AI-based localization and classification of skin disease with erythema

A PROJECT REPORT

Submitted by

DIVYA S(19EUEC040)

DINESHKUMAR N (19EUEC039)

DINESH T (19EUEC038)

DURAIYARASU S(19EUEC041)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University Chennai -600025)

NOVEMBER 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

BONAFIDE CERTIFICATE

Certified that this project report titled **"AI-based localization and classification of skin disease with erythema"** is the bonafide work of **Ms.DIVYA S(19EUEC040)**, **Mr.DINESHKUMAR N(19EUEC039), Mr.DINESH T(19EUEC038), Mr.DURAIYARASU S(19EUEC041)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.SASIPRIYA M.E., Ph.D.,	**Mr.C.VISVESVARAN M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**

Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Kuniamuthur,Coimbatore

Submitted for the Project viva-voce examination held on _____

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ABSTARCT

Skin diseases are a major and worrying problem in societies due to their physical and psychological effects on patients. Detecting skin diseases at an early stage has an important role in treatment. The process of diagnosing and treating skin injury is related to the skill and experience of the specialist doctor. The diagnostic process must be accurate and timely. Recently, artificial intelligence science has been used in the field of diagnosing skin diseases through the use of machine learning algorithms and the exploitation of the vast amount of data available in health centres and hospitals. In this paper, quite many previous studies related to methods of classification of skin diseases based on the principle of machine learning were collected. In a group of previous studies, the researchers used some systems, mechanisms, and algorithms. Several systems have been successful in classifying skin diseases and achieving varying diagnostic accuracy. Various systems have relied on methods of image processing and feature extraction that help predict and detect disease type. There are other systems designed to identify specific types of skin disease through clinical features and features obtained from tissue analyses after a skin biopsy of the affected area. This survey shows that the diagnostic accuracy in image processing methods was relatively uneven, ranged between (50% to 100%). As for the methods of treating tissue features, the accuracy was of an excellent level of 94% or more. The results provide an overview of the actual relevant studies found in the literature and highlight most of which research gaps have emerged.

TABLE OF CONTENTS

LIST OF FIGURES

1.INTRODUCTION

1.1.OVERVIEW

Skin diseases are the large number of spread diseases in the world. Their diagnoses are very difficult because of its difficulties in skin texture, presence of hair on skin and colours. It is required to develop methods like machine learning in order to increase the accuracy of diagnosis for various types of skin diseases. Machine learning techniques are widely used in medical fields for diagnosis. These algorithms use feature values from images as input to make a decision. The process consists of three stages-The feature extraction stage, the training stage and the testing stage. The process makes use of machine learning technology to train itself with the various skin images. The objective of this process is to increase accuracy of skin disease detection. Three important features in image classification are texture, colour, shape, and combination of these. In this work, colour and texture features are used to classify the skin disease. Normal skin colour is different from the skin with disease. Smoothness, coarseness, and regularity is effectively identified using texture features in the images. Hence, these two features are explored to identify skin disease effectively. In this work, entropy, variance and maximum histogram value of Hue-Saturation-Value(HSV) features are used. These features are used to build machine learning algorithm by using Decision Tree(DT) and Support Vector Machine(SVM). At first level, entropy measure is used to split the tree. At second level, variance is used to get leaves for textures. In colour features, maximum histogram value of HSV measure is used to split the tree. Accuracy is used to test the performance of the proposed algorithm.

1.2.PURPOSE

The purpose of this process is to increase accuracy of skin disease detection. Three important features in image classification are texture, colour, shape, and combination of these. The skin covers the whole body. An important function of the skin is to protect the body from infection. We need automated computer application for skin disease classification to be installed at the medical facilities like rural health clinics for remote areas where skin specialists are unavailable.

2.LITERATURE SURVEY

2.1. EXISTING PROBLEM

Medicine is an area that is not yet fully understood. Information is not completely transparent. The characteristics of dermatology determine that the majority of the data cannot be obtained. At the same time, the AI technology route is immature, the identification accuracy of which must be improved owing to the uncertainty of manual diagnosis. There is no strict correspondence between the symptoms and results of a disease and no clear boundary between the different diseases. Thus, the use of deep learning for disease diagnosis continues to require considerable effort.

Before systematic debugging, extensive simulation, and robust validation, flawed algorithms could harm patients, which could lead to medical ethical issues, and therefore require forward-looking scrutiny and stricter regulation. As a “black box”, the principle of deep learning is unexplained at this stage, which could result in unpredictable system output. Moreover, it is possible that humans could not truly understand how a machine functions, even though it is actually inspired by humans. Hence, whether or not patient care can be accepted using an opaque algorithm remains a point of discussion.

2.2. REFERENCES

[1] Doi, K. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. Comput. Med. Imaging Graph. 31, 198–211.

[2]Yoshida, H. & Dachman, A. H. Computer-aided diagnosis for CT colonography. Semin. Ultrasound CT MRI 25, 419–431.

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

[4]Rajab, M. I., Woolfson, M. S. & Morgan, S. P. Application of region-based segmentation and neural network edge detection to skin lesions. Comput. Med. Imaging Graph. 28, 61–68.

[5]Keke, S., Peng, Z. & Guohui, L., Study on skin color image segmentation used by fuzzy-c-means arithmetic. In 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, 612–615.

[6]Hongmao, S. Quantitative Structure-Activity Relationships: Promise, Validations, and Pitfalls in A Practical Guide to Rational Drug Design 163–192 (Woodhead Publishing, Sawston, 2016).

[7]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.

[8]Sumithra, R., Suhil, M. & Guru, D. S. Segmentation and classification of skin lesions for disease diagnosis. Proced. Comput. Sci. 45, 76–85.

[9]Maglogiannis, I., Zafiropoulos, E. & Kyranoudis, C. Intelligent segmentation and classification of pigmented skin lesions in dermatological images in Advances in Artificial Intelligence. SETN 2006.

[10]Tsumura, N., Haneishi, H. & Miyake, Y. Independent-component analysis of skin color image. J. Opt. Soc. Am. A 16, 2169–2176.

[11]Albawi, S., Mohammed, T. A. & Al-Zawi, S., Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), Antalya, 1–6.

[12]Selvaraju, R. et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. 128, 336–359.

[13]Gutman, D., Codella, N., Celebi, E., Helba, B., Marchettic, M., Mishra, N., & Halpern, A., Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on [14]Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC).

[15]Codella, N., Gutman, D., Celebi, ME., Helba, B., Marchetti, MA., Dusza, S., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., & Halpern, A., Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC).

[16]Taha, A. & Hanbury, A. An efficient algorithm for calculating the exact hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 37(11), 2153–2163.

[17]Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

[18] M. Chen, P. Zhou, D. Wu, L. Hu, M. M. Hassan, and A. Alamri, "AI-Skin: Skin disease recognition based on self-learning and wide data collection through a closed-loop framework," Inf. Fusion, vol. 54, pp. 1–9, 2020.

[19] N. I. A. Dabowsa, N. M. Amaitik, A. M. Maatuk, and S. A. Aljawarneh, "A hybrid intelligent system for skin disease diagnosis," Proc. 2017 Int. Conf. Eng. Technol. ICET 2017, vol. 2018-Janua, pp. 1–6, 2018.

[20] S. Chan, V. Reddy, B. Myers, Q. Thibodeaux, N. Brownstone, and W. Liao, "Machine Learning in Dermatology: Current Applications, Opportunities, and Limitations," Dermatol. Ther. (Heidelb)., vol. 10, no. 3, pp. 365–386, 2020.

[21] A. Mobiny, A. Singh, and H. Van Nguyen, "Risk-Aware Machine Learning Classifier for Skin Lesion Diagnosis," J. Clin. Med., vol. 8, no. 8, p. 1241, 2019.

[22] D. de A. Rodrigues, R. F. Ivo, S. C. Satapathy, S. Wang, J. Hemanth, and P. P. R. Filho, "A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system," Pattern Recognit. Lett., vol. 136, pp. 8–15, 2020, doi: 10.1016/j.patrec.2020.05.019.

[23] W. Lei, R. Zhang, Y. Yang, R. Wang, and W. S. Zheng, "Class-Center Involved Triplet Loss for Skin Disease Classification on Imbalanced Data," Proc. - Int. Symp. Biomed. Imaging, vol. 2020-April, pp. 16–20, 2020.

2.3. PROBLEM STATEMENT DEFINITION

The data used for evaluation are frequently overly small to allow a convincing statement regarding a system's performance to be made. Although it is not impossible to collect an abundance of relevant data through the Internet in this information age, this information, with significant uncertainty, apparently cannot meet the requirements of independent and identical distribution, which is one of the important prerequisites for deep learning to be successfully applied. For certain rare diseases and minorities, only a limited number of images are available for training. To date, a large number of algorithms have demonstrated prejudice against minority groups, which could cause a greater gap in health service between the "haves" and the "have-nots" [105]. Numerous cases are required for the training process using deep learning techniques. In addition, although the deep learning technique has been successfully applied to other tasks, the developed models in skin are valid in only specific dedicated diseases and are not applicable to common situations. Diagnosing dermatology is a complex process that, in addition to image recognition, must be supplemented by other means such as palpation, smell, temperature change, and microscopy.

Fig.1.1.Problem Statement

3.IDEATION AND PROPOSED SOLUTION

3.1.EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy Map for Plasma Donor Application :

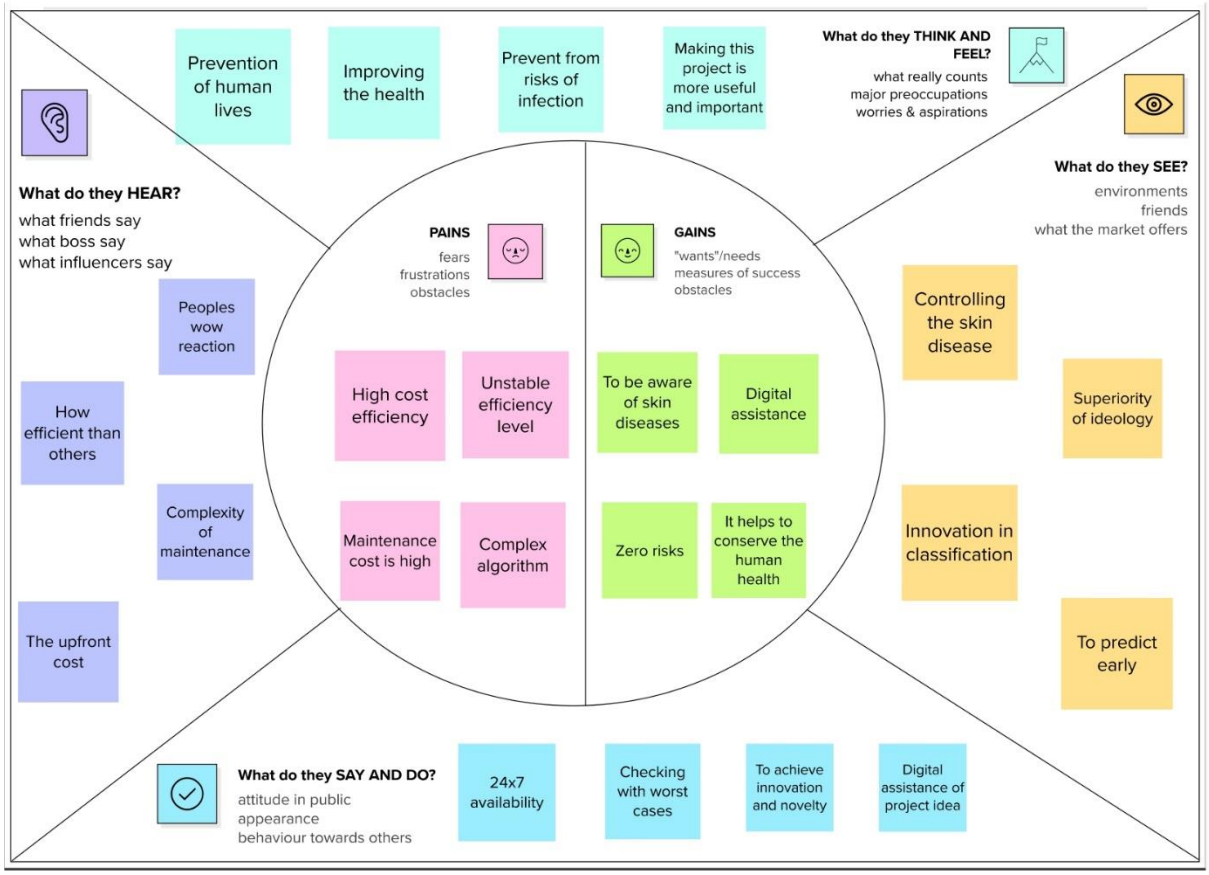

Fig.1.2.Empathy map

3.2.IDEATION AND BRAINSTROMING

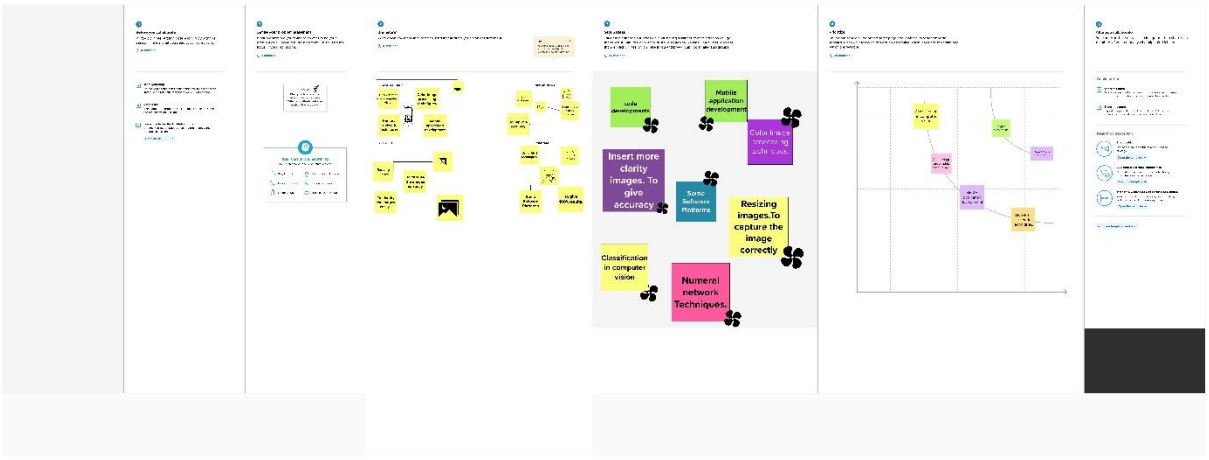

Fig.1.3.Ideation and Brainstroming

3.3.PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem of skin disease by bacteria, fungus and some other viruses trapped in skin pores and hair follicles.
2.	Idea / Solution description	Classification of skin disease using Artificial Intelligence based image processing in order to be aware from the risks of infection.
3.	Novelty / Uniqueness	Classification is done with computer aided diagnosis, the medical imaging will become easier than others, since it uses the artificial intelligence techniques in order improve the accuracy of dermatology diagnosis.
4.	Social Impact / Customer Satisfaction	The physical problems like irritation and itching, they will be also feeling the psychological problems like stress, anxiety can be prevented.
5.	Business Model (Revenue Model)	Application
6.	Scalability of the Solution	Computers or mobile phones are portable. The effects of skin disease could be avoided by using image processing technique in order to achieve the classification of skin disease.

3.4. PROBLEM SOLUTION FIT

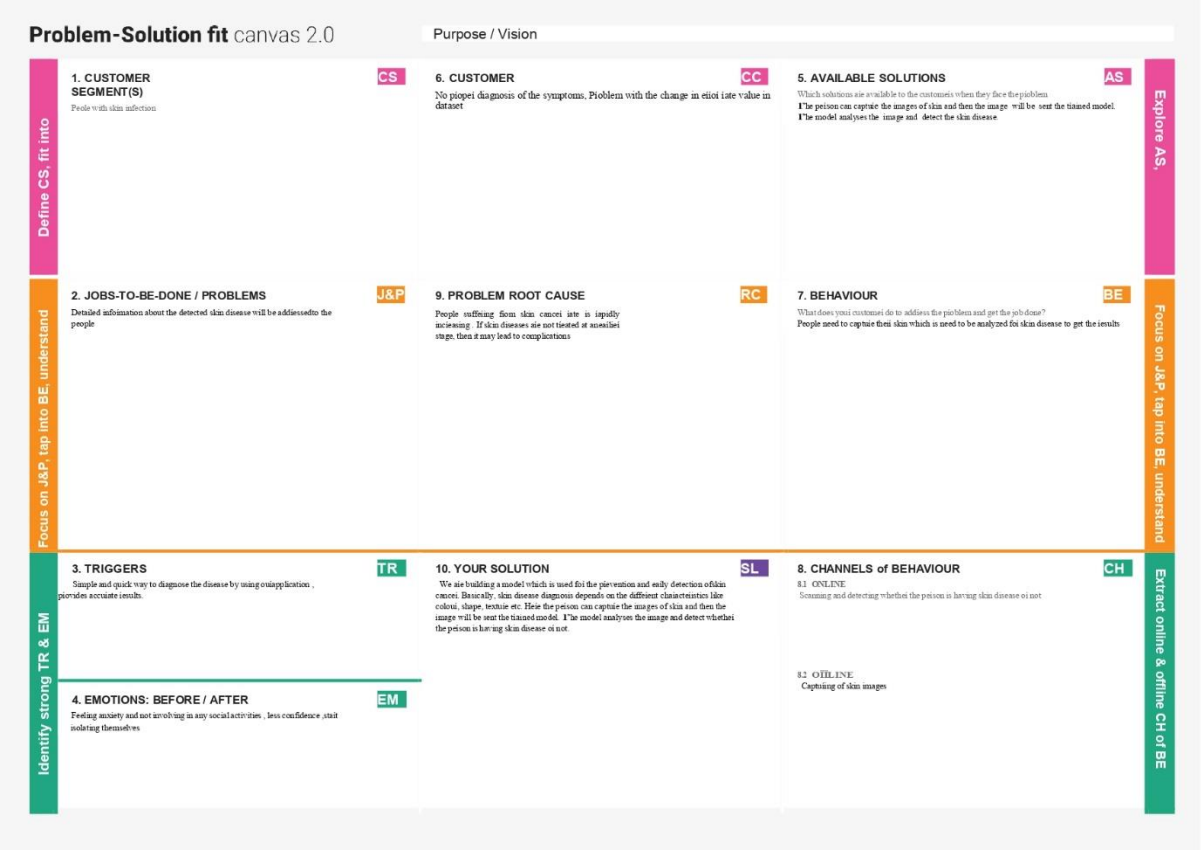

Fig.1.4.Problem Solution Fit

4.REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	User will provide their medical details and save in thesystem
FR-4	Input	User capture the skin which is affected or upload the taken image as jpeg format
FR-5	Output Analysis	Image will be processed through YOLO and othertrained model
FR-6	Provides Description	Gives detailed description of type of the skin disease affected

4.2.NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to classify skin disease with erythema
NFR-2	**Security**	It offers greater security and prevents unauthorized users to access the data.
NFR-3	**Reliability**	Even with more users, there will be a good performance without failure.
NFR-4	**Performance**	Performance is very high and it provides result with high accuracy and precision.
NFR-5	**Availability**	With a good system, all authorized users can access and view the medical reports of patients.
NFR-6	**Scalability**	Performance will be good even with high user traffic.

5.PROJECT DESIGN:

5.1.DATAFLOW DIAGRAM:

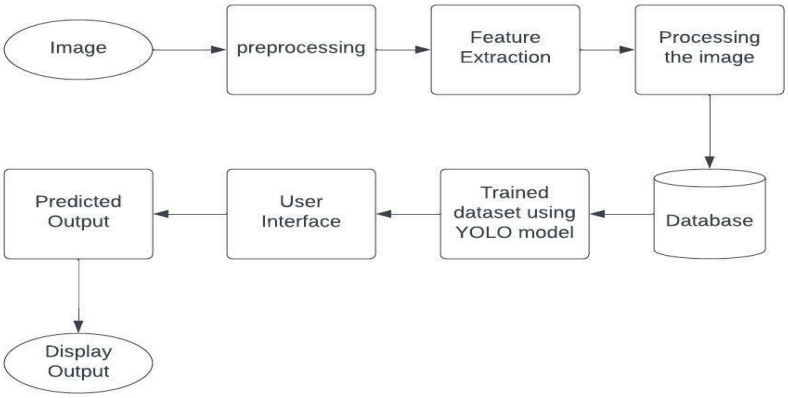

Fig.2.1.Dataflow Diagram

5.2.SOLUTION AND TECHNICAL ARCHITECTURE:

SOLUTION ARCHITECTURE:

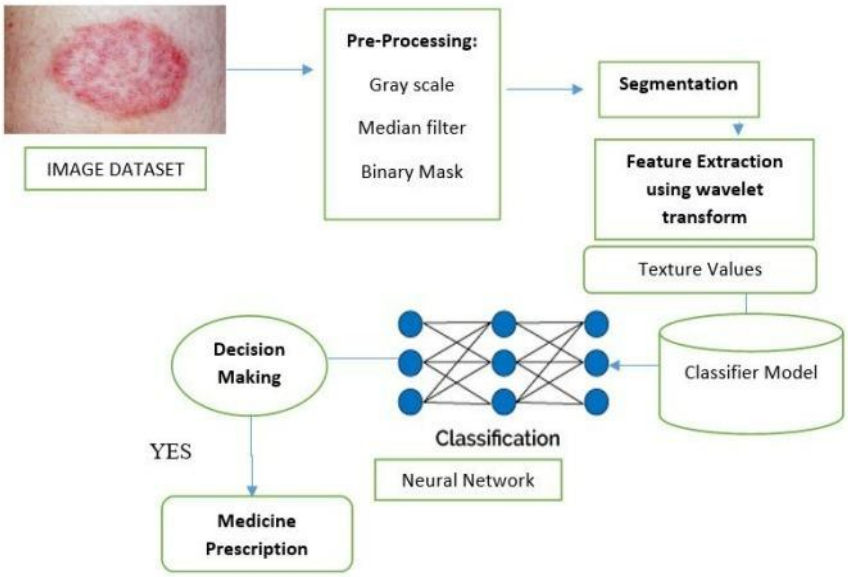

Fig.2.2.Solution Architecture

TECHNICAL ARCHITECTURE:

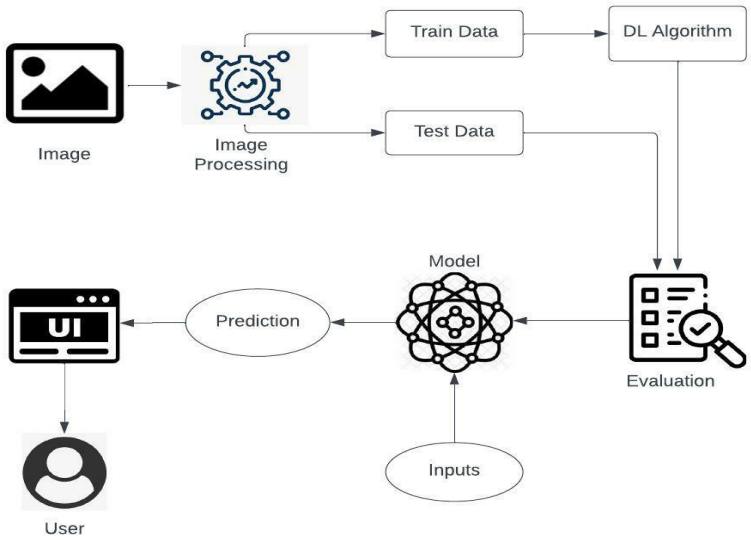

Fig.2.3.Technical Architecture

5.3.USER STORIES AND CUSTOMER JOURNEY MAP:

CUSTOMER JOURNEY MAP:

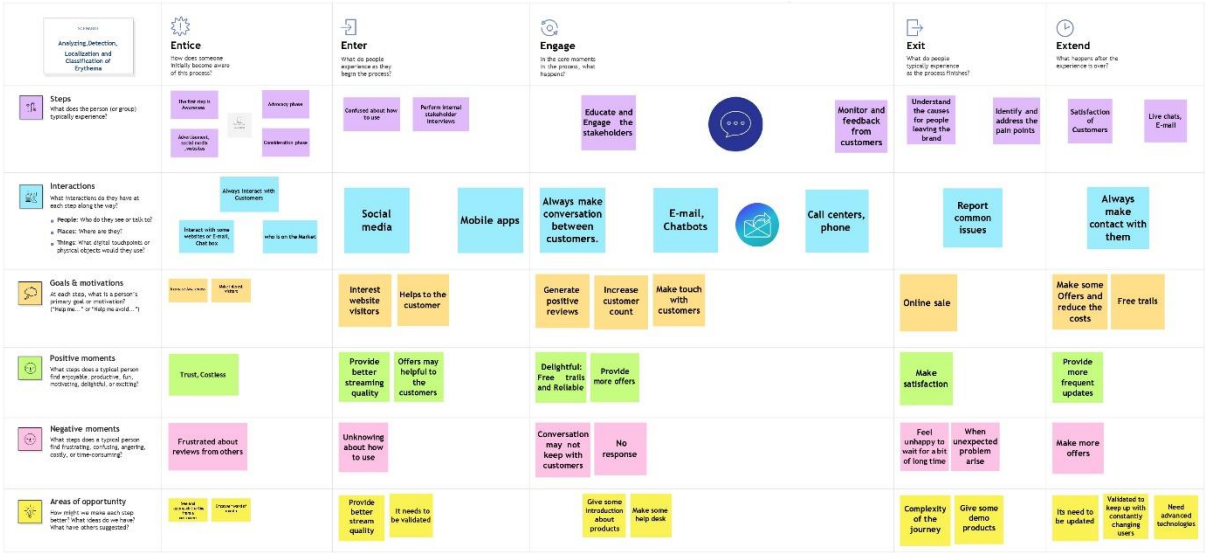

Fig.2.4.Customer Journey Map

USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Patient (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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm
		USN-3	As a user, I can register for the application through Gmail	I can receive confirmation email
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login anytime with my login credentials
	Dashboard	USN-5	As a user, I will be informed to fill my Details	I can view and edit my information
		USN-6	As a user, I will be informed to submit an Image	I will get the Analyzed results
		USN-7	As a user, I can login anytime anywhere with network connection	I will get the Analyzed results

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

1. Project objective

   Project abstract

2. Problem statement

3. Proposed solution

4. Required software installation

- PythonIDE
- IBMcloud
- IBMWASTONstudio
- IBMCLOUDANTDB
- YOLOv3algorithm

5. Technical Architecture

6. Building project

- Install python IDE-Spyder/Pycharm IDE
- Install python packages

  Tensorflow

  KerasOpenC

  VProgressBar

7. Install Microsoft's Visual Object Tagging Tool(VOTT)

Download and install the version of your operating system.

8. Training YOLO

Download and convert pre-trained weights.

9. CLOUDANTDB

Register and Login to IBM cloud

10. Create service instance

11. Launch CLOUDANTDB

12. Create database

Create Dataset from scratch

13. Ideation Phase

- Literature survey
- Empathy Map Canvas
- Ideation

14. Project design phase-I

- Proposed solution
- Problem solution fit
- Solution Architecture

15. Project design phase-II

- Customer Journey
- Functional Requirements
- Data flow diagram
- Technology Architecture

16. Project planning phase

- Prepare Milestones & Activitylist
- Sprint Delivery plan

17. Project development phase

- Projectdevelopment-deliveryofsprint-1
- Projectdevelopment-deliveryofsprint-2
- Projectdevelopment-deliveryofsprint-3
- Projectdevelopment-deliveryofsprint-4

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-1	Prerequisites	USN-1	Install Python IDE, Python packages, Microsoft Visual Object Tagging Tool, YoloStructure
Sprint-1	Data Collection	USN-2	Dataset should be collected from google or using a Chrome extension such as Fatkun Batch Downloader
Sprint-1	Annotate Images	USN-3	Create A Project in VOTT (Microsoft's VisualObject Tagging Tool)
Sprint-2	Training YOLO	USN-4	train our model using YOLO weights
Sprint-2		USN-5	To Download and Convert Pre-Trained Weights

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-2		USN-6	To Train YOLOv3 Detector
Sprint-3	Cloudant DB	USN-7	Register & Login to IBM Cloud
Sprint-3		USN-8	Create Service Instant and Credentials
Sprint-3		USN-9	Launch DB and Create database
Sprint-3	Development Phase	USN-10	To build a web application
Sprint-3		USN-11	Building HTML pages with python code
Sprint-3		USN-12	To run the application
Sprint-4	Testing Phase	USN-13	As a user login to dashboard
Sprint-4		USN-14	As a user import the images with skin diseases to the software application
Sprint-4		USN-15	YOLO processes the image and give the necessary details

7. CODING & SOLUTIONING

7.1 Microsoft's Visual Object Tagging Tool (VoTT):

It is an open source annotation and labeling tool for image and video assets.

VoTT is a React + Redux Web application, written in TypeScript.

Features include:

- The ability to label images or video frames
- Extensible model for importing data from local or cloud storage providers
- Extensible model for exporting labeled data to local or cloud storage providers

Using VoTT:

- Creating Connections
- Creating a New Project
- Project Settings
- Security Tokens
- Labeling an Image
- Labeling a Video
- Exporting Labels

7.2 YOLO Project Structure:

It was proposed by Joseph Redmond et al. in 2015. It was proposed to deal with the problems faced by the object recognition models at that time, Fast R-CNN is one of the state-of-the-art models at that time but it has its own challenges such as this network cannot be used in real-time, because it takes 2-3 seconds to predicts an image and therefore cannot be used in real-time. Whereas, in YOLO we have to look only once in the network i.e. only one forward pass is required through the network to make the final predictions.

Code:

```
from PIL import Image
from os import path, makedirs
import os
import re
import pandas as pd
import sys
import argparse
from Convert_Format import convert_vott_csv_to_yolo
def get_parent_dir(n=1):
    """ returns the n-th parent dicrectory 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
sys.path.append(os.path.join(get_parent_dir(1), "Utils"))
Data_Folder = os.path.join(get_parent_dir(1), "Data")
VoTT_Folder = os.path.join(
```

```
    Data_Folder, "Source_Images", "Training_Images", "vott-csv-export"
)
VoTT_csv = os.path.join(VoTT_Folder, "Annotations-export.csv")
YOLO_filename = os.path.join(VoTT_Folder, "data_train.txt")
model_folder = os.path.join(Data_Folder, "Model_Weights")
classes_filename = os.path.join(model_folder, "data_classes.txt")
if __name__ == "__main__":
    # surpress any inhereted default values
    parser = argparse.ArgumentParser(argument_default=argparse.SUPPRESS)
    parser.add_argument(
        "--VoTT_Folder",
        type=str,
        default=VoTT_Folder,
        help="Absolute path to the exported files from the image tagging step with
VoTT. Default is "
        + VoTT_Folder,
    )
    parser.add_argument(
        "--VoTT_csv",
        type=str,
        default=VoTT_csv,
        help="Absolute path to the *.csv file exported from VoTT. Default is "
        + VoTT_csv,
    )
    parser.add_argument(
        "--YOLO_filename",
        type=str,
        default=YOLO_filename,
```

```
        help="Absolute path to the file where the annotations in YOLO format should be saved. Default is "
        + YOLO_filename,
    )
    FLAGS = parser.parse_args()
    # Prepare the dataset for YOLO
    multi_df = pd.read_csv(FLAGS.VoTT_csv)
    labels = multi_df["label"].unique()
    labeldict = dict(zip(labels, range(len(labels))))
    multi_df.drop_duplicates(subset=None, keep="first", inplace=True)
    train_path = FLAGS.VoTT_Folder
    convert_vott_csv_to_yolo(
        multi_df, labeldict, path=train_path, target_name=FLAGS.YOLO_filename
    )
    # Make classes file
    file = open(classes_filename, "w")
    # Sort Dict by Values
    SortedLabelDict = sorted(labeldict.items(), key=lambda x: x[1])
    for elem in SortedLabelDict:
        file.write(elem[0] + "\n")
    file.close()
```

7.3. Database Schema:

A database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline.

A database schema is considered the "blueprint" of a database which describes how the data may relate to other tables or other data models. However, the schema does not actually contain data.

key benefits of database schemas include:

- Access and security: Database schema design helps organize data into separate entities, making it easier to share a single schema within another database.
- Organization and communication: Documentation of database schemas allow for more organization and better communication among internal stakeholders.
- Integrity: This organization and communication also helps to ensure data validity.

Code:

```
from cloundant.client import cloudant
client=Cloudant.iam('username','apikey','connect True')
my_database=client.create_database('my_database')
```

8. TESTING

8.1 User Acceptance Testing:

User acceptance testing, a testing methodology where the clients/end users involved in testing the product to validate the product against their requirements. It is performed at client location at developer's site.

For industry such as medicine or aviation industry, contract and regulatory compliance testing and operational acceptance testing is also carried out as part of user acceptance testing.

UAT is context dependent and the UAT plans are prepared based on the requirements and NOT mandatory to execute all kinds of user acceptance tests and even coordinated and contributed by testing team.

Acceptance criteria are defined on:

- Functional Correctness and Completeness
- Data Integrity
- Data Conversion
- Usability
- Performance
- Timeliness
- Confidentiality and Availability
- Installability and Upgradability
- Scalability
- Documentation

9. RESULTS

9.1 Performance Metrics:

the performance metrics used for evaluating a classification model:

- Accuracy - The overall accuracy of a model is simply the number of correct predictions divided by the total number of predictions.
- Precision and Recall - Precision measures how good the model is at correctly identifying the positive class. Recall tell us how good the model is at correctly predicting all the positive observations in the dataset.
- F1-score - The F1 score is the harmonic mean of precision and recall. The F1 score will give a number between 0 and 1.
- AUC-ROC - The AUC is the measurement of the entire two-dimensional area under the curve and The ROC (Receiver Operating Characteristics) curve is a plot of the performance of the model

10. ADVANTAGES & DISADVANTAGES

Merits:

- 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 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.

Demerits:

- An inherent disadvantage of clustering a skin disease is its lack of robustness against noise.
- Centroid that can generalize a cluster of data can significantly degrade the performance of these algorithms.
- the degradation problem that occurs when CNN models become too large and complex.
- Hence, we implement skip-connections in both segmentation and classification models.

11. CONCLUSION

The Project AI-Based Localization of Skin Disease With Erythema is used to find whether the person is having erythema or not. And our project helps lots of people to find whether their skin disease is erythema or not. Our website shows the accurate result so it helps the user to check their skin Disease. It is User Friendly Website.

12. FUTURE SCOPE

Future Scope of Our Project AI - Based Localization Of Skin Disease With Erythema is to try new algorithms and improve the accuracy of the result. And also developing a mobile application is our scope of the project.

13.APPENDIX

SOURCE CODE:

Prediction.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/d
AiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93
hXpG5KkN" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvf
a0b4Q" crossorigin="anonymous"></script>
```

```
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PV
CmYl" crossorigin="anonymous"></script>

        <script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>

    <link
href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&dis
play=swap" rel="stylesheet">

    <link rel="stylesheet" href="style1.css">

        <script defer src="JScript.js"></script>

    <title>Prediction</title>

</head>

<body>

    <header id="head" class="header">

        <section  id="navbar">

                <h1 class="nav-heading"></i>Skin Disease Detection</h1>

            <div class="nav--items">

                <ul>

                    <li><a  href="index.html">Home</a></li>

                                    <li><a href="logout.html">Logout</a></li>

                <!-- <li><a href="#about">About</a></li>
```

```
            <li><a href="#services">Services</a></li> -->
                        </ul>
          </div>
       </section>
    </header>
    <!-- dataset/Training/metal/metal326.jpg -->
       </br>
    <section id="prediction">
    <h2 class="title text-muted">AI-based localization and classification of skin disease with erythema</h1>
    <div class="line" style="width: 1000px;"></div>
                </section>
                </br>
         <section id="about">
    <div class="body">
<div class="left">
    <p>                 </p>
</div>
<div class="left">
        <div class="prediction-input">
        <img class="d-block" src="" alt="" style="width:80%!important; padding-left:100px">
```

```
    </br>
        <form id="form" action="/result" method="post"
enctype="multipart/form-data">
            <!-- <img id="img" src="/static/uploaded_images/aas_disease.png"
alt="" width="300" height="200">
            <input type="submit" class="submitbtn" value="Click here for
Prediction"> -->
            <input type="file" id="image-input" accept="image/jpeg,
image/png, image/jpg">
            <div id="display-image"></div>
            <div>
              <bold><p id="predict_img"></p></bold>
              </p>
              <p id="predict_img1">
              </p>
            </div>
          </form>
        </div>
        <h5 style="color:Red">
        <b style="color:Red" /b>
      </h5>
</div>
</div>
```

```
</section>
    <section id="footer">
    </section>
</body>
<script>
    const image_input = document.querySelector("#image-input");
image_input.addEventListener("change", function() {
  const reader = new FileReader();
  reader.addEventListener("load", () => {
    const uploaded_image = reader.result;
    document.querySelector("#display-image").style.backgroundImage =
`url(${uploaded_image})`;
  });
  reader.readAsDataURL(this.files[0]);
});
</script>
</html>
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/d
AiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93
hXpG5KkN" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvf
a0b4Q" crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
```

```
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PV
CmYl" crossorigin="anonymous"></script>

    <script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>

    <link
href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&dis
play=swap" rel="stylesheet">

    <link rel="stylesheet" href="style1.css">

    <!-- <script defer src="../static/js/main.js"></script> -->

    <title>AI-based localization and classification of skin disease with
erythema</title>

</head>

<body>

    <header id="head" class="header">

   <section  id="navbar">

          <h1 class="nav-heading">AI-based localization and classification of skin
disease with erythema</h1>

       <div class="nav--items">

          <ul>

                                <li><a  href="index.html">Home</a></li>

                                <li><a href="login.html">Login</a></li>

                                <li><a href="register.html">Register</a></li>

             <!-- <li><a href="#about">About</a></li>
```

```
            <li><a href="#services">Services</a></li> -->
            <li><a href="prediction.html">Prediction</a></li>
          </ul>
        </div>
    </section>
    <section id="slider">
     <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
        <ol class="carousel-indicators ">
            <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active "></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
        </ol>
        <div class="carousel-inner">
                <div class="carousel-item active">
             <img class="d-block w-100"
src="https://storage.googleapis.com/gweb-uniblog-publish-prod/images/derm_-
_hero_image_2_1.max-1000x1000.jpg" alt="First slide" style="height:500px">
            </div
        <a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
```

```
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>
  </section>
</header>
<section id="about">
    <div class="top">
        <h3 class="title text-muted">
        </h3>
        <div class="line"></div>
    </div>
<div class="body">
<div class="left">
    <h2>Problem</h2>
    <p>


                </p>
</div>
```

```
<div class="right">
    <h2>Solution</h2>
    <p>
    </p>
</div>
</div>
</section>
<section id="services">
<h3 class="title text-muted">WE CLASSIFY</h3>
<div class="line"></div>
<div class="testimonials">
    <div class="card" style="width: 25rem;">
        <img
src="https://upload.wikimedia.org/wikipedia/commons/6/69/Erythema_multiforme_minor_of_the_hand.jpg" class="card-img-top" alt="Erythema multiforme (EM)" style="height: 300px">
        <div class="card-body text-muted">
          <h5 class="card-title text-muted">Erythema multiforme (EM)</h5>
        </div>
      </div>
    <div class="card" style="width: 25rem;">
        <img
src="https://i.pinimg.com/736x/7e/ec/2e/7eec2ee1c2e592aa94dac11036fad68d-
```

```
-lyme-disease-nursing.jpg" class="card-img-top" alt="Erythema chronicum migrans" style="height:300px">
      <div class="card-body text-muted">
                  <h5 class="card-title text-muted">Erythema chronicum migrans</h5>
       </div>
     </div>
   <div class="card" style="width: 25rem;">
      <img src="https://phil.cdc.gov//PHIL_Images/9875/9875_lores.jpg" class="card-img-top" alt="Erythema migrans" style="height:300px">
      <div class="card-body">
        <h5 class="card-title text-muted">Erythema migrans</h5>
      </div>
      <div class="card" style="width: 25rem;">
         <img src="https://www.researchgate.net/publication/304555872/figure/fig1/AS:379660283203584@1467529473118/Erythema-marginatum-on-the-arm-of-a-patient-with-hereditary-angioedema-Note-that-lesions.png" class="card-img-top" alt="Erythema marginatum" style="height:300px">
         <div class="card-body">
          <h5 class="card-title text-muted">Erythema marginatum</h5>
         </div>
      <div class="card" style="width: 25rem;">
```

```
            <img
src="https://images.medicinenet.com/images/image_collection/skin/erythema-
infectiosum.jpg" class="card-img-top" alt="Erythema infectiosum"
style="height:300px">
            <div class="card-body">
                <h5 class="card-title text-muted">Erythema infectiosum</h5>
              </div>
       <div class="card" style="width: 25rem;">
            <img
src="https://media.healthdirect.org.au/images/inline/original/erythma_nodosum
_a4gprc-ccee8a.jpg" class="card-img-top" alt="Erythema nodosum"
style="height:300px">
            <div class="card-body">
              <h5 class="card-title text-muted">Erythema nodosum</h5>
          </div>
     </div>
   </div>
</section>
<section id="footer">
   <div class="social">
     <a href="#" target="_blank"><i class="fab fa-2x fa-twitter-
square"></i></a>
      <a href="#" target="_blank">
```

```
        <i class="fab fa-2x  fa-linkedin"></i></a>
        <a href="#">
            <i class="#"></i>
        </a>
    </div>
</section>
</body>
</html>
```

import.py

```
import re
import numpy as np
import os
from flask import Flask, app,request,render_template
import sys
from flask import Flask, request, render_template, redirect, url_for
import argparse
from tensorflow import keras
```

```
from PIL import Image
from timeit import default_timer as timer
import test
from pyngrok import ngrok
import pandas as pd
import numpy as np
import random
def get_parent_dir(n=1):
    """ returns the n-th parent dicrectory 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'/content/drive/MyDrive/IBM_PROJECT/yolo_structure/2_Training/
src'
print(src_path)
utils_path=r'/content/drive/MyDrive/IBM_PROJECT/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,
detect_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), "yolo_structure", "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

from cloudant.client import Cloudant

# Authenticate using an IAM API key

client = Cloudant.iam('bb9ac486-ac17-48ff-9bd8-7df213ab022b-
bluemix','MCWIFbc76YNRr6g5ZXW8Y04it5gIw6mxPd3X_4BejxvO',
connect=True)

# Create a database using an initialized client

my_database = client.create_database('my_database')

app=Flask(__name__)

port_no=5000

ngrok.set_auth_token("2H7aM94zEuTa40t3J6jKpIqWAc3_B2UxzZs6qxetntga
dxQW")

public_url =  ngrok.connect(port_no).public_url

print(f"To acces the Gloable link please click {public_url}")


#default home page or route
```

```
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/index.html')
def home():
    return render_template("index.html")
#registration page
@app.route('/register')
def register():
    return render_template('register.html')


@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
    '_id': x[1], # Setting _id is optional
    'name': x[0],
    'psw':x[2]
    }
    print(data)
```

```
    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        url = my_database.create_document(data)
        #response = requests.get(url)
        return render_template('register.html', pred="Registration Successful, please login using your details")
    else:
        return render_template('register.html', pred="You are already a member, please login using your details")

#login page
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST'])
```

```
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        return render_template('login.html', pred="The username is not found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')


@app.route('/logout')
```

```
def logout():
    return render_template('logout.html')


@app.route('/prediction')
def prediction():
    return render_template('prediction.html',path="../static/img/6623.jpg",)



@app.route('/result',methods=["GET","POST"])
def res():
    # Delete all default flags
    parser = argparse.ArgumentParser(argument_default=argparse.SUPPRESS)
    """
    Command line options
    """
    f = request.files['file']
    f.save("./drive/MyDrive/IBM_PROJECT/Flask/static/img/"+f.filename)




    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 images
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(
        "--gpu_num", type=int, default=1, help="Number of GPU to use. Default
is 1"
    )
    parser.add_argument(
        "--confidence",
        type=float,
        dest="score",
        default=0.25,
```

```
        help="Threshold for YOLO object confidence score to show predictions. Default is 0.25.",
    )
    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"',
    )
    yolo = YOLO(
        **{
            "model_path": FLAGS.model_path,
```

```
        "anchors_path": FLAGS.anchors_path,
        "classes_path": FLAGS.classes_path,
        "score": FLAGS.score,
        "gpu_num": FLAGS.gpu_num,
        "model_image_size": (416, 416),
    }
)
img_path="/drive/MyDrive/IBM_PROJECT/Flask/static/img/"+f.filename
prediction, image,lat,lon= detect_object(
        yolo,
        img_path,
        save_img=save_img,
        save_img_path=FLAGS.output,
        postfix=FLAGS.postfix,
)
yolo.close_session()
return
render_template('prediction.html',prediction=str(prediction),path="../static/img/
"+f.filename)
""" Running our application """
if __name__ == "__main__":
    app.run(port=port_no)
```

PROJECT DEMO LINK:

https://divyaslp14zn.hippovideo.io/video/play/2AiwJFgjPmn5xH_5lAMffWfgJdKJ_JO25ZO2Tv_LEDY

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-18710-1659688879