2 SPRINT DELIVERY SCHEDULE:

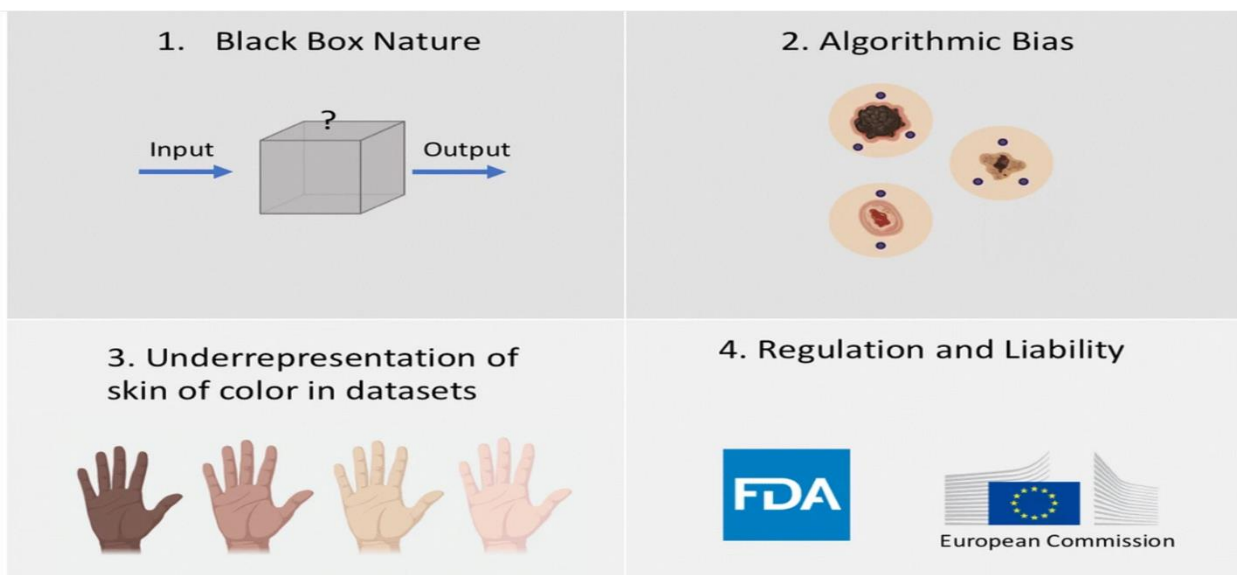
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

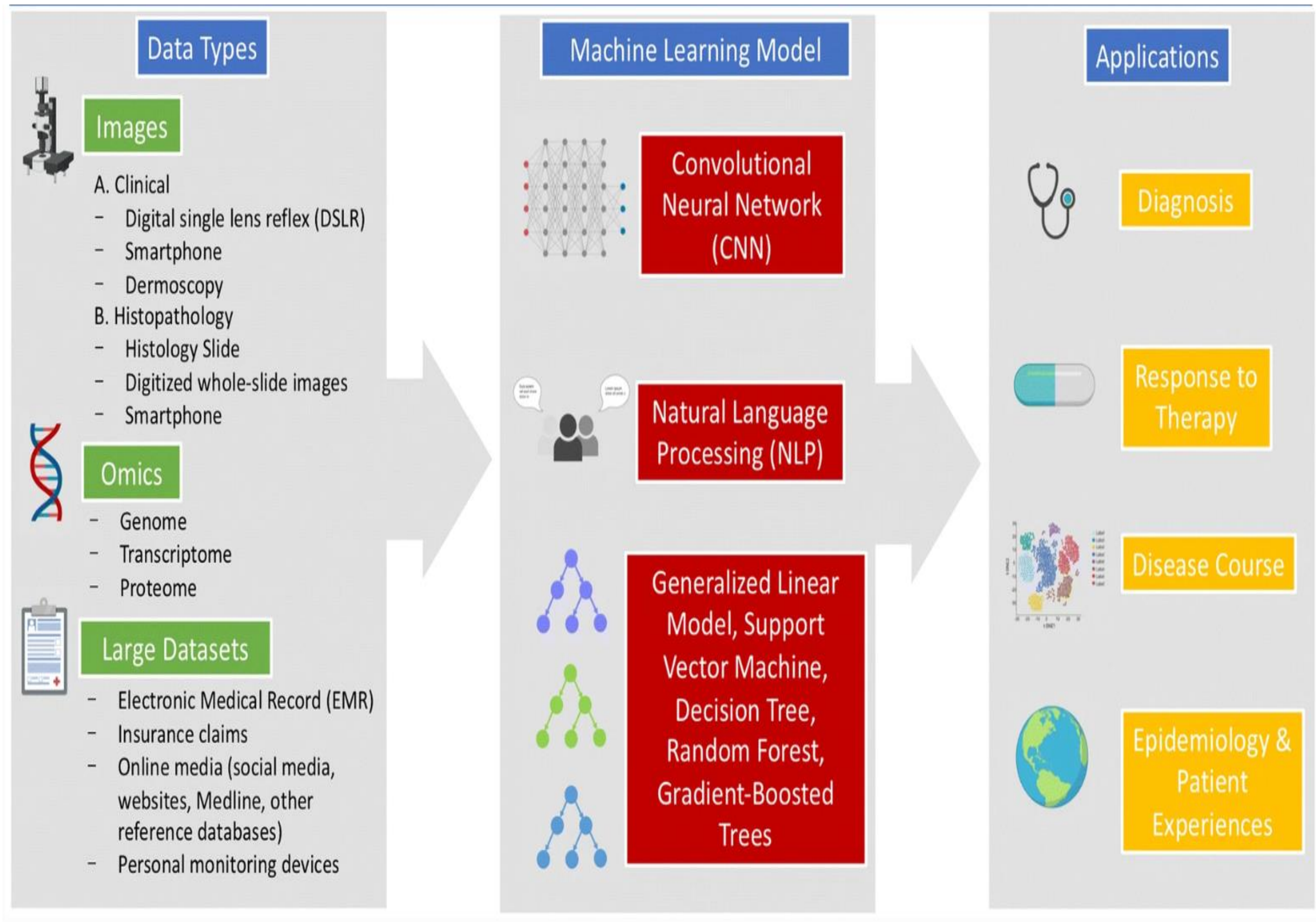
Velocity:

Imagine we have a 10-day 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)

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

6.3 REPORTS FROM JIRA:





7. CODING AND SOLUTIONING

7.1 FEATURE 1:

import re

import numpy as np

import os

from flask import Flask, url_for, render_template, request, redirect, session

import sys

from cloudant.client import db

from flask import flask , app,request,render_template

```
import argparse
from tensorflow import keras
from PIL import Image
from timeit import default_timer as timer
import test
import pandas as pd
import numpy as np
import random

from cloudant.client import Cloudant
client = Cloudant.iam('14f4d73b-0587-43e7-a135-8f0b1ee838e6-bluemix',
'_w4Lugdv8gtUI45fihmyuboqra99qGXGD9Ey1hrr6Grz', connect=True)
my_database = client.create_database('my_database')

def get_parent_dir(n=1):
""" returns the n-th parent directory of the current
working 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
src_path = r'C:\Users\akile\OneDrive\Desktop\yolo_structure\2_Training\src'
print(src_path)
utils_path = r'C:\Users\akile\OneDrive\Desktop\yolo_structure\Utils'
print(utils_path)
```

```
sys.path.append(src_path)
sys.path.append(utils_path)
import argparse
from keras_yolo3.yolo import YOLO, detect_video
from PIL import Image
from timeit import default_timer as timer
from utils import
load_extractor_model,load_features,parse_input,detyect_object
import test
import utils
import pandas as pd
import numpy as np

from Get_File_Paths import GetFileList
import random
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "3"
#set up folder names for default values
data_folder = os.path.join(get_parent_dir(n=1),"Skin Disease-Flask", "Data")
image_folder = os.path.join(data_folder,"Source_Images")
image_test_folder = os.path.join(image_folder, "Test_images")
detection_results_folder =
os.path.join(image_folder,"Test_Image_Detection_Results")
detection_results_file = os.path.join(detection_results_folder ,
"Detection_results.csv")
model_folder = os.path.join(data_folder , "Model_Weights")
```

```
model_weights = os.path.join(model_folder , "trained_weights_final.h5")
model_classes = os.path.join(model_folder, "data_classes.txt")
anchors_path = os.path.join(src_path ,
"keras_yolo3","model_data","yolo_anchors.txt")
FLAGS = None
app = Flask(__name__)
app.debug = True

@app.route('/', methods=['GET'])
def index():
if session.get('logged_in'):
return render_template('Index.html')
else:
return render_template('index.html')

@app.route('/register/', methods=['GET', 'POST'])
def register():
if request.method == 'POST':
try:
db.session.add(User(username=request.form['username'],

password=request.form['password']))
db.session.commit()
return redirect(url_for('login'))
except:
```

```
return render_template('index.html', message="User Already

Exists")
else:
return render_template('register.html')

@app.route('/login/', methods=['GET', 'POST'])
def login():
if request.method == 'GET':
return render_template('login.html')
else:
u = request.form['username']
p = request.form['password']
data = User.query.filter_by(username=u, password=p).first()
if data is not None:
session['logged_in'] = True
return redirect(url_for('index'))
return render_template('index.html', message="Incorrect Details")

@app.route('/logout', methods=['GET', 'POST'])
def res():
# Delete all default flags
parser = argparse.ArgumentParser(argument_default=argparse.SUPPRESS)
#Command Line options
```

```
parser.add_argument(
"--input_path",
type=str,
default=image_test_folder,
help="Path to image/video directory. ALL subdirectories will be
included. Default is "
+image_test_folder,
)
parser.add_argument(
"--output",
type=str,
default=detection_results_folder,
help="Output path for detection results. Default is "
+ detection_results_folder,
)

parser.add_argument(
"--no_save_img",
default=False,
action="store_true",
help="Only save bounding box coordinates but do not save output imoges
with annotated boxes. Default is False.",
)
parser.add_argument(
"--file_types",
```

```
" -names-List",
nargs="*",
default=[],
help="Specify list of file types to include. Default is --file_types
.jpg .jpeg .png .mp4",
)
parser.add_argument(
"--yolo_model",
type=str,
dest="model_path",
default=model_weights,
help="Path to pre-trained weight files. Default is + model_weights",
)
parser.add_argument(
"--anchors",
type=str,
dest="anchors_path",
default=anchors_path,
help="Path to YOLO anchors. Default is " + anchors_path,
)
parser.add_argument(
"--classes",
type=str,
dest= "classes_path",
```

```
default=model_classes,
help="Path to YOLO class specifications. Default is "+model_classes,
)
parser.add_argument(
"--confidence",
type=float,
dest="score",
default=0.25,
help="Threshold for YOLO object confidence score to show predictions.
Default is 8.25.",
)
parser.add_argument(
"--box_file",
type=str,
dest="box",
default=detection_results_file,
help="File to save bounding box results to. Default is "
+ detection_results_file,
)
parser.add_argument(
"--postfix",
type=str,
dest="postfix",
default=" disease", help="Specify the postfix for images with bounding
boxes. Default is disease",
```

```
)

FLAGS= parser.parse_args()
save_img = not FLAGS.no_save_img
file_types = FLAGS.no_save_img
#print(input_path)
if file_types:
input_paths =GetFileList (FLAGS.input_path, endings=file_types)
print(input_paths)
else:
input_paths =GetFileList(FLAGS. input_path)
print(input_paths)
# Split images and videos
img_endings = (".jpg", ".jpeg", ".png")
vid_endings = (".mp4", ".mpeg", ".mpg", ".avi")
input_image_paths = []
input_video_paths = []
for item in input_paths:
if item.endswith(img_endings):
input_image_paths.append(item)
elif item.endswith(vid_endings):
input_video_paths.append(item)
output_path = FLAGS.output
if not os.path.exists(output_path):
os.makedirs (output_path)
```

```
# define YOLO detector
yolo = YOLO(
**{
"model_path": FLAGS.model_path,
"anchors_path": FLAGS.anchors_path,
"classes_path": FLAGS.classes_path,
"gpu_num": FLAGS.gpu_num,
"score": FLAGS.score,
"model_image_size": (416, 416),
}
)

# Make a dataframe for the prediction outputs
out_df =pd.DataFrame(
columns=[
"image",
"image_path",
"xin",
"ymin",
"xmax"
"ymax",
"Label",
"confidence",
"x_sixe",
"y_size",
```

```
]
)
# labels to draw on images
class_file=open (FLAGS.classes_path, "r")
input_labels= [line.rstrip("\n") for line in class_file.readlines()]
print("Found () input Labels: {}".format(len(input_labels), input_labels))
if input_image_paths:
print(
"Found () input images: {}".format( len(input_image_paths),
[os.path.basename(f) for f in input_image_paths[:5]],
)
)
start=timer()
text_out=""
#This is for images
for i, img_path in enumerate(input_image_paths):
print(img_path)
prediction,image,lat,lon=detect_object(
yolo,
img_path,
save_img=save_img,
save_img_path=FLAGS.output,
postfix=FLAGS.postfix,
)
print(lat,lon)
```

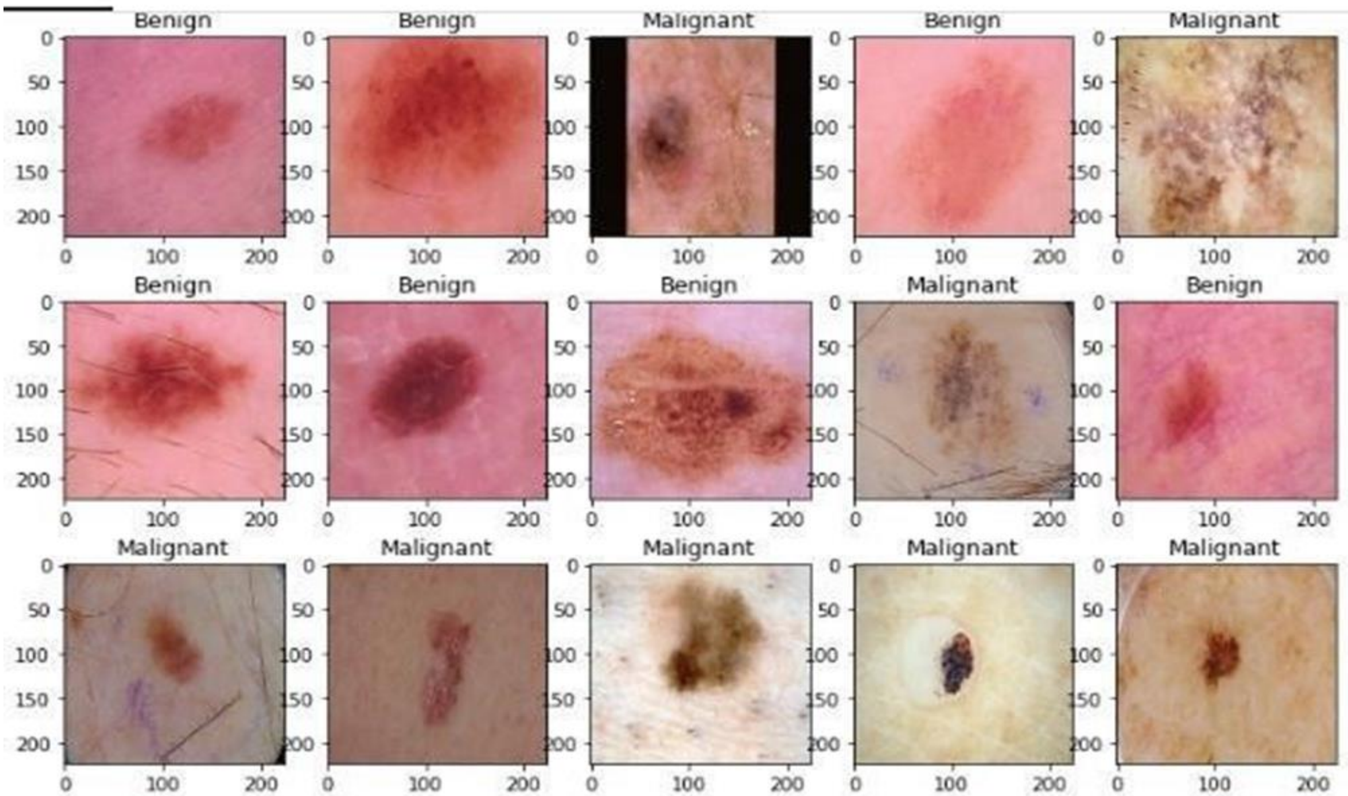
```
y_size, x_size, _ = np.array(image).shape

for single_prediction in prediction:
out_df = out_df.append(
pd.DataFrame(
[
[
os.path.basename(img_path.rstrip("\n")),
img_path.rstrip("\n"),
]
+ single_prediction
+ [x_size,y_size]
],
columns=[
"image",
"image_path",
"xmin",
"ymin",
"xmax",
"ymax",
"label",
"confidence",
"x_size",
"y_size",
],
```

```
)
)
end = timer()
print(
"Processed {} videos in {:.1f}sec".format(
len(input_video_paths) / (end - start),
)
)
out_df.to_csv(FLAGS.box , index=False)
# this is for videos
if input_video_paths:
print(
"found {} input videos: {} ...".format(
len(input_video_paths),
[os.path.basename(f) for f in input_video_paths[:5]],
)
)
start = timer()
for i , vid_path in enumerate(input_video_paths):
output_path = os.path.join(
FLAGS.output,
os.path.basename(vid_path).replace(".",FLAGS.postfix + "."),
)
detect_video(yolo , vid_path,output_path=output_path)
```

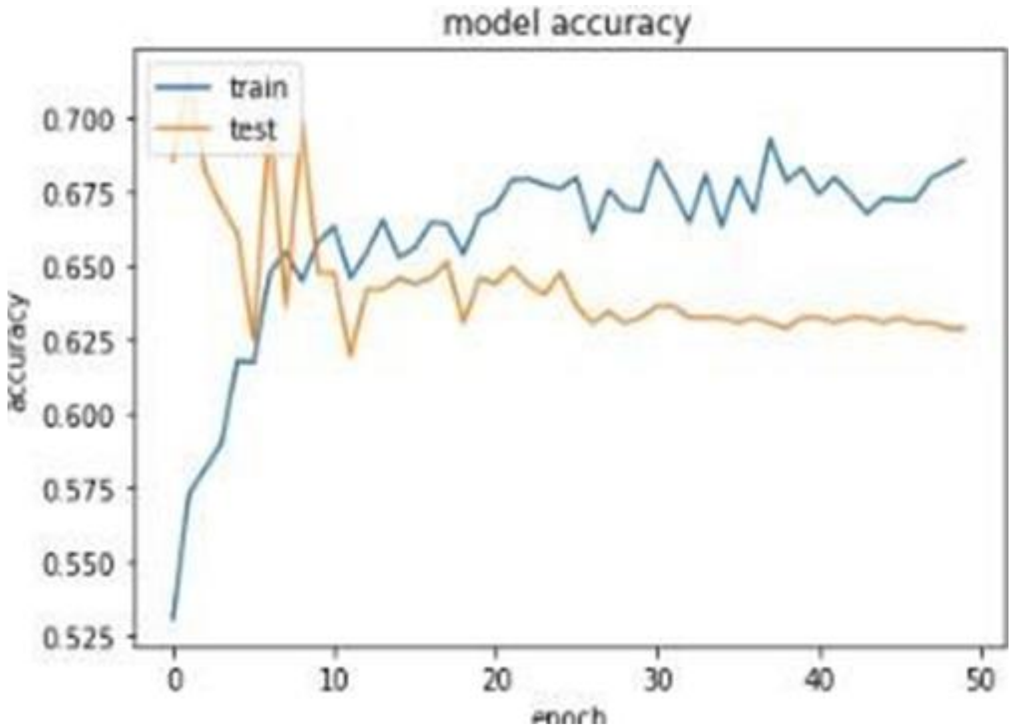
```
end = timer()
print(
"Processed {} videos in {:.1f}sec".format(
len(input_video_paths), end - start
)
)
# CLose the current yolo session
yolo.close_session()
return render_template ('prediction.html')
def logout():
session['logged_in'] = False
return redirect(url_for('index'))
if(__name__ == '__main__'):
app.run()
```

9. RESULTS

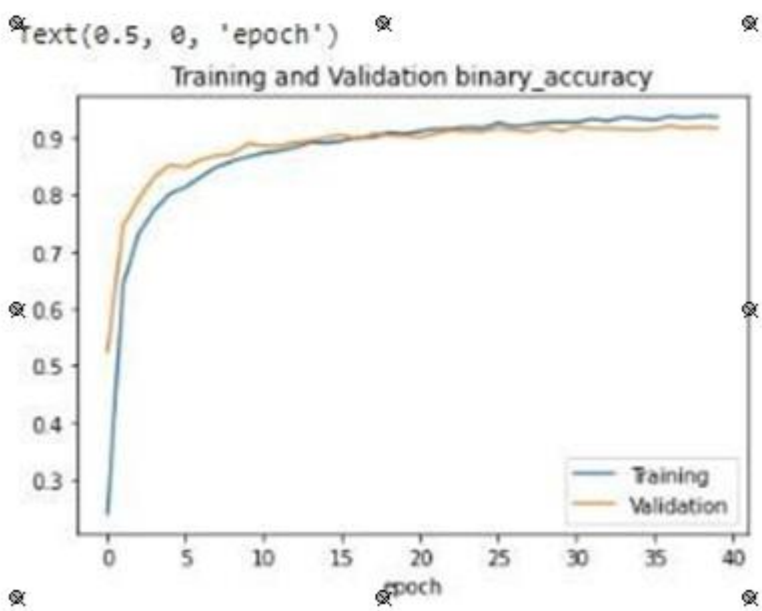
9.1 Testing image (Skindatasetimages):



Model training accuracy:



Training and testing accuracy:



10.ADVANTAGES & DISADVANTAGES

10.1 Advantages:

* High Enhancement.
* No therapeutic limitation.
*Easily incorporate depot.
* Delivery not majorly affected by diseased state of skin.

10.2 Disadvantages:

* High cost.
*Security concern

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 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.

12. FUTURE SCOPE

In future, this machine learning model may bind with various websites which can provide real-time data for skin disease prediction. Also, we may add large historical data on skin disease which can help to improve the accuracy of the machine learning model. We can build an android app as a user interface for interacting with the user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates, and train it on clusters of data rather than the whole dataset.

13. APPENDIX

GITHUB : https://github.com/IBM-EPBL/IBM-Project-37770-1660323038

PROJECT DEMO LINK: https://youtu.be/_OL1MYRsvOk