Phase 1: Problem Definition and Design Thinking

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem. Please think on a design and present in form of a document.

Problem Definition:

[to be explained in detail]

Design Thinking:

[In this part you will need to give all steps how will you proceed ahead with solving the problem.]

NOTE:

File Naming Convention: **AI_Phase1**

After completion upload your file to your private GitHub account. Please give access to your faculty evaluators of your college and industry evaluator [IndustryEvaluator@skillup.online] to your private GitHub repository for evaluation process

Go to the Project Submission Part 1 section and add your college code, the link of your GitHub in the space provided, upload your documents, and click on submit.

Phase 2: Innovation

In this phase you need to put your design into innovation to solve the problem.

Explain in detail the complete steps that will be taken by you to put your design that you thought of in previous phase into transformation.

Create a document around it and share the same for assessment.

NOTE:

File Naming Convention: **TechnologyName_Phase2**

After completion upload your file to your private GitHub account. Please give access to your faculty evaluators[**facultyevaluator@gmail.com**] and industry evaluator [**IndustryEvaluator@skillup.online**] to your private GitHub repository for evaluation process.

SKIN DISEASE PREDICTION USING CNN

SOCIALLY RELEVANT PROJECT

Submitted by

ABARNA E (411620243001)

HARSITHA KANWER V (411620243010)

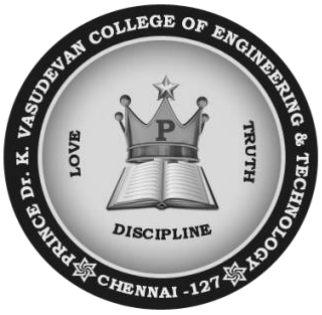
In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE



PRINCE Dr K VASUDEVAN COLLEGE OF ENGINEERING AND TECHNLOGY, PONMAR, CHENNAI-600 127

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2022

BONAFIDE CERTIFICATE

Certified that this project report **"MOVIE AND MUSIC RECOMMENDATION SYSTEM"** is Bonafide of **ABARNA E (411620243001)** and **HARSITHA KANWER V (411620243010)** who carried out the project work under my supervision.

SIGNATURE

MRS.M.BIRUNTHA, M.E.,

HEAD OF DEPARTMENT

(incharge)

Department Of CSE,

Prince Dr. K. Vasudevan College

Of Engineering and Technology,

Chennai – 600127

SIGNATURE

MRS.R.DEEPA,M.E.,

SUPERVISOR

Department of AI & DS,

Prince Dr . K. Vasudevan

College Of Engineering and

Technology,

Chennai - 600127

Submitted to the Viva voce Examination held on _____

INTERNAL EXAMINER **EXTERNALEXAMINER**

ACKNOWLEDGEMENT

We wish to express our sincere thanks to our Founder and Chairman, **Dr. K. VASUDEVAN, M.A., B.Ed., Ph.D.,** for his endeavour in educating us in his premier institution

We would like to extend our heartfelt gratitude and sincere thanks to our Vice-Chairman, **Dr. V. VISHNU KARTHIK, M.D.,** for his keen interest in our studies and the facilities offered in this premier institution

We would like to express our deep gratitude and sincere thanks to our Administrative Officer, **Dr. K. PARTHASARATHY BE.,** for his valuable support.

We wish to express our sincere thanks to our Honourable Principal, **Dr. T. SUNDER SELWYN, M.E., Ph.D.,** for permitting to access various resources in the college to complete the project work.

We also wish to convey our thanks and regard to our HOD ,

Mrs. M.BIRUNTHA, M.E., Department of Artificial Intelligence and Data Science , for her guidance and support throughout our project.

We wish to express our great deal of gratitude to our Project In charge **Mrs. R.DEEPA, M.E.,** Department of Artificial Intelligence and Data Science for the completing of our project successfully.

We would like to extend our thanks to all teaching and non-teaching staffs of Department of Artificial Intelligence and Data Science for their continuous support.

ABSTRACT

- Skin diseases are more common nowadays they are usually caused by fungal infection, bacteria, allergy or viruses etc,.. So to diagnose this, image processing technique is used . So by using this technique the digital image of disease effect skin area is taken then by analysing the image, the disease type is identified. From the image the features are extracted using Convolutional Neural Network after the features are classified, multiclass SVM is used and then finally results are shown to the user, including the type of disease and the ayurvedic medicines and solution to that particular disease. The diagnosis of the skin disease requires a high level of expertise and accuracy for dermatologist, so computer aided skin disease diagnosis model is proposed to provide more objective and reliable solution. Many researches were done to help detect skin diseases like skin cancer and tumor skin. But the accurate recognition of the disease is extremely challenging due to the following reasons: low contrast between lesions and skin, visual similarity between Disease and non-Disease area, etc. This paper aims to detect skin disease from the skin image and to analyze this image by applying filter to remove noise or unwanted things, convert the image to greyto help in the processing and get the useful information. This help to give evidence for any type of skin disease and illustrate emergency orientation. Analysis result of this study can support doctor to help in initial diagnoses and to know the type of disease. That is compatible with skin and to avoid side effects

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACRONYMS	DESCRIPTION
CRNN	Convolutional Recurrent Neural Network
DBSCAN	Density Based Spatial Clustering

CHAPTER 1

INTRODUCTION

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 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 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. The advancement of lasers and photonics based medical technology has made it possible to diagnose the skin diseases much more quickly and accurately. But the cost of such diagnosis is still limited and very expensive. Therefore, we propose an image processing-based approach to diagnose the skin diseases. This method takes the digital image of disease effect skin area then use image analysis to identify the type of disease. Our proposed approach is simple, fast and does not require expensive equipment's other than a camera and a computer.

One of the medical areas which needs mobile health technology is skin analysis to identify diseases. Around 24% of the population in England and Wales (12.9 million people) visited their general practitioner with a skin problem in 2006, with the most common reasons being skin infection and eczema (WHO). One of the most preventable types of cancer is the skin cancer and the best ways to keep the skin healthy and cancer free is checking and examining skin once month for suspicious moles or spots.

This exam is a visual and clinical skin exam, it costs more than Rs.5000 and it needs hospital scan which is difficult for disabled people. In our paper, we propose a low cost Smartphone based intelligent scheme that allows people for regular skin examinations. We use a Smartphone camera and an intelligent learning algorithm to scan skin images.

CHAPTER 2

LITERATURE SURVEY

Krizhevsky et al. won the classification challenge of the ImageNet large scale visual recognition challenge 2012, Deep Neural Networks (DNN) reached to the top in computer vision (Krizhevsky et al., 2012). As Deep Learning methods have shown promising solutions for many applications such as natural language processing, speech and facial recognition, object detection and image classification at this time, there is a trend to use Deep Learning methods for high accuracy medical image classifications. Some researchers started to use Convolutional Neural Networks (CNNs) for melanoma classification due to its higher discriminating capability in recognition. CNN is a type of Deep Learning method, where trainable filters are applied on raw data to extract selectable features automatically from complex systems (Nasr-Esfahani et al., 2016). In one of the research studies, a CNN classifier was trained by large number of trained clinical skin images to distinguish melanoma from benign cases reaching 81% of sensitivity and 80% of specificity (Nasr-Esfahani et al., 2016).

Chakraborty et al have proposed a hybrid model using multi objective optimization algorithm NSGA - II and ANN for diagnosis of for diagnosis of skin decision brewage being benign or malignant. Gessert et al introduced patch based method to obtain fine-grain differences between various skin lesions from high resolution images. Kulhalli et al has proposed a 5-stage, 3-stage and 2 stage hierarchical approach to 7 discases using Inception 1/3 CNN architecture.

Rehman et at have proposed CNN as chitecture by setting 16 different filters of 7+7 kernel size with pooling layers for down sampling chatterjee et al proposed spatial and frequency. domain based technique for identification of skins lesion being benign or me malignant.

The proposed solution in this papes is a prototype with a database of some commo skin diseases, using which patient can self-diagnose and get some prior knowledg

of before consulting the desmatologist. This prototype. can be used Most in mobile hospitals in rural areas. Thus, this prototype can be accessed even in remote areas of the country

The proposed prototype provides. an easiest and convenient method of skin disease. detection where the provi patient provides a picture of the infected area as an input prototype and the further analysis is done CNN Techniques on this input in a golutions for the disease and the ayurvedic given. No pricking of the skin is required.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

3.1.1. ARTIFICIAL NEURAL NETWORK(ANN).

An artificial neuron network (ANN) is a statistical nonlinear predictive modelling method which is used to learn the complex relationships between input and output. The structure of ANN is inspired by the biological pattern of our brain neuron [2]. An ANN has three types of computation node. ANNs learn computation at each node through back-propagation. There are two sorts of data set trained and untrained data set which produces the accuracy by employing a supervised and unsupervised learning approach with different sort of neural network architectures like feed forward, back propagation method which uses the info set at a special manner. Using Artificial Neural Network, accuracy obtained in various researches is jni 80% which isnt optimum [2]. Also, ANNs require processors with parallel processing power. ANN produces a probing solution it does not give a clue as to why and how it takes place which reduces trust in the network

3.1.2. BACK PROPAGATION NETWORK(BPN).

Back propagation, a strategy in Artificial Neural Networks to figure out the error contribution of each neuron after a cluster of information (in image recognition, multiple images) is processed. Back Propagation is quite sensitive to noisy and uproarious data. The BNN classifier achieves 75%-80% accuracy [2]. BNN is benefits on prediction and classification but the processing speed is slower compared to other learning algorithms [5] [2].

3.2 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. Economic feasibility
2. Technical feasibility
3. Social feasibility
4. Operational feasibility

3.2.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project’s potential for success.

3.2.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3 PROPOSED SYSTEM

The proposed solution in this paper is a prototype with a database of some common skin diseases, using which patient can self -diagnose and get some prior knowledge of their skin disease before consulting the dermatologist. This prototype can be used in mobile hospitals in rural areas. Thus, this prototype can be accessed even in most remote areas of the country.

The proposed prototype provides an easiest and convenient method of skin disease detection where the patient provides a picture of the infected area as an input prototype and the further analysis is done on this input image using CNN techniques and the ayurvedic solutions for the disease is given. No pricking of the skin is required.

The Proposed methodology is an effective tool which can analyze the people input skin disease to predict skin disease. In this proposed system, hybrid architecture with image processing and machine learning techniques are used to predict type of disease with promising accuracy in a short period of time. The image processing phase invokes preprocessing, segmentation, feature extraction steps.

The machine learning phase invokes 3 steps: processing, training and detection steps. The proposed system uses 2D Wavelet Transform algorithm for feature extraction in which color, texture and shape features are extracted from the skin input images. The correlation values are also been extracted from the input image. These values are passed onto classifier model.

For classification the proposed system uses convolutional neural network (CNN). The classifier model detects common skin diseases like Psoriasis, Lichen Planus, Pityriasis Rosea. Integration of neural network provides good accuracy results. The proposed system act as a Common knowledge base for skin disease detection and medicinal prescription. This proposed system analyses different type of skin disease

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

System Architecture is a generic discipline to handle objects (existing or to be created) called “systems”, in a way that supports reasoning about the structural properties of these objects. The system architecture is a response to the conceptual and practical difficulties of the description and the design of complex systems.

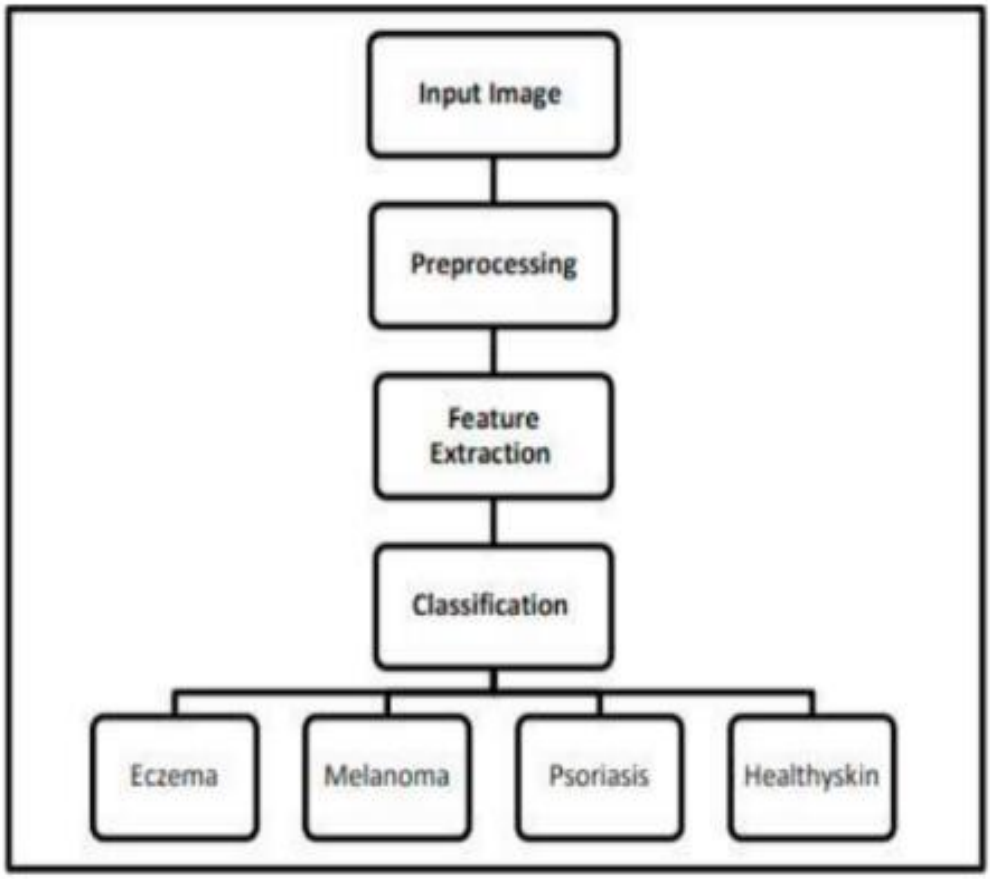


Fig:-4.1 Architecture

4.1.1. PREPROCESSING

Achieving high performance of skin disease detection system requires overcoming some major difficulties. Such as creating a database

and unifying image dimensions. In the following section, the technique used in image resizing is explained.

4.1.1.1 IMAGE RESIZING

To resolve the problem of different image sizes in the database an input image is either increase or decrease in size. Unifying the image size will get the same number of features from all images. Moreover, resizing the image reduces processing time and thus increases system performance. Fig 3 shows the original image of size is 260×325 pixels. Fig 4 shows the resized image with the new size of 227×227 pixels.

4.1.2. FEATURE EXTRACTION

At the beginning, Convolutional Neural Network (CNN) is a set of stacked layers involving both nonlinear and linear processes. These layers are learned in a joint manner. The main building blocks of any CNN model are: convolutional layer, pooling layer, nonlinear Rectified Linear Units (ReLU) layer connected to a regular multilayer neural network called fully connected layer, and a loss layer at the backend. CNN has known for its significant performance in applications as the visual tasks and natural language processing .

AlexNet is a deep CNN model, developed by Krizhevsky et al. [8], to model the 2012 ImageNet for the Large Scale Visual Recognition Challenge (ILSVRC-2012). AlexNet consists of five convolutional layers; where a nonlinear ReLU layer is stacked after each convolutional layer. In addition, the first, second, and fifth layers contain maxpooling layers, as shown in Figure 5. Moreover, two normalization layers are stacked after the first and the second convolutional layers. Furthermore, two fully connected layers at the top of the model preceded by softmax layer. AlexNet was trained using more than 1.2 million images belonging to 1000

classes . We proposed feature extraction from a pretrained convolutional neural network. Because it is the easiest and robust approach to use the power of pretrained deep learning networks.

4.1.3. CLASSIFICATION:

Classification is a computer vision method. After extracting features, the role of classification is to classy the image via Support Vector Machine (SVM). A SVM can train classifier using extracted features from the training set.

4.2 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process.

The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. 7. Integrate best practices.

4.2.1. USE-CASE DIAGRAM

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs. Because it is a very powerful planning instrument, the use-case model is generally used in all phases of the development cycle by all team members.

An effective use case diagram can help your team discuss and represent:

1. Scenarios in which your system or application interacts with people, organizations, or external systems.

2. Goals that your system or application helps those entities (known as actors) achieve.

3. The scope of your system.

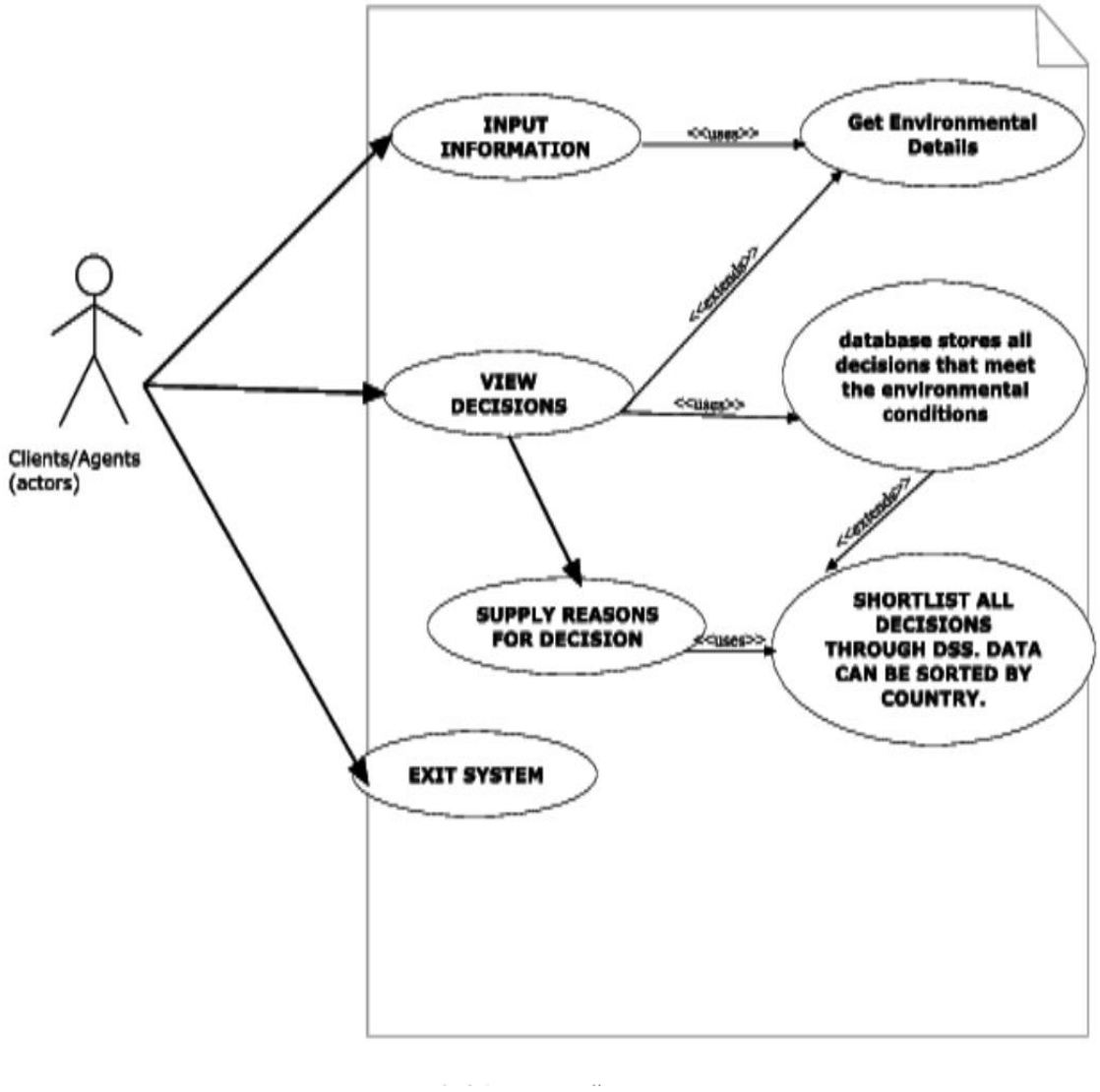


Fig: 4.2.1.2 Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations.

Try drawing a sequence diagram to:

1.Represent the details of a UML use case.

2.Model the logic of a sophisticated procedure, function, or operation.

3.See how objects and components interact with each other to complete a process.

4.Plan and understand the detailed functionality of an existing or future scenario.

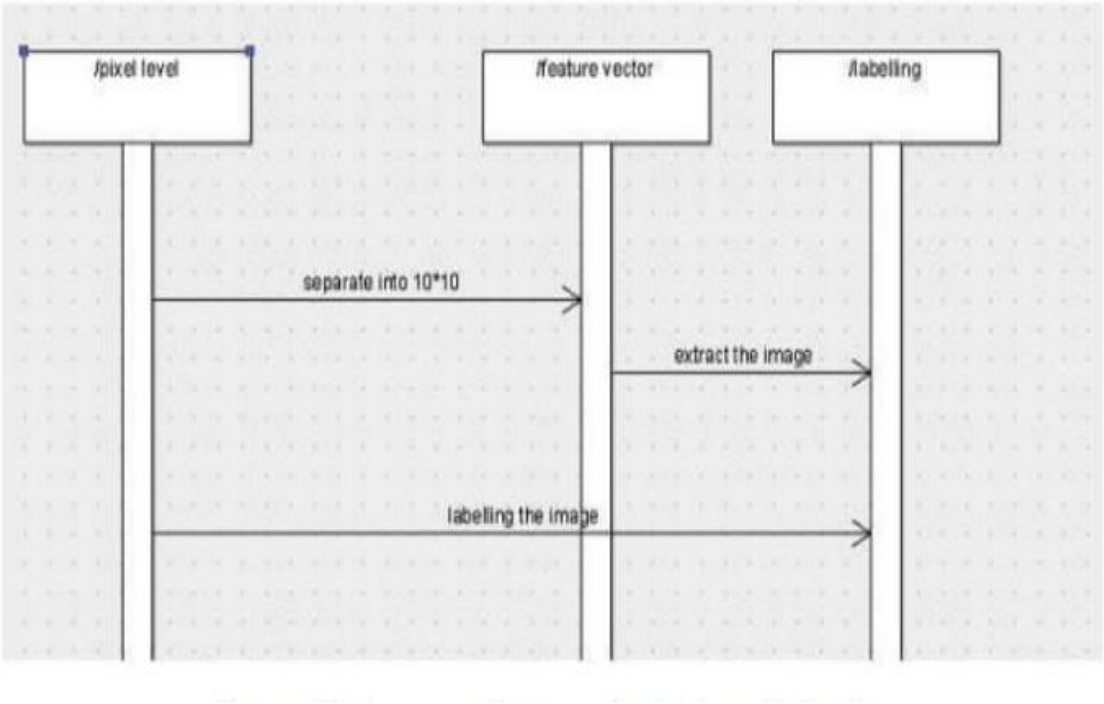


Fig: 4.2.2.2 Sequence Diagram For Feature Extraction

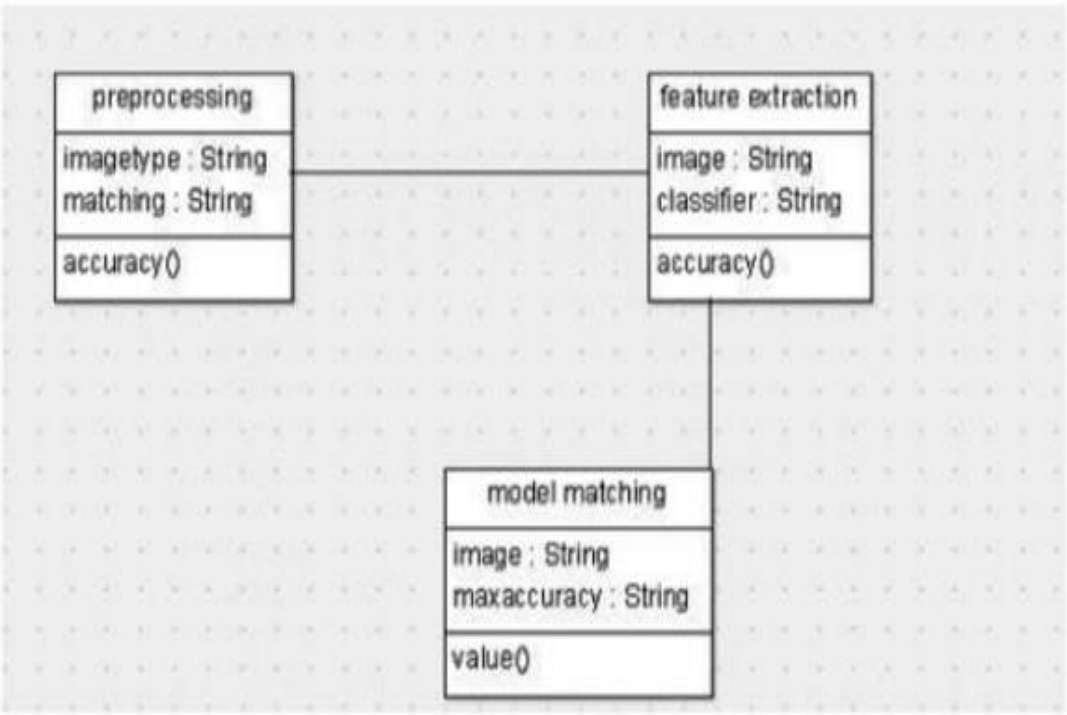


Fig: 4.2.2.2 Sequence Diagram For Trainer Module

4.2.3 ACTIVITY DIAGRAM

An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. It describes the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). The initiator is generally a function module that gets called when an activity starts.

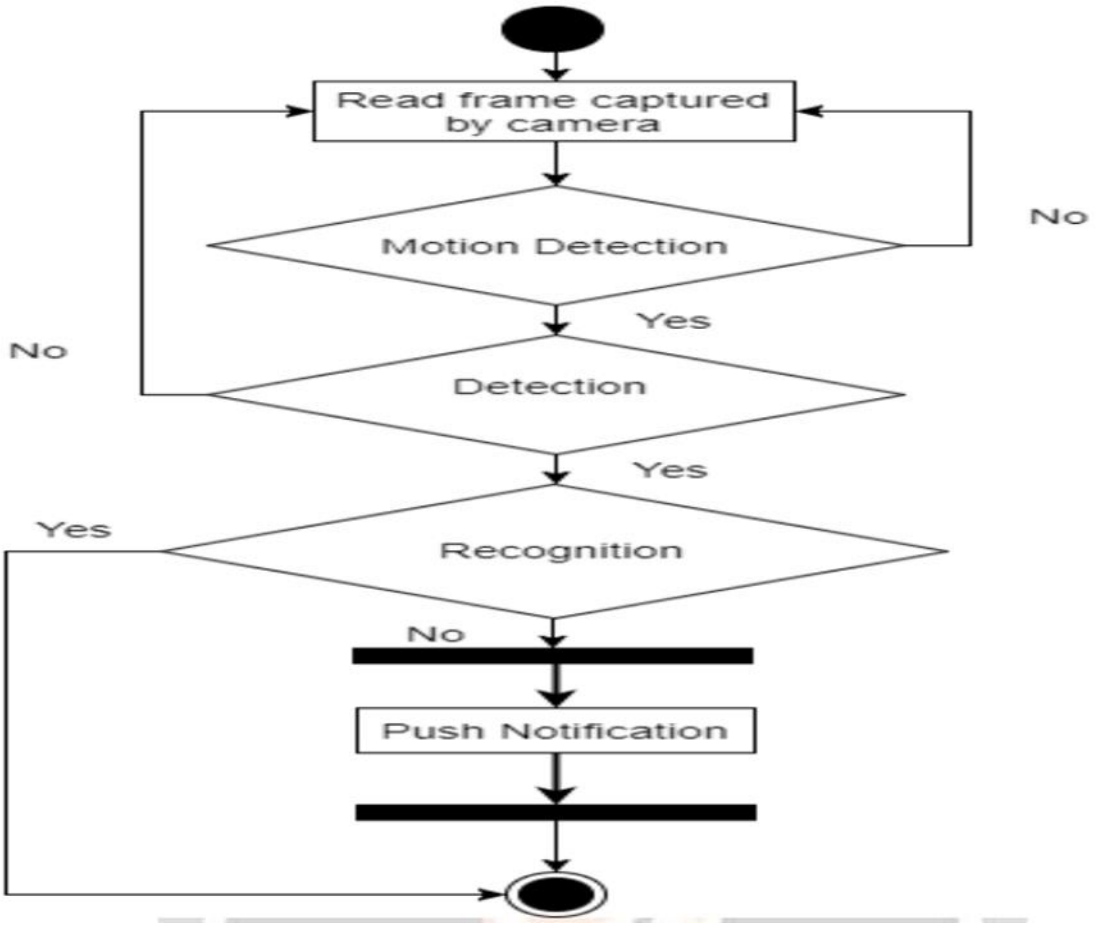


Fig: 4.2.3.1 Activity Diagram

4.2.4 DATA FLOW DIAGRAM

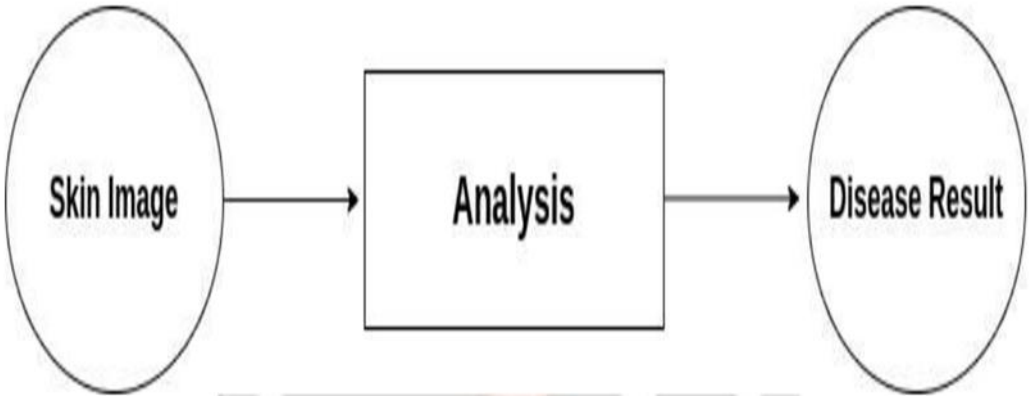


Fig:- 4.2.4.1 Data Flow Diagram For Level 1

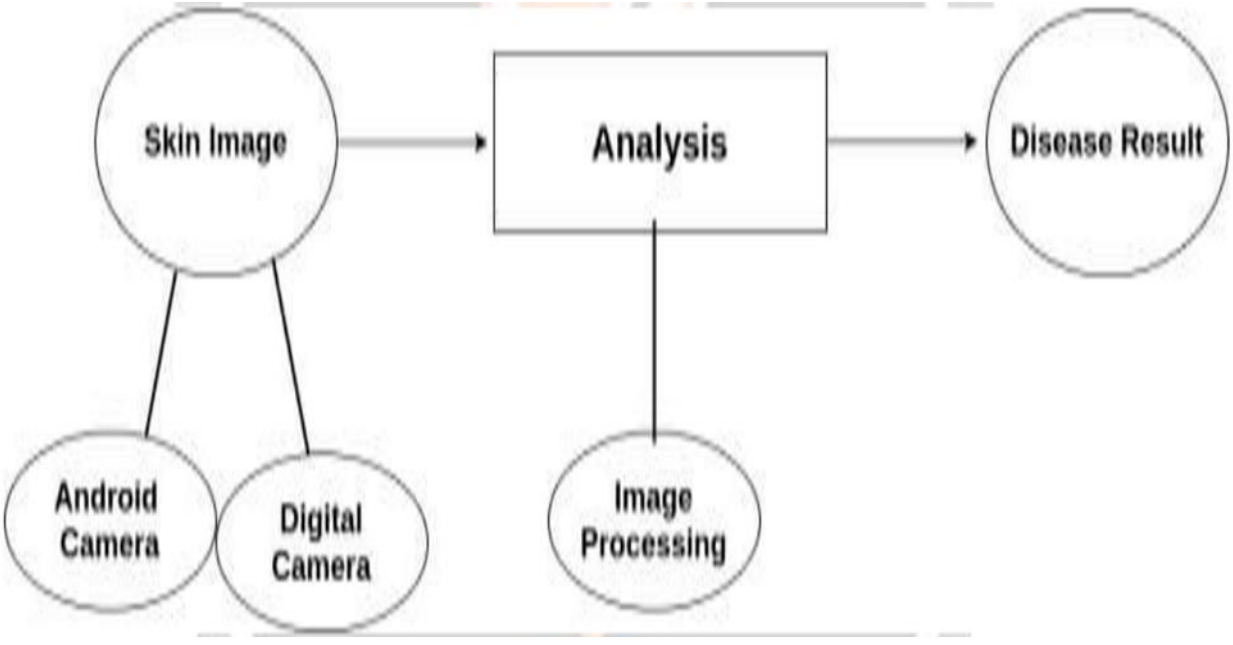


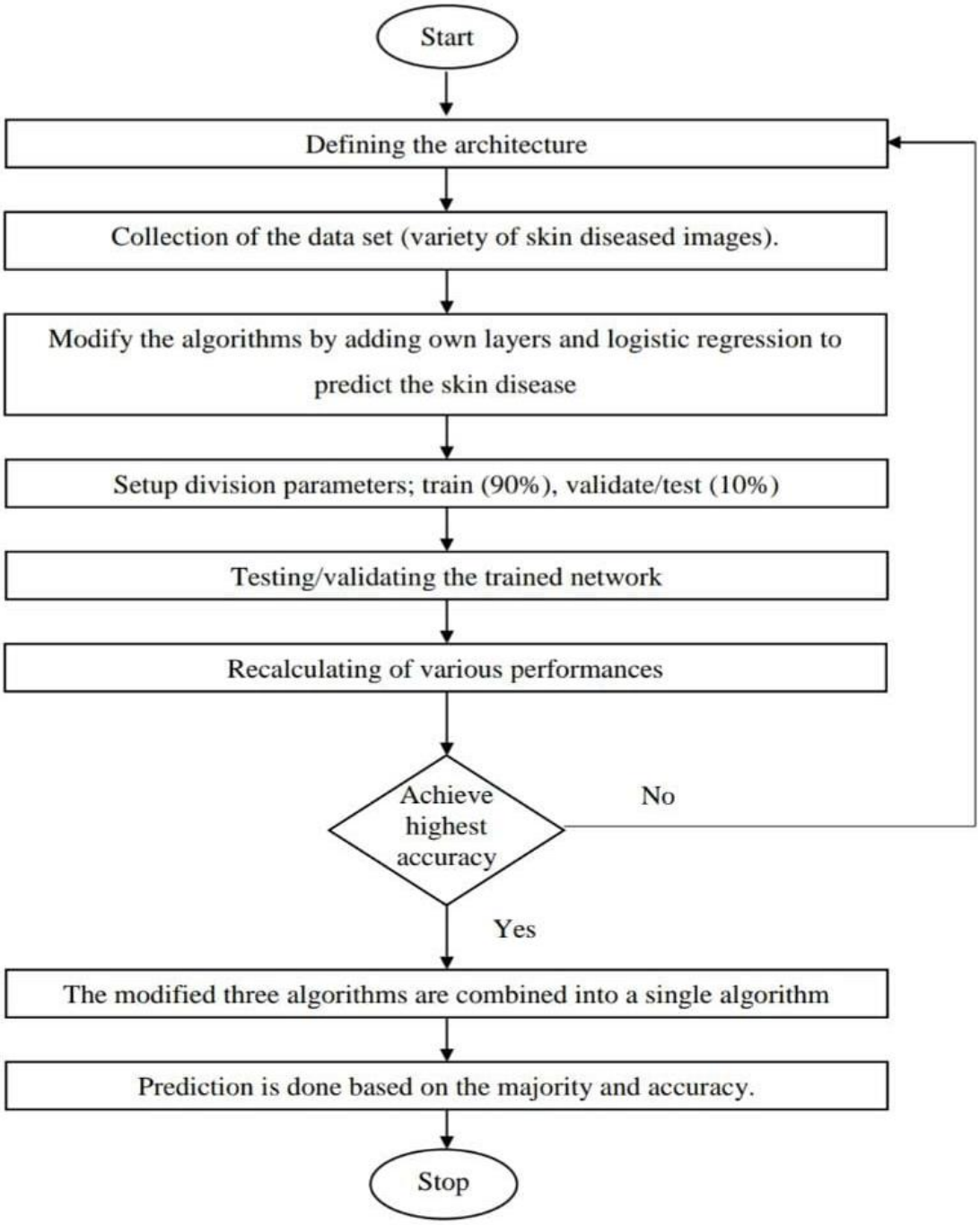
Fig:- 4.2.4.2 Data Flow Diagram For Level 1

Initially load the data sets that are required to build a model the data set that are required in this project are movies.csv, ratinfg.csv, users.csv all the data sets are available in the Kaggle.com. Basically, two models are built in this project content based and collaborative filtering each produce a list of movies to a particular user by combining both based on the use id a single final list of movies are recommended to the particular user.

CHAPTER 5

METHODOLOGY

Development of a widespread plan to test the special features and general functionality on a range of platform combination is firstly initiated by the test process. The procedures used are strictly quality control.



5.1: Framework of the methodology used

The process verifies that the application is bug free and it meets the requirements stated in the requirements document of system [18]. The following are the considerations used to develop the framework from developing the testing methodologies

5.1 Sample Data

In this study, the sample data employed to train the system model is presented in [Figure 3.14 and 3.15]. The database is split into; training set, validating/testing set. A training set is adopted for learning to fit the parameters and is specifically applied to alter the varying weights and errors of the system in each training run. Validation/testing set tunes the parameters and is used only to assess the effectiveness and efficiency of the system.

5.1.1 Image From The Training Data

5.1.2 Image From The Testing Data

CHAPTER 5

SYSTEM REQUIREMENTS

This chapter involves both the hardware and software requirements needed for the project and detailed explanation of the specifications.

5.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor with 1.7-2.4gHz speed
- Minimum of 8gb RAM
- 2gb Graphic card

5.2 Software Specification

- Text Editor (VS-code/WebStorm)
- Anaconda distribution package (PyCharm Editor)
- Python libraries

5.3 Software Requirements

5.3.1 Anaconda distribution:

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

5.3.2 Python libraries:

For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SKlearn, Numpy, pandas, Matplotlib, Flask framework, etc are needed.

SKlearn: It features various classification, regression and clustering

algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 SOFTWARE DESCRIPTION

6.1.1 Python Language

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Pythons are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code

- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

6.1.2 Anaconda

Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.

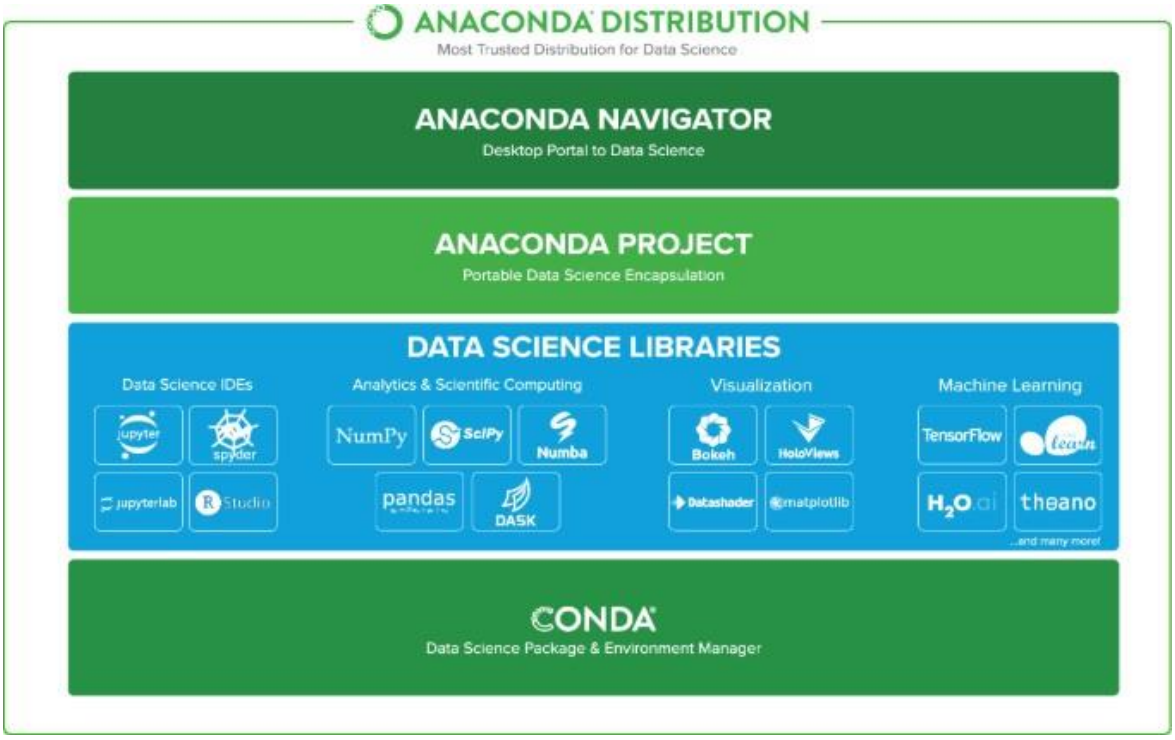


Fig: 6.1.2.1 Anaconda Distribution

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.

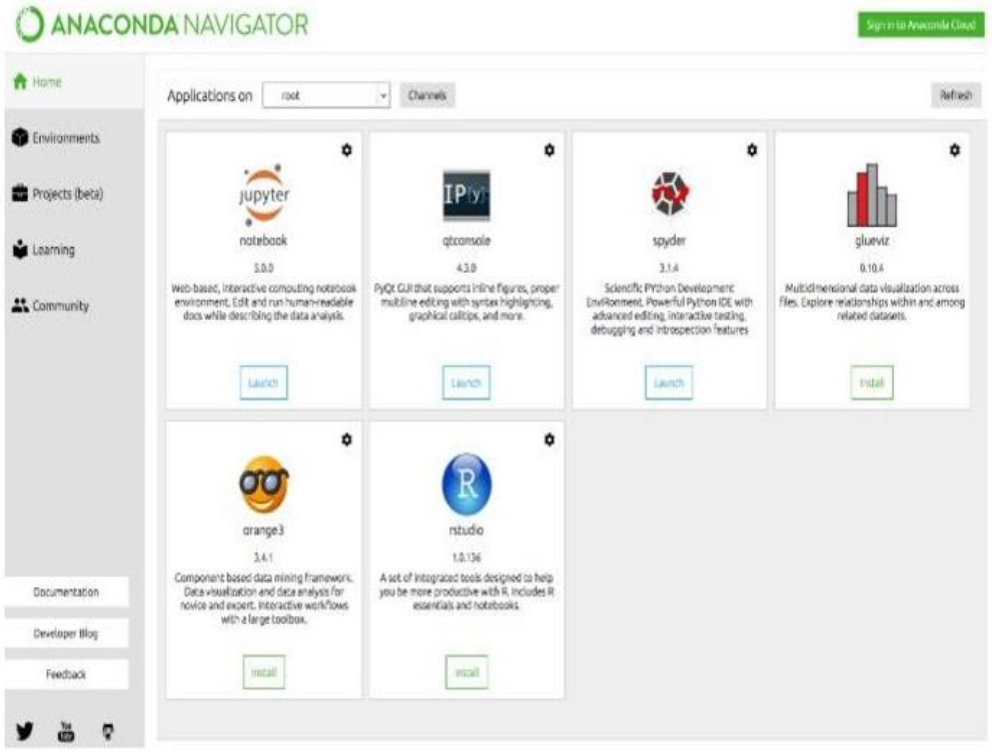


Fig:6.1.2.2 Anaconda Navigator

Applications Provided In Anaconda Distribution

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

1. Jupyter Lab
2. Jupyter Notebook
3. Qt Console
4. Spyder

5. Glue viz

6. Orange3

7. RStudio

8. Visual Studio Code

SQLite

SQLite is not directly comparable to client/server SQL database engines such as MySQL, Oracle, PostgreSQL, or SQL Server since SQLite is trying to solve a different problem.

Client/server SQL database engines strive to implement a shared repository of enterprise data. They emphasize scalability, concurrency, centralization, and control. SQLite strives to provide local data storage for individual applications and devices. SQLite emphasizes economy, efficiency, reliability, independence, and simplicity

SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards.

To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements.

Python sqlite3 module APIs

Following are important sqlite3 module routines, which can suffice your requirement to work with SQLite database from your Python program. If you are looking for a more sophisticated application, then you can look into Python sqlite3 module's official documentation.

sqlite3.connect(database [,timeout ,other optional arguments])

This API opens a connection to the SQLite database file. You can use ":memory:" to open a database connection to a database that resides in RAM instead of on disk. If database is opened successfully, it returns a connection object.

When a database is accessed by multiple connections, and one of the processes modifies the database, the SQLite database is locked until that transaction is committed. The timeout parameter specifies how long the connection should wait for the lock to go away until raising an exception. The default for the timeout parameter is 5.0 (five seconds).

If the given database name does not exist then this call will create the database. You can specify filename with the required path as well if you want to create a database anywhere else except in the current directory.
connection.cursor([cursorClass])

This routine creates a cursor which will be used throughout of your database programming with Python. This method accepts a single optional parameter cursorClass. If supplied, this must be a custom cursor class that extends sqlite3.

connection.close()

This method closes the database connection. Note that this does not automatically call commit(). If you just close your database connection without calling commit() first, your changes will be lost!

Connect To Database

Following Python code shows how to connect to an existing database. If the database does not exist, then it will be created and finally a database object will be returned

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')

print "Opened database successfully";
```

Here, you can also supply database name as the special name :memory: to create a database in RAM. Now, let's run the above program to create our database test.db in the current directory. You can change your path as per your requirement. Keep the above code in sqlite.py file and execute it as shown below. If the database is successfully created, then it will display the following message.

```
$chmod +x sqlite.py

$./sqlite.py

Open database successfully
```

CHAPTER 7

SYSTEM TESTING

7.1 GENERAL

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself.

For example, the design must not have any logical faults. The fault in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walk through.

7.2 TESTING TECHNIQUES

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineer

7.2.1 UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology

is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

The design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken 32into account after the individual united is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

7.2.2 INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field.

The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

7.2.3 FUNCTIONAL TESTING

Functional testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Functional testing mainly involves black box testing and it is not concerned about the source code of the application. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test. The testing can be done either manually or using automation.

It helps in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

7.2.4 SYSTEM TESTING

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

7.2.5 WHITE BOX TESTING

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

Working process of white box testing:

1.Input: Requirements, Functional specifications, design documents, source code.

2.Processing: Performing risk analysis for guiding through the entire process.

3.Proper test planning: Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.

4.Output: Preparing final report of the entire testing process.

In this testing, by knowing the specific functions that a product has been designed to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is white box testing.

7.2.6 BLACK BOX TESTING

Black box testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products. Black box testing can be done in the following ways:

1. Syntax Driven Testing – This type of testing is applied to systems that can be syntactically represented by some language. For example-compilers, language that can be represented by context-free grammar. In this, the test cases are generated so that each grammar rule is used at least once.

2. Equivalence partitioning – It is often seen that many types of inputs work similarly so instead of giving all of them separately we can group them and test only one input of each group. The idea is to partition the input domain of the system into several equivalence classes such that each member of the class works similarly, i.e., if a test case in one class results in some error, other members of the class would also result in the same error.

7.2.7 ACCEPTANCE TESTING

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

The convolutional neural network based system was implemented to classify the disease present in the input skin image (either from a dataset or WebCam). Skin image with different shape & size of the disease images has been fed at the input for training the system. The proposed system is able to classify the different types of skin disease with accuracy of about 99%. The proposed system is able to detect the skin disease with promising results combining computer vision and machine learning techniques. It can be used to help people from all over the world and can be used in doing some productive work. The tools used are free to use and are available for the user, hence, the system can be deployed free of cost. The application developed is light-weight and can be used in machines with low system specifications. It has also a simple user interface for the convenience of the user. The image processing and machine learning algorithms were successfully implemented. For commercial viability, the prototype must have a database that includes most skin diseases with pictures of all skin tones and types. This would lead to efficient and accurate detection. As this system advances, dermatologists will need to understanding of hoe neural networks works, along with when and how it should be appropriately used in an open environment.

APPENDICES

A. Sample Code

- Step1: Install The Required Libraries

```
pip3 install tensorflow tensorflow_hub matplotlib seaborn numpy pandas sklearn imblearn
```

- Step2: Select features of datasets

```
import tensorflow as tf
import tensorflow_hub as hub
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from tensorflow.keras.utils import get_file
from sklearn.metrics import roc_curve, auc, confusion_matrix
from imblearn.metrics import sensitivity_score, specificity_score

import os
import glob
import zipfile
import random

# to get consistent results after multiple runs
tf.random.set_seed(7)
np.random.seed(7)
random.seed(7)

# 0 for benign, 1 for malignant
class_names = ["benign", "malignant"]
```

- Step3: Preparing The Data

```python
# preparing data
# generate CSV metadata file to read img paths and labels from it
def generate_csv(folder, label2int):
    folder_name = os.path.basename(folder)
    labels = list(label2int)
    # generate CSV file
    df = pd.DataFrame(columns=["filepath", "label"])
    i = 0
    for label in labels:
        print("Reading", os.path.join(folder, label, "*"))
        for filepath in glob.glob(os.path.join(folder, label, "*")):
            df.loc[i] = [filepath, label2int[label]]
            i += 1
    output_file = f"{folder_name}.csv"
    print("Saving", output_file)
    df.to_csv(output_file)

# generate CSV files for all data portions, labeling nevus and seborrheic keratosis
# as 0 (benign), and melanoma as 1 (malignant)
# you should replace "data" path to your extracted dataset path
# don't replace if you used download_and_extract_dataset() function
generate_csv("data/train", {"nevus": 0, "seborrheic_keratosis": 0, "melanoma": 1})
generate_csv("data/valid", {"nevus": 0, "seborrheic_keratosis": 0, "melanoma": 1})
generate_csv("data/test", {"nevus": 0, "seborrheic_keratosis": 0, "melanoma": 1})
```

- Step 4: Loading The Data

```python
# loading data
train_metadata_filename = "train.csv"
valid_metadata_filename = "valid.csv"
# load CSV files as DataFrames
df_train = pd.read_csv(train_metadata_filename)
df_valid = pd.read_csv(valid_metadata_filename)
n_training_samples = len(df_train)
n_validation_samples = len(df_valid)
print("Number of training samples:", n_training_samples)
print("Number of validation samples:", n_validation_samples)
train_ds = tf.data.Dataset.from_tensor_slices((df_train["filepath"], df_train["label"]))
valid_ds = tf.data.Dataset.from_tensor_slices((df_valid["filepath"], df_valid["label"]))
```

Now we have loaded the dataset (`train_ds` and `valid_ds`), each sample is a tuple of `filepath` (path to the image file) and `label` (0 for benign and 1 for malignant), here is the output:

- Step5: Load The Image

```
# preprocess data
def decode_img(img):
  # convert the compressed string to a 3D uint8 tensor
  img = tf.image.decode_jpeg(img, channels=3)
  # Use `convert_image_dtype` to convert to floats in the [0,1] range.
  img = tf.image.convert_image_dtype(img, tf.float32)
  # resize the image to the desired size.
  return tf.image.resize(img, [299, 299])


def process_path(filepath, label):
  # load the raw data from the file as a string
  img = tf.io.read_file(filepath)
  img = decode_img(img)
  return img, label


valid_ds = valid_ds.map(process_path)
train_ds = train_ds.map(process_path)
# test_ds = test_ds
for image, label in train_ds.take(1):
    print("Image shape:", image.shape)
    print("Label:", label.numpy())
```

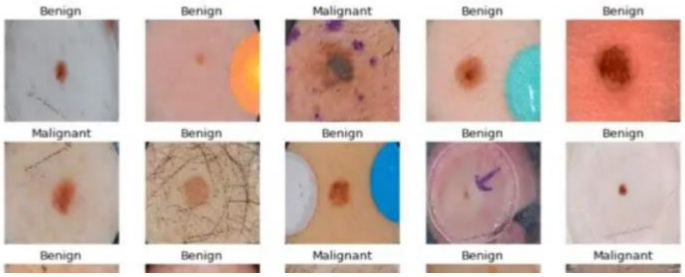
- Step6: The below cell gets the first validation batch and plots the images along with their corresponding label

```
batch = next(iter(valid_ds))

def show_batch(batch):
  plt.figure(figsize=(12,12))
  for n in range(25):
      ax = plt.subplot(5,5,n+1)
      plt.imshow(batch[0][n])
      plt.title(class_names[batch[1][n].numpy()].title())
      plt.axis('off')

show_batch(batch)
```

Output:



- Step7: Build The Model

```
# building the model
# InceptionV3 model & pre-trained weights
module_url = "https://tfhub.dev/google/tf2-preview/inception_v3/feature_vector/4"
m = tf.keras.Sequential([
    hub.KerasLayer(module_url, output_shape=[2048], trainable=False),
    tf.keras.layers.Dense(1, activation="sigmoid")
])

m.build([None, 299, 299, 3])
m.compile(loss="binary_crossentropy", optimizer=optimizer, metrics=["accuracy"])
m.summary()
```

- Step8: Train The Model

```
model_name = f"benign-vs-malignant_{batch_size}_{optimizer}"
tensorboard = tf.keras.callbacks.TensorBoard(log_dir=os.path.join("logs", model_name))
# saves model checkpoint whenever we reach better weights
modelcheckpoint = tf.keras.callbacks.ModelCheckpoint(model_name + "_{val_loss:.3f}.h5", save_best_only=True, verbos

history = m.fit(train_ds, validation_data=valid_ds,
                steps_per_epoch=n_training_samples // batch_size,
                validation_steps=n_validation_samples // batch_size, verbose=1, epochs=100,
                callbacks=[tensorboard, modelcheckpoint])
```

B. SCREENSHOTS

```
print("Evaluating the model...")
loss, accuracy = m.evaluate(X_test, y_test, verbose=0)
print("Loss:", loss, "  Accuracy:", accuracy)
```

Output:

```
Evaluating the model...
Loss: 0.4476394319534302   Accuracy: 0.8
```

```
threshold = 0.23
# get predictions with 23% threshold
# which means if the model is 23% sure or more that is malignant,
# it's assigned as malignant, otherwise it's benign
y_pred = get_predictions(threshold)
accuracy_after = accuracy_score(y_test, y_pred)
print("Accuracy after setting the threshold:", accuracy_after)
```

Output:

```
19/19 [==============================] - 2s 123ms/step
Accuracy after setting the threshold: 0.7883333333333333
```

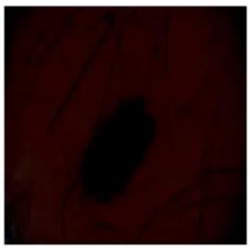
Output:

```
This image is 68.04% malignant.
```

Now a random nevus image:

Output:

```
This image is 78.41% benign.
```



A random seborrheic keratosis one:

Output:

```
This image is 86.32% benign.
```

REFERENCES

1. Arifin, S., Kibria, G., Firoze, A., Amini, A., & Yan, H. (2012) “Dermatological Disease Diagnosis Using Color-Skin Images.” Xian: International Conference on Machine Learning and Cybernetics.
2. Yasir, R., Rahman, A., & Ahmed, N. (2014) “Dermatological Disease Detection using Image Processing and Artificial Neural Network. “Dhaka: International Conference on Electrical and Computer Engineering.
3. Santy, A., & Joseph, R. (2015) “Segmentation Methods for Computer Aided Melanoma Detection.” Global Conference on Communication Technologies.
4. Zeljkovic, V., Druzgalski, C., Bojic-Minic, S., Tameze, C., & Mayorga, P. (2015) “ Supplemental Melanoma Diagnosis for Darker Skin Complexion Gradients.” Pan American Health Care Exchanges
5. Suganya R. (2016) “An Automated Computer Aided Diagnosis of Skin Lesions Detection and Classification for Dermoscopy Images.” International Conference on Recent Trends in Information Technology.
6. Alam, N., Munia, T., Tavakolian, K., Vasefi, V., MacKinnon, N., & Fazel-Rezai, R. (2016) “Automatic Detection and Severity Measurement of Eczema Using Image Processing.” IEEE.
7. Kumar, V., Kumar, S., & Saboo, V. (2016) “Dermatological Disease Detection Using Image Processing and Machine Learning.” IEEE.
8. Krizhevsky, A., ILYA, S., & Geoffrey, E. (2012) “ImageNet Classification with Deep Convolutional Neural Networks.” Advances in Neural Information Processing Systems.
9. Cristianini, N., Shawe, J., “Support Vector Machines”, 2000. [10] SOMMERVILLE, I., “Software Engineering”. 9th .2011

Phase 4: Development Part 2

In this phase you will continue building your project. Please refer below the requirements technology wise:

AI

In this technology you will continue building your project by selecting a machine learning algorithm, training the model, and evaluating its performance. Perform different analysis as needed. After performing the relevant activities create a document around it and share the same for assessment.

ADS:

In this technology you will continue building your project by performing feature engineering, model training and evaluation. Perform different analysis as needed. After performing the relevant activities create a document around it and share the same for assessment.

DAC:

In this technology projects you will continue building your project by performing different analysis, model building and evaluation as per the project requirement. Perform different analysis and visualization using IBM Cognos. After performing the relevant activities create a document around it and share the same for assessment.

IOT:

In this technology project you will continue building your project by developing the platform as per project requirement. Use web development technologies wherever needed. After performing the relevant activities create a document around it and share the same for assessment.

CAD:

In this technology projects you will continue building your project using IBM Cloud Foundry. Perform different functions as per project requirement. After performing the relevant activities create a document around it and share the same for assessment.

NOTE:

File Naming Convention: **TechnologyName_Phase4**

After completion upload your file to your **same private GitHub account** that has been created earlier. Please give access to your **college evaluators email ids**. Also please give access to faculty evaluator[**facultyevaluator@gmail.com**] and industry evaluator [**IndustryEvaluator@skillup.online**] to your private GitHub repository for evaluation process.