Project Report on

Common Skin Disease Diagnosis and Prediction

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Engineering in Information Technology

BY

Atharva Gurav (2243059) **Deep Khadke (2243025)**

Samruddhi Gaikwad (2243020) **Ruchika Gaikwad (2243022)**

Under the guidance of

Mr. R. V. Panchal

Department of Information Technology

Vidya Pratishthans

Kamalnayan Bajaj Institute of Engineering and Technology

Baramati-413133, Dist-Pune (M.S.)

India

May 2023

VPKBIET, Baramati

Certificate

This is to certify that the project report on

Common Skin Disease Diagnosis and Prediction

submitted by

Atharva Gurav (2243059) **Deep Khadke (2243025)**

Samruddhi Gaikwad (2243020) **Ruchika Gaikwad(2243022)**

in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Information Technology at Vidya Pratishthan's Kamalnayan Bajaj Institute Of Engineering and Technology, Baramati under the Savitribai Phule Pune University, Pune. This work is done during year 2022-23 Sem-II, under our guidance.

Mr. R. V. Panchal	Dr. S. A. Takale	Dr. R. S. Bichkar
Project Guide	Head of Dept.	Principal

Examiner 1:——- Examiner 2: - - - - - - -

Acknowledgements

I would like to thank all those who gave us the opportunity to complete this report.Special thanks I give you our seminar guide. Mr. R. V. Panchal whose contribution to the promotion of suggestions and encouragement, helped me to compile my project especially in writing this report.

In addition, I would like to thank you very much for the important role of the project coordinator Mr. P.M. Patil and Head of Department Dr. S.A. Takale gave them permission to use all the necessary materials and materials to complete the project on "Common Skin Disease Diagnosis and Prediction".

I should appreciate the guidance given to the other director and panels especially in our project presentation which has improved our presentation skills thanks to their comments and advice. Finally, but not least, we would like to thank everyone who helped us and encouraged us to work on this project.

Abstract

The integration of computer technology into the healthcare industry has been driven by the proliferation of electronic data. Skin diseases, which can range from common to rare disorders, present a unique challenge for medical professionals in terms of diagnosis. Machine learning and deep learning algorithms have shown potential for improving the early detection of high-risk skin disorders and displacing traditional diagnostic systems.

This paper aims to evaluate the performance of various machine learning and deep learning models in diagnosing skin diseases by analyzing performance indicators. We trained our model using deep learning, a type of machine learning that leverages large data sets, reducing the need for multiple classifiers. This approach enhances dermatology by allowing the machine to continuously learn, categorize input data into appropriate prediction levels, and provide accurate results in a timely manner. Our model utilized Convolutional Neural Network (CNN),MobileNet, Residual Neural Network (ResNet) a widely used method for image classification.

Contents

List of Figures

List of Tables

Keywords

List of keywords-

- **Skin Disease**
- **Machine Learning**
- **Deep Learning**
- **Prediction**
- **Convolutional Neural Network(CNN)**
- **MobileNet**
- **Residual Neural Network(ResNet)**
- **Dermatoscopic Images**

Notation and Abbreviations

- CNN - Convolutional Neural Network
- ResNet- Residual Neural Network
- MobileNet
- DL - Deep Learning
- Relu - REctified Linear Unit
- ML - Machine Learning
- Convolution layer
- Pooling Layer
- Flattening
- Fully Connected
- Android Studio
- Bitmap

Chapter 1

Introduction

1.1 Introduction

Dermatology remains the most uncertain and complex scientific field, because it is complicated in the procedures of diagnosing diseases of hair, skin, and nails. Differences in these diseases can be observed due to many environmental and geographical variations in factors. Usually, it's caused by factors such as different cells of the organism, different diets and internal and external factors such as the hierarchical genetic group of cells, hormones and immune system conditions. These factors may act together or in sequence in skin disease. There are chronic and incurable diseases, such as eczema and psoriasis, and malignant diseases such as malignant melanoma. Recent researchers have found the availability of cures for these diseases if they are detected in the early stages.

Deep learning is a part of the broader family of machine learning wherein the learning can be supervised, unsupervised or semi supervised. Deep learning unlike machine learning uses a large dataset for the learning process and the number of classifiers used gets reduced substantially. The training time for the deep learning algorithm increases because of the usage of the very large dataset. Deep learning algorithm chooses its own features unlike the machine leaning making the prediction process easier for the end user as it does not use much of pre-processing.

A CNN is a type of artificial neural network used in major recognition. CNNs are a category of neural networks that have proven very effective in areas such as image recognition and classification. Convolutional networks have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self-driving cars. CNNs are important tools for most machine learning practitioners today. CNN is a deep learning neural network designed for processing structured arrays of data such as images. Skin diseases occur in almost all age groups of people. The rate of skin diseases has increased due to lifestyle and changing environments. Today, skin diseases are becoming a more common problem in human life. Most of these diseases are dangerous and harmful, especially if not treated in the initial stage. People do not take skin diseases seriously. The most common diseases in the dataset mainly include:

1. Actinic Keratosis
2. Basal Cell Carcinoma
3. Benign Keratosis-Like Lesions
4. Dermatofibroma
5. Melanocytic Nevi
6. Pyogenic Granuloma
7. Melanoma

Despite being common its diagnosis is extremely difficult because of the complexities of skin tone, colour, and presence of hair. We used a convolutional neural network (CNN), MobileNet, ResNet, and Statistical Analysis to detect these skin diseases. A convolutional neural network is a category of deep neural networks where the machine learns on its own and divides the provided data into prediction levels and in a very short time it gives accurate results.

1.2 Motivation

Skin disease is the most common disease in the world. Diagnosing a skin disease requires a high standard expertise and accuracy for a dermatologist, so the computer-aided skin disease diagnosis model is designed to provide more objective and reliable solution. Much research has been done to help detect skin diseases such as skin cancer and skin tumors. However, accurate disease recognition is extremely challenging for the following reasons: low contrast between them lesions and skin, visual similarity between diseased area and non-diseased area, etc. The aim of this article is to detect skin disease from skin image and gray scale the image using a filter to remove noise or unwanted stuff to analyze this image help with processing and get useful information. This helps to provide evidence for any type of skin disease and they show the emergency orientation. The result of the analysis of this study can help the doctor to help in the initial diagnosis and to know the type of disease. It is compatible with the skin and prevents side effects.

Chapter 2

Literature Survey

"Intelligent System for Skin Disease Prediction using Machine Learning"[1]
Skin diseases occur in almost all age groups of people. There are chronic and incurable diseases such as eczema and psoriasis and malignant diseases such as malignant melanoma. Recent scientists have found the availability of drugs for these diseases if they are detected in the early stages. It has been published for the detection of these diseases using an image processing method. The most dangerous form of skin cancer among other types of skin is melanoma. Skin diseases are often difficult to detect at an early stage and even more difficult to classify on their own. Image classification is one of the classic problems in image processing. This article gives us an overview of the existing machine learning and image processing algorithms for skin disease detection through Android application development. One in five people in the US is infected with some kind of skin disease. There are chronic and incurable diseases such as eczema and psoriasis and malignant diseases such as malignant melanoma. To detect these diseases using the image processing method. Skin disease detection methods using methods such as Naïve Bayes, CNN, SVM were used. The most dangerous form of skin cancer is melanoma because it is much more likely to spread to other parts of the body if not diagnosed and treated early. The literature review shows that CNN and SVM are the most suitable algorithms for the detection of skin diseases. In these articles, we have used OpenCV image processing along with machine learning algorithms to detect various skin diseases. The application also provides the doctor with a control panel to manage

his patient remotely and can identify the patient's illness at a remote location. There are approximately 2.3 billion Android devices in use worldwide, which is 1/3 of the total world population. In short, identifying the disease can help reduce the problem of the spread of skin diseases. This will provide an inexpensive method of medical treatment. Most skin diseases can be easily spread by touch. In our application, we used a modified pre-trained convolutional neural network model and SVM algorithm. For the classification of six classes, 92 accuracy is achieved.

"Skin Disease Detection using Machine Learning"[2]
A new system has been developed for the diagnosis of the most common skin lesions. 93 accuracy is achieved in classification using Convolution Neural Networks (CNN) with Keras Application API. The watchword in these steps is "Data Preprocessing and Enhancement: Trash In-Good Out". They examined various properties of the data set, their distributions, and actual counts. Data transformation involves converting data from one format to another. Model Building involves building a deep neural network (CNN or ConvNet).

"Automated Skin Disease Identification using Deep Learning Algorithm"[3]
In order to forecast numerous skin disorders that are prevalent yet challenging to detect owing to complications such skin tone and colour, this study provides a computer vision-based solution employing deep learning. The algorithm predicts skin illnesses based on the highest number of votes using three modified, freely accessible image recognition models (InceptionV3, InceptionResnetV2, and Mobile Net). These models undergo a three-stage process of feature extraction, training, and testing/validation before being pre-trained to recognise 1000 classes using skin photos. The technology aims to anticipate skin disorders with the greatest possible precision. Due to the wide range of illnesses affecting the skin, hair, and nails as well as the difficulties in diagnosing these illnesses, dermatology is a complicated and unreliable branch of study. For the proper diagnosis of skin disorders, a variety of pathological laboratory tests are required. , however, this

research suggests a technique that enables users to forecast skin problems using computer vision without requiring time-consuming laboratory tests. The study outlines a method for predicting skin conditions using computer vision and deep learning. With changes for skin disease prediction, the system leverages three publically accessible image recognition architectures (InceptionV3, InceptionResnetV2, Mobile Net) and predicts the illness based on the combination of votes from the three networks. The technology aims to anticipate skin illnesses as accurately as possible. Due to advancements in medical technology and computers' capacity to handle and analyse massive volumes of data, the use of computer technology in the detection of skin disorders has increased. The study emphasizes the use of supervised, unsupervised, and semi-supervised learning techniques for this purpose, concentrating on machine learning and deep learning algorithms. Three parts make up the proposed computer vision system for predicting skin diseases: feature extraction, training, and testing/validation. The method employs deep learning technology to extract significant characteristics from photos of skin diseases during the feature extraction phase. These architectures have been pre-trained to identify up to 1000 classes of pictures. The system checks the algorithm using validation data during the test/validation phase in order to determine how accurate it is at predicting skin diseases. To forecast skin problems, the algorithm takes the most votes from the three networks. The major objective of this method is to forecast skin diseases as accurately as possible. In comparison to manual, time-consuming approaches that call for specialized expertise, the system employs computer vision and deep learning to deliver a more effective and automated method for identifying skin diseases.

"Detection and Classification of Skin diseases using Deep Learning" [4]
The fastest-growing and most vital tissue in the human body, according to research, is skin. Doctors and modern tools are necessary to identify various skin disorders due to the low visual resolution of skin disease pictures. basis for a picture Dermatologists must be often consulted for manual skin disease diagnosis. Deep learning algorithms have recently been employed in studies on the categorization of skin diseases. The system has an accuracy of 85 on the HAM10000 dataset for skin diseases. A system that divides

pictures of skin lesions into benign, malignant melanoma, benign acne, and eczema categories was designed, built, and tested using AlexNET, a pretrained CNN model. 750 additional photos of burns and skin lesions have been added to the collection, which has been enlarged. Burns and skin wounds are now included in the modern classifications of skin disorders. In this study, warts, shellfish, systemic illness, seborrheic keratosis, nevus, bullous, actinic keratosis, acne, and rosacea were all examined. A deep learning-based technique for detecting skin diseases is included in the proposed framework. This method recognises and categorises skin conditions. Using CNN, Resnet, Alexnet, and Inceptionv3, researchers proposed developing a worldwide categorization system for skin problems. Additionally, it has been demonstrated that Resnet detects skin problems more precisely than other networks.

" Deep Learning Approaches for Prognosis of Automated Skin Disease."[5]
One of the most common diseases on the planet is skin problems. People often ignore the early symptoms of skin conditions. In this study, a classification system for skin disorders was developed using MobileNetV2 and LSTM. Simulated skin injury, chemical exposure, infection of the embryo, immune system and genetic problems are all factors in the development of skin diseases. Technological advances have made it possible to plan and conduct skin observations early in the diagnosis of underlying skin disorders. Through automated skin disease diagnosis methods, skin diseases can be predicted quickly and accurately with high throughput. Skin conditions detected early can help prevent more serious conditions such as skin cancer. This section describes the components of a combined approach for detecting skin disorders. In order to develop a device that can accurately detect skin problems, several factors must be observed. Good image separation is needed to predict skin disease. Standardization is important for computerized methods for identifying skin problems.

“Automatic Skin Disease Diagnosis Using Deep Learning from Clinical Image and Patient Information”[6]

Using deep learning and a pre-trained mobilenet-v2 model, an unique method was developed to identify five prevalent skin illnesses. Overall, the invention of this system marks a huge step forward in medical science and has the potential to significantly improve the quality of life of persons suffering from skin illnesses. A pre-trained mobilenet-v2 model was used to create an automated skin disease diagnostic system. To efficiently classify skin illnesses, the method blends skin photos with clinical patient information. This technology has the potential to give more thorough diagnoses, resulting in improved treatment results for patients by merging skin pictures and patient information. The use of pre-trained models also reduces the time and resources required for training and development, making the system more accessible and convenient to use in a range of scenarios. Dr. Gerbi’s central clinic collected 1137 photographs and patient information, whereas our study collected 239 photos and data from 286 patients at two medical clinics in Ethiopia. The material included skin photos as well as patient information such as age, gender, anatomic areas of skin illness, and symptoms. The study also revealed common symptoms and anatomical areas for five different skin disorders. To categorise skin conditions, the scientists used transfer learning to a pre-trained MobileNet-v2 model. They discovered that the Adam optimizer, a cross-entropy loss function, and a learning rate of 0.0001 produced the best results for both binary and multiclass classification. The algorithm was trained on huge picture datasets before being fine-tuned for skin disease categorization. In conclusion, our work effectively utilised transfer learning to skin disease categorization using the pre-trained MobileNet-v2 model and produced good results. In this work, deep learning techniques were used on clinical photos and patient information to build a smartphone-based skin disease detection system. For the diagnosis of five prevalent skin illnesses, the findings demonstrated good performance with an average accuracy of 97.5, precision of 97.7, recall of 97.7, F1-score of 97.5, and kappa score of 0.976.

“Classification of Skin Disease Using Deep Learning Neural Networks with Mobile Net V2 and LSTM”[7]

A deep learning strategy for skin disease categorization utilizing Mobile Net V2 and Long Short-Term Memory (LSTM) models was developed in this paper. With more than 85 percent accuracy, the suggested system beat existing cutting-edge models such as Fine-Tuned Neural Networks (FTNN), Convolutional Neural Networks (CNN), and Visual Geometry Group (VGG) Very Deep Convolutional Networks. Skin illnesses may be identified using image processing techniques and AI technologies such as Machine Learning, Deep Learning, Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Support Vector Machines (SVM), Bayesian Classifier, Genetic Algorithm (GA), and morphological operations. SVM is not ideal for processing noisy image data, ANN and CNN require a huge quantity of training data, fine-tuned neural network-based models offer high accuracy but require significant effort in calibration, and so on. Back Propagation Neural Networks (BPNN) may forget previously linked weights, Fuzzy Recurrent Neural Networks (FRNN) and Takagi-Surgeon-Kang Fuzzy Classifier are ideal for processing variable-size inputs, and GLCM is a statistical technique that is not invariant to rotation and texture changes. The suggested model, which is based on Mobile Net V2 and LSTM, was evaluated and produced accurate results for skin illnesses (85.34 percent).

Chapter 3

Proposed Work

3.1 Problem Definition

The patient provides an image of the infected area of the skin as an input to the system. Image processing techniques and deep learning techniques are performed on this image.

3.2 Project Objectives

Our goal of the project is to easily and accurately detect the type of skin disease.

- The first phase of the skin disease image is subjected to various kinds of pre-processing techniques, followed by feature extraction.
- The second phase then involves using machine learning algorithms to identify diseases based on skin analysis and observation.
- The proposed system is very advantageous in rural areas where there is access to dermatologists is limited. For this proposed system, we use a Python script based on Pycharm for experimental results.

3.3 Scope of Project

The skin is the largest organ in the human body, which is important for covering the human bone and for protecting the person from any damage, fighting bacteria and other types of diseases and can have a number of potential abnormalities. Several factors they can directly or indirectly affect the skin and cause diseases that can be treated with special drugs and that require other consultations with a doctor. This document will help people to know what the required procedures for the treatment of skin diseases by analyzing the image and extracting useful information to help visualize the infected skin area and classify the image based on the type of skin disease.

3.4 Project Constraints

- Image recognition is a big challenge.
- Dealing with blur images, images should not be blurred and should be in proper format and angle.
- Images should be taken in an illuminated area.
- It does not support GIF or video format files.
- Images other than human skin should not be included.
- Only seven common skin diseases are detected.

Chapter 4

Software Requirement Specification

These are basically qualitative limitations that the system must meet project contracts. The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements. They basically solve problems like:

4.1 Performance Requirements

For the best performance of the software, the user must follow the sequence of activities to achieve the required results.do not proceed to recognize text before the picture is captured. while using the software, the user's action must be consistent and unique. input to the software must be in a proper format.

4.2 Software Quality Attribures/Requirements

- Portability: in API portability can be defined as compatibility of the application with platform-upgraded or downloaded versions.
- Flexibility: The architecture of the application will be flexible enough for some
- later requirements change or application enhancement.
- Maintainability: Whenever there is a change in requirement or bug found the

- application will be easily maintainable.
- Usability: The system is easy to use or user-friendly for users.
- Availability: The application will execute the tasks it is assigned to perform.

4.3 Safety Requirements

- Keep track of loss trends.
- Validate your incoming data.
- The information need to be secured from ethical hackers.

4.4 Security Requirements

Security requirements are needed to prevent any malicious attack that can take place on the project. These requirements are as follows:-

- The information need to be secured from ethical hackers.
- The type of data is used for text detection should not be exposed to any one.

4.5 Hardware Requirements

- Hardware: intel core i5.
- Speed : 2.40 GHz
- RAM : 8GB
- SSD : 2 GB

4.6 Software Requirements

- Operating System : Windows 11, 64 bit
- Technology : Python, Java
- IDE : Jupyter Notebook
- Frontend : Android Studio

Chapter 5

System Architecture

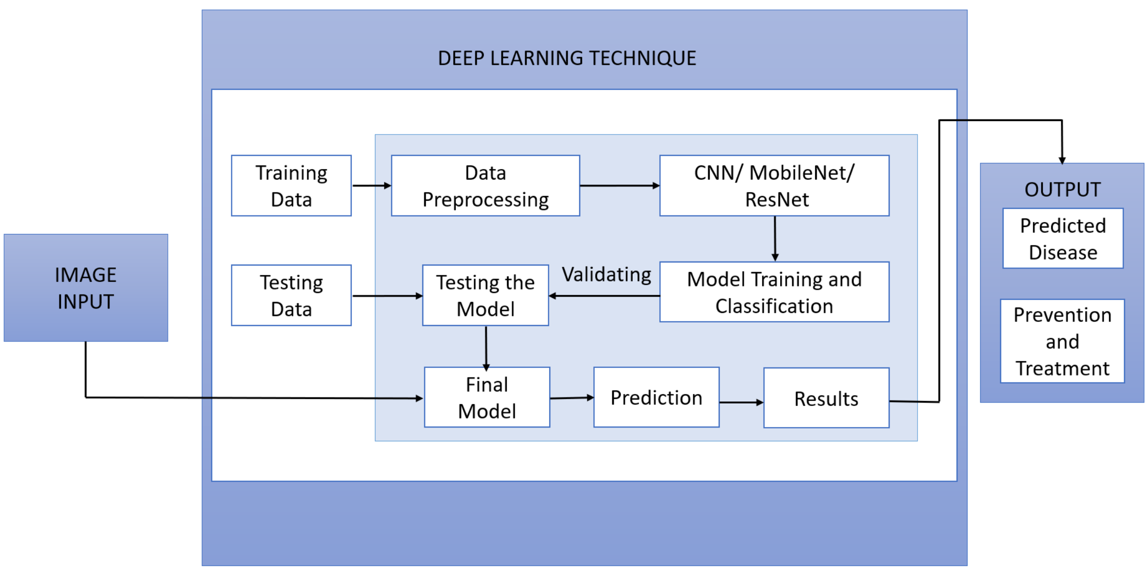

Figure 5.1: System Architecture

1. **Data Collection** - The proposed system has been assessed on skin disease images which is collected from publicly available dataset based on Skin- Cancer-MNIST (Modified National Institute of Standards and Technology Database)-HAM10000. The number of options is endless. To save time and effort one can use publicly available data.

2. **Data Pre-processing** - Dirty data can cause confusion and results in unreliable and poor output. Hence first step in Data Pre-processing is Data Cleaning. Cleaning of data is done by filling in missing values, smoothing noisy data by identifying and/or removing outliers, and removing inconsistencies. In pre-processing the input image data to convert it into meaningful floating-point tensors for feeding into Convolutional Neural Networks. Data Transformation involves converting data from one format into another. It involves transforming actual values from one representation to the target representation.

3. **CNN** - We used a convolutional neural network (CNN). A convolutional neural network (CNN) is a category of deep neural networks where the machine learns itself and divides the given data into predictive levels and provides accurate results in a very short time. A convolutional neural network (CNN) is a deep learning algorithm that consists of a combination of convolutional and pooling layers in sequence and then followed by fully connected layers at the end as a multilayer neural network. CNN excels among all alternative algorithms in image classification. Sparse connectivity, shared weights, and pooling features are critical features to get the best features. Also, the use of graphics processing units (GPUs) reduced the training time of deep learning methods. Huge databases of labeled data and pre-trained networks are now publicly available.

4. **RestNet** - Residual Neural Network (ResNets) is a common neural network architecture used for deep learning computer vision applications such as object detection and image segmentation. Residual Network (ResNet) is a convolutional neural network (CNN) architecture that has overcome the ”vanishing gradient” problem and allowed networks with up to thousands of convolutional layers to be built that outperform shallower networks. It works in two stages:

   - ResNet creates multiple layers that are initially not used, and skips them, reusing activation functions from previous layers.
   - At a second stage, the network re-trains again, and the “residual” convolutional layers are expanded. This makes it possible to explore additional parts of the

feature space which would have been missed in a shallow convolutional network architecture.

5. **MobileNet:** It is a type of convolutional neural network designed for mobile and embedded vision applications. They are based on a simplified architecture that uses depth-separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices. Due to the small size of the model, these models are considered very useful for implementation on mobile and embedded devices. Hence the name MobileNet. The first layer of MobileNet is a full convolution, while all subsequent layers are depth-separable convolutional layers. All layers are followed by batch normalization and ReLU activation. The last classification layer has softmax activation. MobileNet-v2 is a convolutional neural network that is 53 layers deep. You can load a pre-trained version of the network trained on over a million images from the ImageNet database.

6. Model Evaluation - More the accuracy, better is the model. Every model is evaluated based on the accuracy achieved and the loss obtained. There are two accuracies involved: Validation accuracy And Test accuracy. Before this Validation set is different from Train set i.e., Validation set is independent from the Train set, Validation set is used for selecting parameters. Just for an instance if your model has 73 percent train accuracy and 72 percent validation accuracy then your model is expected to have 72 percent accuracy on new data.

Chapter 6

Project Planning

6.1 Project Estimates

Sr. No.	Task Name	No. of Lines in Code
1.	Pre-processing	17
2.	Model Building (CNN) (MobileNet) (ResNet)	 20 17 24
3.	Testing (CNN) (MobileNet) (ResNet)	 15 15 15
4.	GUI (Main Activity) (XML)	 202 129

Table 6.1: Project Estimates

Parameter Estimation Task Duration * Employee's Hourly Rate = Task Cost

Compute Value Adjustment Factor(VAF): Use the following formula to calculate VAF

VAF = (TDI * 0.01) + 0.65

Find the Function Point Count: Use the following formula to calculate

FPC FPC = UFP * VAF

Phases involved in Project Cost Estimation :

- Research/planning
- Design
- Copy-writing
- Front-end/back-end development
- Testing/Bug fixes
- Launch

Software Engineering Estimation Models: Cost estimation simply means a technique that is used to find out the cost estimates. The cost estimate is the financial spend that is done in the efforts to develop and test software in Software Engineering.

1. **Empirical Estimation Technique:-** Empirical estimation is a technique or model in which empirically derived formulas are used to predict data that is a required and necessary part of the planning step of a software project. These techniques are usually based on data collected earlier from the project as well as some guesswork, previous experience developing similar types of projects, and assumptions. It uses the size of the software to estimate the effort.

2. **Analytical Estimation Technique:-** Analytical estimating is a type of technique that is used to measure work. In this technique, a task is first broken down or broken down into its basic sub-operations or elements for analysis. Second, if standard time is available from some other source, then those sources are applied to each element or component of the work.

6.2 Team Structure

Specify your team structure over here. Make use of Latex tables

Names	Tasks
Ruchika Gaikwad	Frontend, Backend Implementation and Documentation
Samruddhi Gaikwad	Frontend, Backend Implementation and Documentation
Atharva Gurav	Frontend, Backend Implementation and Testing
Deep Khadke	Frontend, Backend Implementation and Testing

Table 6.2: Team Structure

Chapter 7

Project Schedule

7.1 Project Breakdown Structure

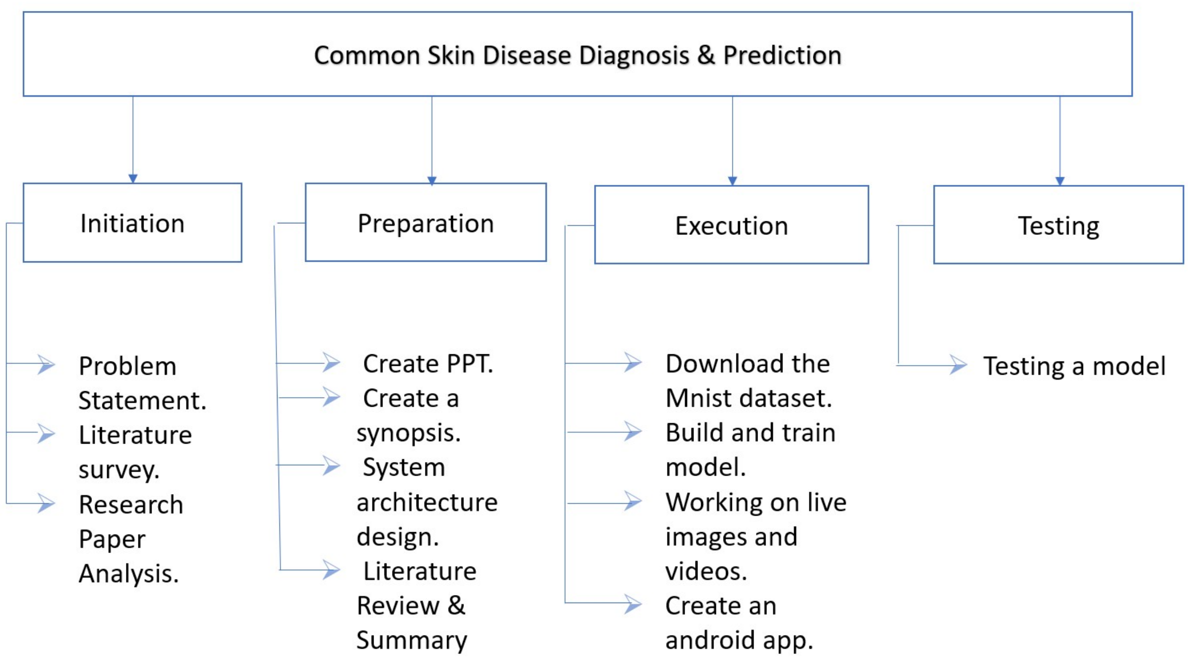

Figure 7.1: Project Breakdown Structure

7.2 Task Network

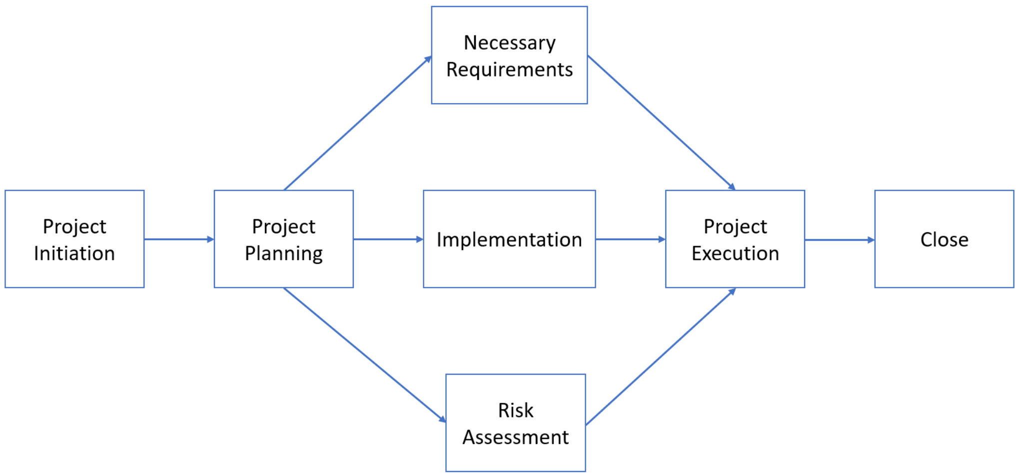

Figure 7.2: Task Network

7.3 Time-line charts

Figure 7.3: Time-line Chart

Chapter 8

Project Design

8.1 UML Diagrams

8.1.1 Use Case Diagram:

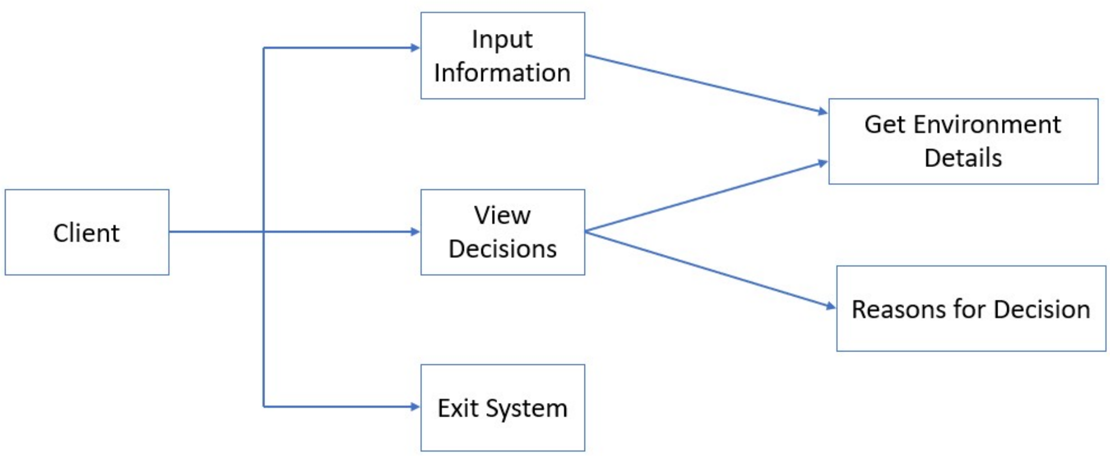

Figure 8.1: Use Case Diagram

8.1.2 Activity Diagram:

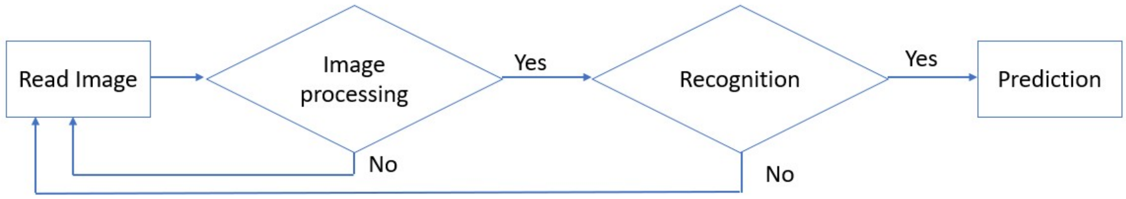

Figure 8.2: Activity Diagram

8.2 Data Flow Diagrams

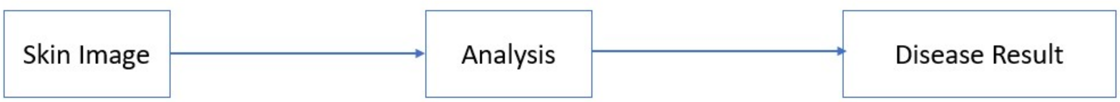

Figure 8.3: Level 0 Data Flow Diagram

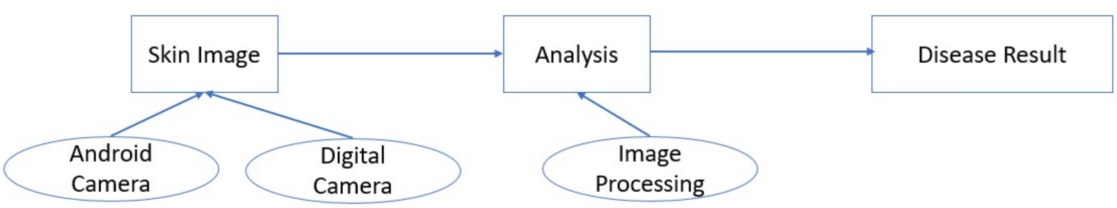

Figure 8.4: Level 1 Data Flow Diagram

8.3 Flowchart

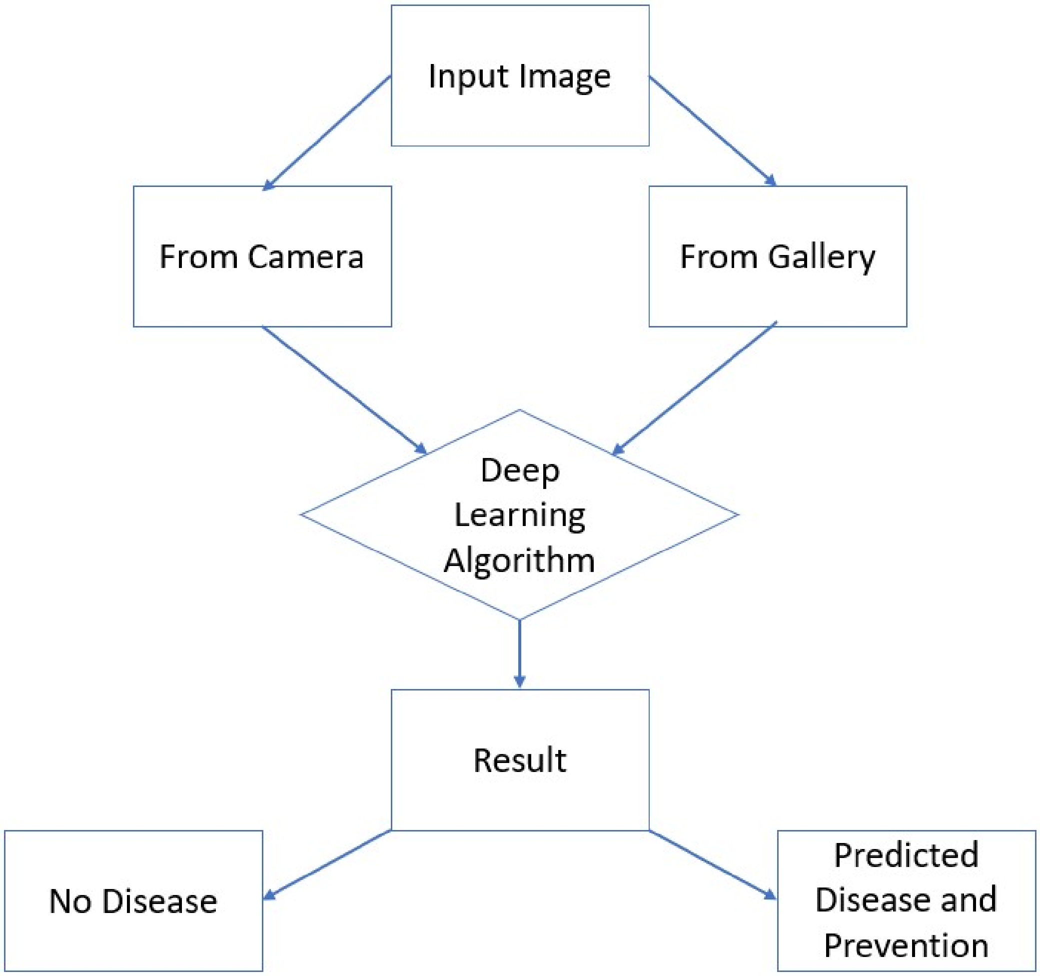

Figure 8.5: Flow Chart

Chapter 9

System Algorithms and Implementation

9.1 Pre-Processing

1. This is Skin Cancer HAM10000 MNIST dataset with 10000+ skin disease images of seven different diseases downloaded from kaggle. The dataset is a csv file with 10015 images consist of 64 by 64 pixel images in rgb format.

2. Here, label column is considered as target variable (Y) and rest columns as independent variables(X).

3. The dataset is divided into 80 percent training and 20 percent testing using train test split method.

4. The independent variables (X) are reshaped into 64 by 64 format and target variable (Y) is converted to categorical format.

5/21/23, 10:02 AM Skin_CNN - Jupyter Notebook

In [1]:
```
import pandas as pd
df = pd.read_csv('hmnist_64_64_RGB.csv')
```

In [2]:
```
df.head()
```

Out[2]:

	pixel0000	pixel0001	pixel0002	pixel0003	pixel0004	pixel0005	pixel0006	pixel0007	pixel000
0	191	152	194	191	153	195	192	149	19
1	24	13	23	24	14	28	37	24	4
2	185	129	140	192	136	151	198	142	15
3	24	11	19	36	19	30	64	38	5
4	138	94	117	158	113	138	178	133	16

5 rows × 12289 columns

In [3]:
```
df.shape
```

Out[3]: (10015, 12289)

In [4]:
```
x = df.drop("label", axis=1).values
y = df["label"].values
```

In [5]:
```
x.shape, y.shape
```

Out[5]: ((10015, 12288), (10015,))

In [6]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state
```

In [7]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[7]: ((8012, 12288), (2003, 12288), (8012,), (2003,))

In [8]:
```
x_train = x_train.reshape(x_train.shape[0], *(64, 64, 3))
x_test = x_test.reshape(x_test.shape[0], *(64, 64, 3))
```

In [9]:
```
x_train.shape, x_test.shape
```

Out[9]: ((8012, 64, 64, 3), (2003, 64, 64, 3))

In [10]:
```
from keras.utils import to_categorical

y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

localhost:8888/notebooks/Skin_CNN.ipynb# 1/7

Figure 9.1: Data Loading and Preprocessing

9.2 CNN Implementation

1. Total 4 convolutional layers and pooling layers are used in the system with 1 flattening layer and 4 fully connected layer and a output layer.

2. Number of filters for first layer or input layer is 32 with kernel size of 3 by 3 and input shape of (64,64,3). The rest three convolutional layer has 64, 128, 256 filters respectively, the activation function used is Relu.

3. Max pooling is used after each convolutional layer with pool size of (2,2).

4. A flattening layer is used to convert outputs to one dimensional array.

5. The fully connected layer has 4 layers and a output layer, numbers of neurons used are 256,128, 64, 32 respectively with Relu activation function and for output layer 7 neurons with softmax activation function.

6. Model compiled using Adam optimizer and loss function categorical crossentropy, model is trained with epoch size 15.

7. Train accuracy = 89 percent, validation accuracy = 73 percent, test accuracy = 72 percent.

5/21/23, 10:02 AM Skin_CNN - Jupyter Notebook

In [12]:
```
#Importing keras libraries
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

#initialising model
cnn = Sequential()

#LAYER 1
#Step 1 - Convolution
cnn.add(Conv2D(filters=32, kernel_size=(3,3), activation='relu', input_shape=(
#step 2 - Pooling
cnn.add(MaxPooling2D(pool_size=(2,2)))

#LAYER 2
#Step 1 - Convolution
cnn.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu'))
#step 2 - Pooling
cnn.add(MaxPooling2D(pool_size=(2,2)))

#LAYER 3
#Step 1 - Convolution
cnn.add(Conv2D(filters=128, kernel_size=(3,3), activation='relu'))
#step 2 - Pooling
cnn.add(MaxPooling2D(pool_size=(2,2)))

#LAYER 4
#Step 1 - Convolution
cnn.add(Conv2D(filters=256, kernel_size=(3,3), activation='relu'))
#step 2 - Pooling
cnn.add(MaxPooling2D(pool_size=(2,2)))


#Flattening Layer
cnn.add(Flatten())

#Fully Connected Layer
cnn.add(Dense(units=256,activation='relu'))
cnn.add(Dense(units=128,activation='relu'))
cnn.add(Dense(units=64,activation='relu'))
cnn.add(Dense(units=32,activation='relu'))
cnn.add(Dense(units=7,activation='softmax'))

#
cnn.summary()
```

localhost:8888/notebooks/Skin_CNN.ipynb# 3/7

Figure 9.2: CNN Implementation

5/21/23, 10:02 AM Skin_CNN - Jupyter Notebook

```
In [13]: cnn.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = [

In [16]: history = cnn.fit(x_train,y_train,epochs=15,verbose=1, validation_split=0.2, b
```

```
Epoch 1/15
201/201 [==============================] - 54s 268ms/step - loss: 0.6849 - ac
curacy: 0.7460 - val_loss: 0.7634 - val_accuracy: 0.7293
Epoch 2/15
201/201 [==============================] - 54s 270ms/step - loss: 0.6847 - ac
curacy: 0.7530 - val_loss: 0.7684 - val_accuracy: 0.7224
Epoch 3/15
201/201 [==============================] - 55s 276ms/step - loss: 0.6682 - ac
curacy: 0.7502 - val_loss: 0.7582 - val_accuracy: 0.7205
Epoch 4/15
201/201 [==============================] - 55s 276ms/step - loss: 0.6462 - ac
curacy: 0.7591 - val_loss: 0.7782 - val_accuracy: 0.7230
Epoch 5/15
201/201 [==============================] - 55s 275ms/step - loss: 0.6220 - ac
curacy: 0.7703 - val_loss: 0.7736 - val_accuracy: 0.7149
Epoch 6/15
201/201 [==============================] - 54s 271ms/step - loss: 0.5907 - ac
curacy: 0.7862 - val_loss: 0.7471 - val_accuracy: 0.7367
Epoch 7/15
201/201 [==============================] - 54s 267ms/step - loss: 0.5837 - ac
curacy: 0.7887 - val_loss: 0.7428 - val_accuracy: 0.7386
Epoch 8/15
201/201 [==============================] - 56s 278ms/step - loss: 0.5434 - ac
curacy: 0.8086 - val_loss: 0.7877 - val_accuracy: 0.7280
Epoch 9/15
201/201 [==============================] - 53s 263ms/step - loss: 0.5084 - ac
curacy: 0.8182 - val_loss: 0.7758 - val_accuracy: 0.7367
Epoch 10/15
201/201 [==============================] - 53s 263ms/step - loss: 0.4841 - ac
curacy: 0.8262 - val_loss: 0.8373 - val_accuracy: 0.7336
Epoch 11/15
201/201 [==============================] - 54s 269ms/step - loss: 0.4569 - ac
curacy: 0.8332 - val_loss: 0.8889 - val_accuracy: 0.7299
Epoch 12/15
201/201 [==============================] - 55s 275ms/step - loss: 0.4616 - ac
curacy: 0.8413 - val_loss: 0.8950 - val_accuracy: 0.7342
Epoch 13/15
201/201 [==============================] - 56s 276ms/step - loss: 0.3833 - ac
curacy: 0.8628 - val_loss: 0.9030 - val_accuracy: 0.7305
Epoch 14/15
201/201 [==============================] - 55s 275ms/step - loss: 0.3381 - ac
curacy: 0.8786 - val_loss: 1.0821 - val_accuracy: 0.7274
Epoch 15/15
201/201 [==============================] - 55s 274ms/step - loss: 0.2984 - ac
curacy: 0.8961 - val_loss: 0.9926 - val_accuracy: 0.7349
```

localhost:8888/notebooks/Skin_CNN.ipynb# 5/7

Figure 9.3: CNN Model Training

```
In [17]: import numpy as np

         x_test=np.array(x_test).reshape(-1,64,64,3)
         score=cnn.evaluate(x_test,y_test)
         print("loss: ",score[0])
         print("Accuracy: ",score[1])

         63/63 [==============================] - 5s 71ms/step - loss: 1.1150 - accuracy: 0.7234
         loss:  1.114950180053711
         Accuracy:  0.7234148979187012
```

Figure 9.4: CNN Model Accuracy

9.3 MobileNet Implementation

1. MobileNet is a transfer learning model used for image classification, input layer has input shape of (64,64,3) with weights as 'imagenet'.
2. It has 85 pre defined layers, 4 new layers and a output layer.
3. The first layer is dropout layer with value of 0.5, second is global average pooling layer.
4. Next is a Dense layer with 128 neurons and Relu activation function and a Batch Normalization layer.
5. output layer is a Dense layer with 7 neurons and Sigmoid activation function.
6. Model compiled using Adam optimizer and loss function categorical crossentropy, model is trained with epoch size 5 and Early Stopping is used
7. Train accuracy = 82 percent, validation accuracy = 73 percent, test accuracy = 73 percent.

```
In [12]: from keras.applications.mobilenet import MobileNet

In [13]: base_model = MobileNet(weights="imagenet", include_top=False , input_shape=(64, 64, 3))

         WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not in [128, 160, 192, 224]
         (224, 224) will be loaded as the default.

In [14]: from keras.models import Sequential , Model
         from keras.layers import Dropout , GlobalAveragePooling2D
         from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization

         x = base_model.output
         x = Dropout(0.5)(x)
         x = GlobalAveragePooling2D()(x)
         x = Dense(128, activation='relu')(x)
         x = BatchNormalization()(x)
         predictions = Dense(7, activation='sigmoid')(x)
         model = Model(inputs=base_model.input, outputs=predictions)
```

Figure 9.5: MobileNet Implementaion

```
In [16]: model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [17]: from tensorflow.keras.callbacks import EarlyStopping

         es = EarlyStopping(monitor='val_accuracy',patience=5,mode='max',restore_best_weights=True)

In [18]: hist = model.fit(x_train,y_train,epochs=5,batch_size=32,verbose=1,validation_split=0.2,callbacks=es)

         Epoch 1/5
         201/201 [==============================] - 114s 554ms/step - loss: 1.1627 - accuracy: 0.6477 - val_loss: 1.3204 - val_accuracy:
         0.7118
         Epoch 2/5
         201/201 [==============================] - 113s 561ms/step - loss: 0.8123 - accuracy: 0.7323 - val_loss: 0.7414 - val_accuracy:
         0.7542
         Epoch 3/5
         201/201 [==============================] - 123s 610ms/step - loss: 0.6362 - accuracy: 0.7747 - val_loss: 0.8563 - val_accuracy:
         0.7442
         Epoch 4/5
         201/201 [==============================] - 112s 556ms/step - loss: 0.5774 - accuracy: 0.7950 - val_loss: 0.7076 - val_accuracy:
         0.7598
         Epoch 5/5
         201/201 [==============================] - 112s 556ms/step - loss: 0.5083 - accuracy: 0.8248 - val_loss: 0.9149 - val_accuracy:
         0.7367

In [19]: import numpy as np

         x_test=np.array(x_test).reshape(-1,64,64,3)
         score=model.evaluate(x_test,y_test)
         print("loss: ",score[0])
         print("Accuracy: ",score[1])

         63/63 [==============================] - 5s 79ms/step - loss: 0.9060 - accuracy: 0.7389
         loss:  0.905958354473114
         Accuracy:  0.7388916611671448
```

Figure 9.6: MobileNet Model Training

9.4 ResNet Implementation

- ResNet50 is a transfer learning model used for image classification, input layer has input shape of (64,64,3) with weights as 'imagenet'.
- It has 50 pre defined layers, 3 new layers and a output layer.
- The first and third layer is dropout layer with value of 0.5.
- Second layer is a Dense layer with 128 neurons and Relu activation function.
- output layer is a Dense layer with 7 neurons and Softmax activation function.
- Model compiled using Adam optimizer and loss function categorical crossentropy, model is trained with epoch size 5 and Early Stopping is used
- Train accuracy = 80 percent, validation accuracy = 76 percent, test accuracy = 73 percent.

```
model = Sequential()
num_labels = 7

base_model = ResNet50(include_top=False,input_shape=(64,64,3),pooling = 'avg', weights="imagenet")
model = Sequential()
model.add(base_model)
model.add(Dropout(0.5))
model.add(Dense(128, activation="relu",kernel_regularizer=regularizers.l2(0.02)))
model.add(Dropout(0.5))
model.add(Dense(num_labels, activation = 'softmax',kernel_regularizer=regularizers.l2(0.02)))

for layer in base_model.layers:
    layer.trainable = False
for layer in base_model.layers[-22:]:
    layer.trainable = True

model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 dropout_2 (Dropout)         (None, 2048)              0

 dense_2 (Dense)             (None, 128)               262272

 dropout_3 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 7)                 903

=================================================================
Total params: 23,850,887
Trainable params: 9,194,503
Non-trainable params: 14,656,384
```

Figure 9.7: ResNet Implementation

```
In [29]: model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [30]: from tensorflow.keras.callbacks import EarlyStopping

         es = EarlyStopping(monitor='val_accuracy',patience=5,mode='max',restore_best_weights=True)

In [31]: hist = model.fit(x_train,y_train,epochs=5,batch_size=32,verbose=1,validation_split=0.2,callbacks=es)

         Epoch 1/5
         201/201 [==============================] - 269s 1s/step - loss: 3.2002 - accuracy: 0.6767 - val_loss: 1.8418 - val_accuracy: 0.
         7293
         Epoch 2/5
         201/201 [==============================] - 263s 1s/step - loss: 1.4201 - accuracy: 0.7284 - val_loss: 1.1383 - val_accuracy: 0.
         7642
         Epoch 3/5
         201/201 [==============================] - 291s 1s/step - loss: 1.0188 - accuracy: 0.7535 - val_loss: 0.9837 - val_accuracy: 0.
         7517
         Epoch 4/5
         201/201 [==============================] - 307s 2s/step - loss: 0.8407 - accuracy: 0.7786 - val_loss: 0.9300 - val_accuracy: 0.
         7330
         Epoch 5/5
         201/201 [==============================] - 311s 2s/step - loss: 0.7322 - accuracy: 0.8028 - val_loss: 0.8999 - val_accuracy: 0.
         7629

In [32]: import numpy as np
         x_test=np.array(x_test).reshape(-1,64,64,3)
         score=model.evaluate(x_test,y_test)
         print("loss: ",score[0])
         print("Accuracy: ",score[1])

         63/63 [==============================] - 18s 292ms/step - loss: 0.9450 - accuracy: 0.7294
         loss:  0.9449655413627625
         Accuracy:  0.7294058799743652
```

Figure 9.8: ResNet Model Training

9.5 Android Implementation

1. Import Tensorflow lite model into android studio.
2. Add Buttons for choosing images and prediction.
3. Choose image either from gallery or camera.
4. reshape image to 64 by 64 pixel.
5. Input image will be converted to 64,64,3 format. and will be provide to model as input.
6. model will return float array with probabilities and maximum value will be choose and displayed the corresponding disease name.

```java
bitmap = Bitmap.createScaledBitmap(bitmap, dstWidth: 64, dstHeight: 64, filter: true);

try {
    Resnet model = Resnet.newInstance(getApplicationContext());

    // Creates inputs for reference.
    TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 64, 64, 3}, DataType.FLOAT32);

    TensorImage tensorImage = new TensorImage(DataType.FLOAT32);
    tensorImage.load(bitmap);
    ByteBuffer byteBuffer = tensorImage.getBuffer();

    inputFeature0.loadBuffer(byteBuffer);

    // Runs model inference and gets result.
    Resnet.Outputs outputs = model.process(inputFeature0);
    TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();

    // Releases model resources if no longer used.


    float[] con = outputFeature0.getFloatArray();
    int maxpos = 0;
    float maxcon = 0;
    for (int i = 0; i < con.length; i++) {
        if (con[i] > maxcon) {
            maxcon = con[i];
            maxpos = i;
        }
```

Figure 9.9: Android Implementation

Chapter 10

Test Cases

Test Case ID	Test Case	Objective	Expected Result	Actual Result
1	Real time image upload	choose image from gallery or camera	should accept 'jpg' and 'png' format file	only 'jpg' and 'png' file allowed
2	Image pre-processing	Employ pre-Processing operations	image should be transformed	yes image is enhanced
3	Model training and testing	pass image to required algorithms	Algorithm should give required results	algorithm and finds textual parts
4	Disease Prediction	predict disease	Disease prediction with max accuracy	Disease prediction and prevention

Table 10.1: Test Cases

Chapter 11

Experimental Results

11.1 GUI

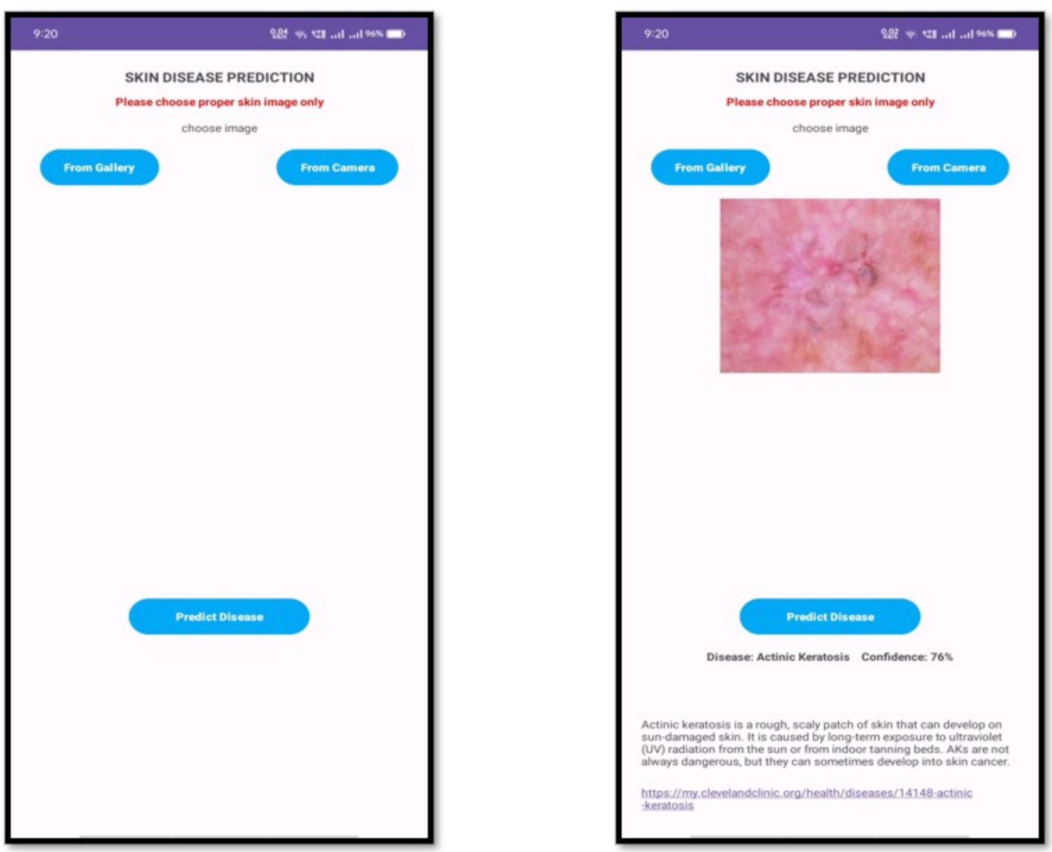

Figure 11.1: GUI

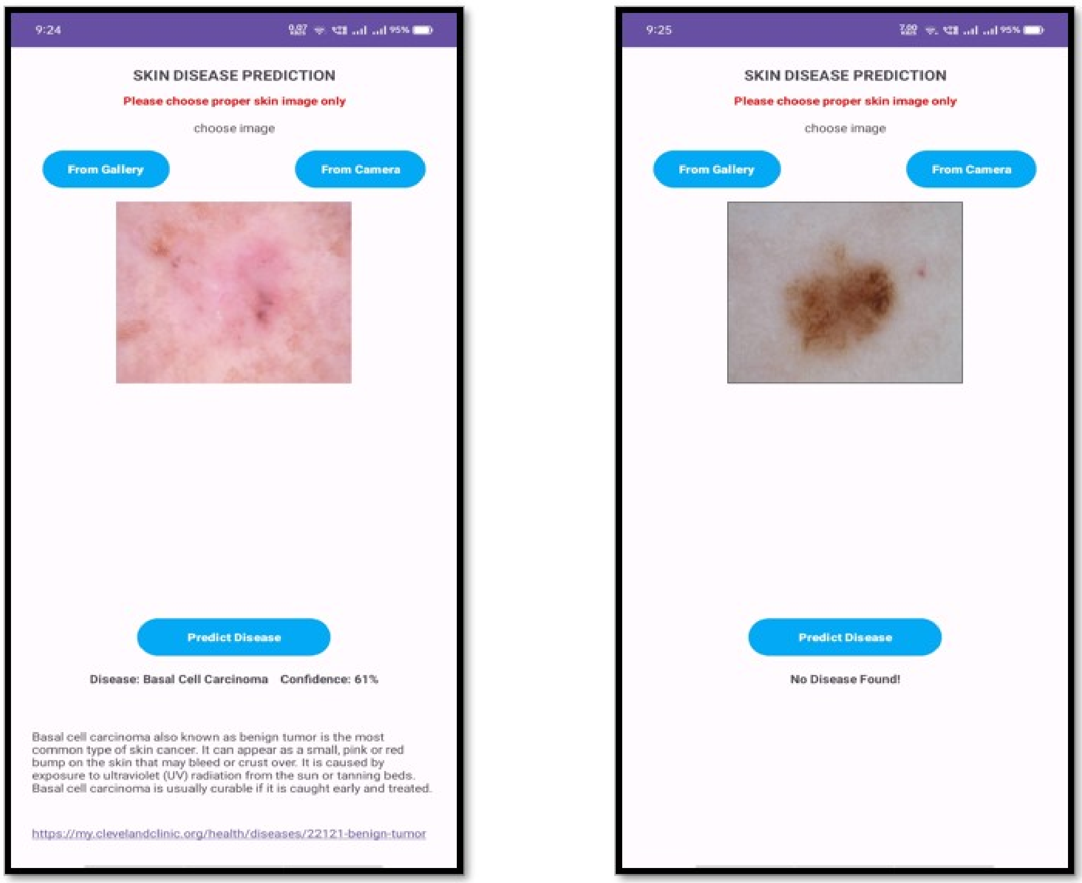

Figure 11.2: GUI

11.2 Working Modules

1. Disease Detection : There are some skin diseases that are the most dangerous. We have selected several common skin diseases as experimental objects Disease problems certainly affect human skin along with other skin factors like hair. In this field, disease detection is an important step. If the disease is not detected at an early stage will lead to various harmful skin diseases like cancer, rashes, etc. Machine vision framework uses image processing techniques to perform such specific work, which is why image processing assumes an exceptionally significant job in their capabilities.

2. Recommendation based on diseases : The first model predicts the disease due to which skin is infected and then by displaying not only the disease name but also gives recommendations that how to prevent the spread of this disease and which precautions should be taken to overcome this problem. It recommends to the user what they should be done.

11.3 Experimental Results and Discussions

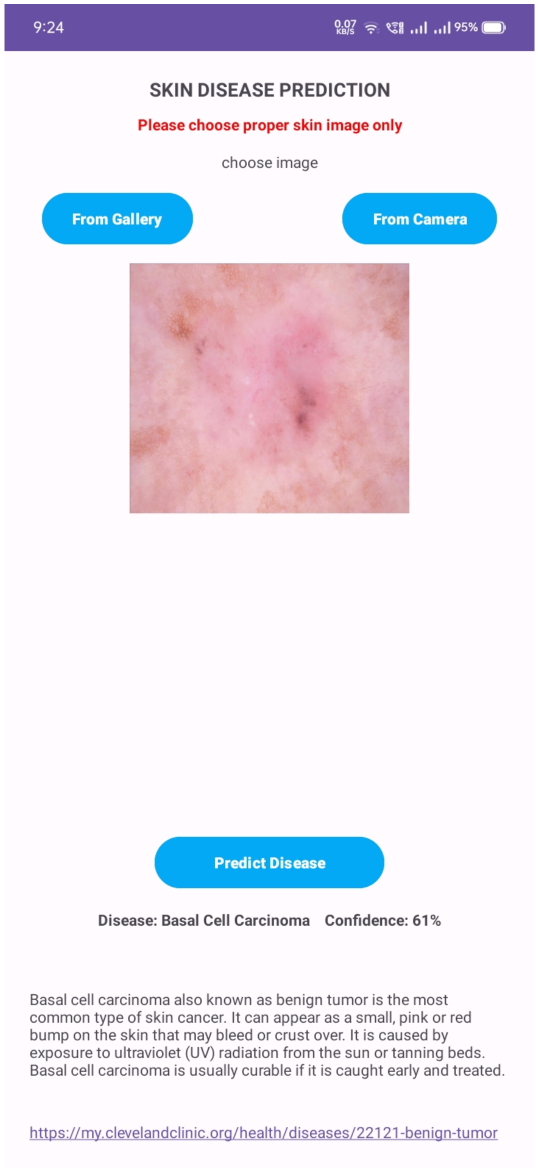

Figure 11.3: Result

Chapter 12

Conclusions

In this work, a model for the prediction of skin diseases using deep learning algorithms is created. It is found that by using feature compounding and deep learning we can achieve higher accuracy and also go for it predicting many more diseases than other previous models done before. Like the previous models in this one area of use were able to report a maximum of six skin conditions with a maximum accuracy of 72 percent. According to by implementing a deep learning algorithm, we are able to predict up to 7 diseases with a accuracy of 73 percent.

This proves that deep learning algorithms have huge potential in real-world skin disease diagnosis. If even better a system with high-end system hardware and software is used with a very large data set, the accuracy can be increased considerably and the model can be used for clinical experimentation as it has any invasive measures.

Future work can be extended to make this model a standard procedure for the method of preliminary diagnosis of skin diseases, as it will reduce time of treatment and diagnosis.

References

[1] Ahmed A. Elngar, Rishabh Kumar, Amber Hayat, Prathamesh Churi, ”Intelligent System for Skin Disease Prediction using Machine Learning”, August 2021.

[2] Kritika Rao, Pooja Yelkar, Omkar Pise and Dr. Swapna Borde,and ,”Skin Disease Detection using Machine Learning”, 2021.

[3] Sourav Kumar Patnaik, Mansher Singh Sidhu, Yaagyanika Gehlot, Bhairvi Sharma and P Muthu “Automated Skin Disease Identification using Deep Learning Algorithm”, September 2018.

[4] T. Swapna, D.A. Vineela, M. Navyasree, N. Sushmtha, P. Bhavana “Detection and Classification of Skin diseases using Deep Learning.”

[5] Pravin R. Kshirsagar, Hariprasath Manoharan, S. Shitharth, Abdulrahman M. Alshareef, Nabeel Albishry and Praveen Kumar Balachandran.” Deep Learning Approaches for Prognosis of Automated Skin Disease.”

[6] K. A. Muhaba1, K. Dese, T. M. Aga, F. T. Zewdu, G. L. Simegn. “Automatic Skin Disease Diagnosis Using Deep Learning from Clinical Image and Patient Information”

[7] Parvathaneni Naga Srinivasu, Jalluri Gnana Siva Sai, Muhammad Fazal Ijaz 3, Akash Kumar Bhoi, Won Joon Kim, and James Jin Kang. “Classification of Skin Disease Using Deep Learning Neural Networks with Mobile Net V2 and LSTM.”

Appendix A

Appendix

turnitin

Similarity Report ID. oid:7916:35980547

PAPER NAME	AUTHOR
BE_Project_Report_Sem_II_2022_23_Template__2_ (2).pdf	D

WORD COUNT	CHARACTER COUNT
9880 Words	55883 Characters
PAGE COUNT	FILE SIZE
65 Pages	6.2MB
SUBMISSION DATE	REPORT DATE
May 23, 2023 12:18 PM GMT+5:30	May 23, 2023 12:19 PM GMT+5:30

27% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 27% Internet database
- 1% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Crossref database
- Submitted Works database
- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)
- Manually excluded sources
- Manually excluded text blocks

Appendix B

Base Paper

Skin Disease Detection using Machine Learning

Authors:

- Kritika Sujay Rao, Department of Computer Engineering Vidyvardhini's College of Engineering And Technology (Mumbai University) Vasai, India.
- Pooja Suresh Yelkar, Department of Computer Engineering Vidyavardhini's College of Engineering And Technology (Mumbai University) Vasai, India.
- Omkar Narayan Pise Department of Computer Engineering Vidyavardhini's College of Engineering And Technology (Mumbai University) Vasai, India.
- Dr. Swapna Borde Department of Computer Engineering Vidyavardhini's College of Engineering And Technology(Mumbai University).

Special Issue - 2021
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NTASU - 2020 Conference Proceedings

Skin Disease Detection using Machine Learning

Kritika Sujay Rao
Department of Computer Engineering
Vidyvardhini's College of Engineering And Technology (Mumbai University)
Vasai, India

Pooja Suresh Yelkar
Department of Computer Engineering
Vidyavardhini's College of Engineering And Technology (Mumbai University)
Vasai, India

Omkar Narayan Pise
Department of Computer Engineering
Vidyavardhini's College of Engineering And Technology (Mumbai University)
Vasai, India

Dr. Swapna Borde
Department of Computer Engineering
Vidyavardhini's College of Engineering And Technology(Mumbai University)

Abstract*— Dermatology is the branch of bioscience that's involved with diagnosing and treatment of skin based mostly disorders. The immense spectrum of dermatologic disorders varies geographically and additionally seasonally because of temperature, humidness and alternative environmental factors. Human skin is one amongst the foremost unpredictable and tough terrains to mechanically synthesize and analyse because of its quality of unevenness, tone, presence of hair and alternative mitigating options. Though, many researches are conducted to find and model human skin victimisation (PC Vision techniques), only a few have targeted the medical paradigm of the matter. Due to lack of medical facilities available in the remote areas, patients usually ignore early symptoms which may worsen the situation as time progresses. Hence, there is a rising need for automatic skin disease detection system with high accuracy. Thus, we develop a multiclass deep learning model to differentiate between Healthy Skin Vs Skin suffering from a Disease and Classification of Skin Diseases into its main classes like MelanocyticNevi, Melanoma, Benign keratosis-like lesions, Basal cell Carcinoma, ActinicKeratoses, Vascular lesion and Dermatofibroma. We have used Deep Learning to train our model, Deep Learning is a part of Machine Learning in which unlike Machine Learning it uses large dataset and hence the number of classifiers is reduced substantially. The machine learns itself and divide the data provided into the levels of prediction and in a very short period of time gives the accurate results, thereby promoting and supporting development of Dermatology. The algorithm that we have used is Convolutional Neural Network (CNN) as it is one of the most preferred algorithm for image classification.

Keywords—Dermatoscopic images, Deep Learning, Data Enhancement, Convolutional Neural Network(CNN), Model Training, Testing and Evaluation.

I. INTRODUCTION

Artificial Intelligence is taking over automation in all fields of application even within the healthcare field. In the past years these diseases have been a matter of concern due to the sudden arrival and the complexities which has increased life risks. These Skin abnormalities are very infectious and the require to be treated at earlier stages to avoid it from spreading. The majority of diseases is caused by unprotected exposure to excessive Ultraviolet Radiation(UR). Among all, benign type is considered to be less dangerous than malignant melanoma and can be cured with proper treatment, whereas the deadliest form of skin lesion is malignant Melanoma. The survey results indicate that the back and lower extremity, trunk and upper extremity are heavily compromised regions of skin cancer. There are large instances of patients with age ranging from 30 to 60. Also, MelanocyticNevi, Carcinoma and Dermatofibroma are not prevalent below the age of 20years.

II. EXISTING TECHNOLOGY

A. *Artifical 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 80% which isn't 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

B. 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].

C . Support Vector Machine(SVM).

SVM is a supervised non-linear classifier which constructs an optimal n-dimensional hyperplane to separate all the data points in two categories [2]. In SVM, choosing an honest kernel function isn't easy. It requires long training time for large datasets. Since the final model is not easy to use we cannot make small calibrations to the model and it becomes difficult to tune the parameters used in SVMs. SVMs when compared with ANNs always give best results [3].

III. LITERATURE

Skin diseases are the 4th common cause of skin burden worldwide. Robust and Automated system have been developed to lessen this burden and to help the patients to conduct the early assessment of

Special Issue - 2021
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NTASU - 2020 Conference Proceedings

the skin lesion. Mostly this system available in the literature only provide skin cancer classification. Treatments for skin are more effective and less disfiguring when found early and it is a challenging research due to similar characteristics of skin diseases. In this project we attempt to detect skin diseases .A novel system is presented in this research work for the diagnosis of the most common skin lesions (Melanocytic nevi, Melanoma, Benign keratosis-like lesions, Basal cell carcinoma, Actinic keratoses, Vascular lesion, Dermatofibroma). The proposed approach is based on the pre-processing, Deep learning algorithm, training the model , validation and classification phase. Experiments were performed on 10010 images and 93% accuracy is achieved for seven-class classification using Convolution Neural Networks (CNN) with the Keras Application API.

IV. DATASET

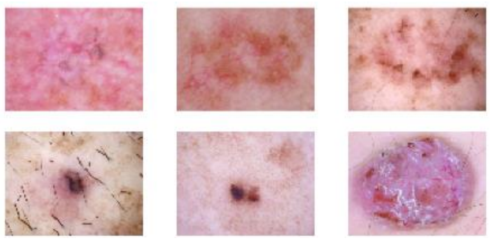

Fig. 1. Sample Data

The fig. above is the sample dataset which we have trained and tested.

V. IMPLEMENTATION (METHODOLOGY)

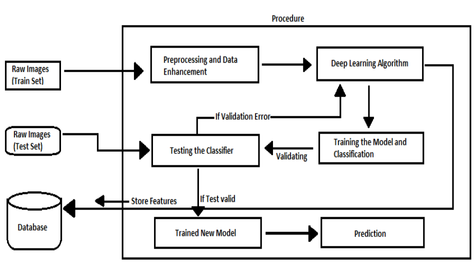

Fig.2. Procedure

To develop any ML-AI based system, be it this system; following steps are to be followed.

1. Data Gathering.

The proposed system has been assessed on dermatoscopic images which is collected from publicly available dataset based on Skin-Cancer-MNIST (Modified National Institute of Standards and Technology Database)-HAM10000. The number of options is endless. To save time and effort one can use publicly available data.

2. Data Preprocessing & Enhancement.

"Trash In- Good Out" is the basic motto in this step [6]. Validating your dataset with some basic profiling procedure will help speeding up the process, by slip-ups and grimy information [4]. AI algorithms don't give great outcomes when working with such information.

2.1. Data Cleaning.

Dirty data can cause confusion and results in unreliable and poor output. Hence first step in Data Pre-processing is Data Cleaning. Cleaning of data is done by filling in missing values, smoothing noisy data by identifying and/or removing outliers, and removing inconsistencies.

2.2. Data Transformation.

Data Transformation involves converting data from one format into another. It involves transforming actual values from one representation to the target representation.

2.3. Exploratory Data Analysis (EDA).

In this we explore different features of the dataset, their distributions and actual counts.

2.4. Label Encoding.

The dataset is labelled into 7 different categories:

1. MelanocyticNevi
2. Melanoma
3. Benign keratosis-like lesions
4. Basal cell carcinoma
5. ActinicKeratoses
6. Vascular lesions
7. Dermatofibroma

3. Training.

For this we have to divide the data into training set and testing set. This division can be in any ratio. Also, the batch size and number of epochs has to be decided beforehand.

4. Model Building.

We have used Convolutional Neural Network (CNN). A **Convolutional Neural Network** (**CNN** or **ConvNet**) is a category of deep neural networks, where the machine learns on its own and divide the data provided into the levels of prediction and in a very short period of time gives the accurate results [2]. A Convolutional Neural Network (CNN) is an algorithm in deep learning which consist of a combination of convolutional and pooling layers in sequence and then followed by fully connected layers at the end as like multilayer neural network [2]. CNN stands out among all alternative algorithms in classifying images. Crucial characteristics are Sparse Connectivity, Shared Weights and Pooling Feature so as to extract the best features. Also, the use of Graphical Processing Units (GPUs) have shrivelled the training time of deep learning methods. Giant databases of lasbelled data and pre-trained networks are now publicly available.

The figure below shows the difference between Sparse and Dense Connectivity.

Special Issue - 2021
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NTASU - 2020 Conference Proceedings

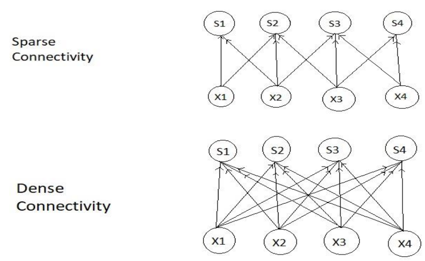

Fig.3. Sparse and Dense Connectivity

4.1. Explanation.

We have used **Keras Sequential API**, where you have just to add one layer at a time, starting from the input. **Conv2D** layer, a set of learnable features. The number of filters used here is thirty two. Each filter transforms a part of the image which is defined by the kernel size using the kernel filter. Transformed images are the filter maps. Next important layer is the pooling layer which simply acts as a down sampling filter. We have Max pooling, MaxPool() picks the **maximal worth** among set **of** two neighbouring pixels. This layer is employed to scale back (to cut back) machine value and additionally reduce overfitting to some extent. Combining both the above layers, CNN gets the ease to combine local features and learn global features. Activation Function **relu** is used to add non-linearity to the network. We use a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample i.e. the **Dropout** function. This improves in generalizing the network. Now, to convert the final feature maps into a one single 1D vector we need to flatten them, thus **Flatten** Layer is used. This flattening step is needed so that you can make use of fully connected layers after some of the above layers. It combines all the found native options of the previous convolutional layers. In the last layer, **Dense()** is used which gives the net output distribution of likelihood of every category. Once layers have been added, we need to set up a score function, a loss function and a proper optimization algorithm. We define **binary cross entropy** as our loss function which will actually measure the error rate between the observed labels and predicted labels. Next most important is the optimizer. **Adam Optimizer** has advantage as it involves functions of other optimizers as well. Adam is a well known and popular algorithm in the field of learning models. Next is the metric function which is used to evaluate the performance of the system, metric **accurac**y is used. **Learning Rate(LR)** is another important term. It is an annealing method. Ideally one should have a decreasing Learning Rate during rate so as to have minimal loss. **ReduceLROnPlateau** is used, the name itself means reduce the LR so as to reach the global minimum of loss function.

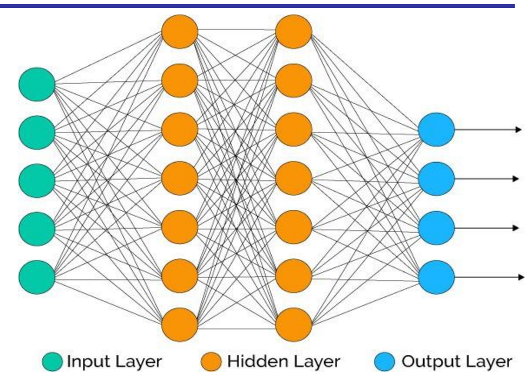

Fig.4. Fully Connected Network

5. *Model Evaluation.*

"More the accuracy, better is the model". Every model is evaluated based on the accuracy achieved and the loss obtained. There are two accuracies involved: Validation accuracy And Test accuracy. Before this Validation set is different from Train set i.e. Validation set is independent from the Train set, Validation set is used for selecting parameters. Just for an instance if your model has 90% train accuracy and 89% validation accuracy then your model is expected to have 89%accuracy on new data.

6. *Graphical Analysis.*

This involves plotting Histogram and the Confusion Matrix.
Confusion Matrix involves TP, TN. FP, FN [2]. The set of decision made by classification algorithm contains correct decision (true positive) and incorrect decision (false positive). False negative are those decision which is declared as negative by classification algorithm but actually they are positive. True negative are those decision which is correctly identified as negative by the classification algorithm.

VI. OVERVIEW

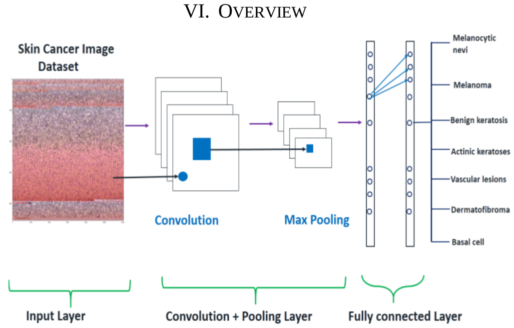

Fig.5. Convolution Neural Network (CNN)

Special Issue - 2021
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NTASU - 2020 Conference Proceedings

VII. Results

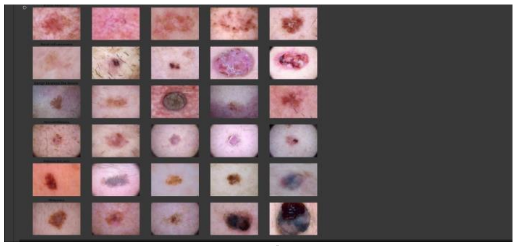

Fig.6. Streamed Images

Fig.7. Model Summary

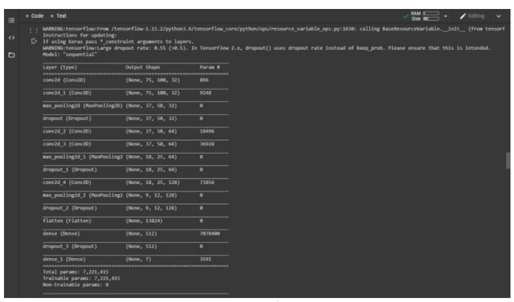

Fig.8. Epoch-50

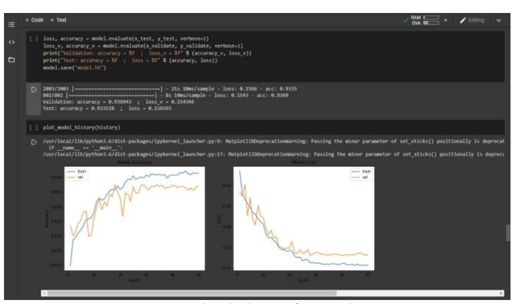

Fig.9. Graphical Plotting for Epoch-50

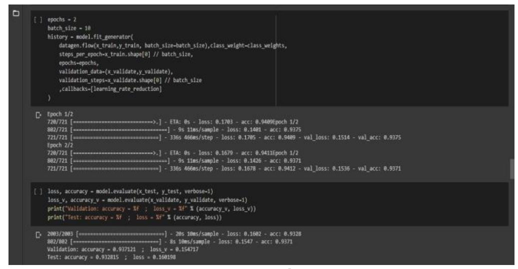

Fig.10. Epoch-2

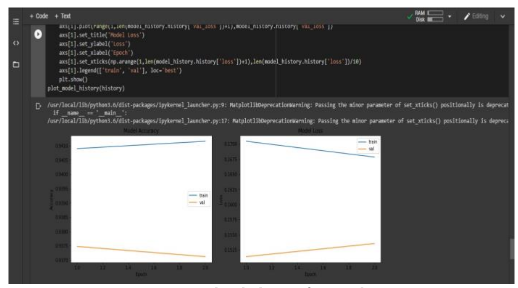

Fig.11. Graphical Plotting for Epoch-2

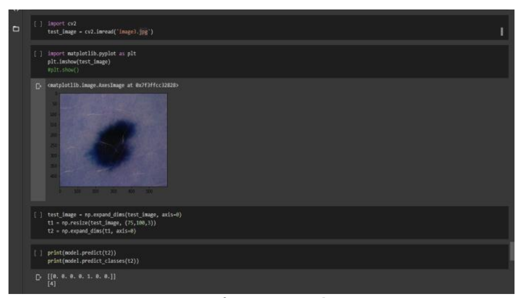

Fig.12. Testing for an Image wih Detection

TABLE I.

EPOCH-50

Sr. No.	Evaluation		
	Metric/Parameter	*Testing*	*Validation*
1.	Accuracy	93.35%	93.35%
2.	Loss	15.65%	15.65%

TABLE II.

EPOCH-2

Sr. No.	Evaluation		
	Metric/Parameter	*Testing*	*Validation*
1.	Accuracy	93.28%	93.28%
2.	Loss	16.01%	16.01%

VII. Discussion

The proposed system aims in automatic computer-based detection of skin diseases so as to reduce life risks. This has been no doubt a challenging task owing to the fine-grained variability in the appearance of skin.

VIII. Conclusion

Skin Diseases are ranked fourth most common cause of human illness, but many still do not consult doctors. We presented a robust and automated method for the diagnosis of dermatological diseases. Treatments for skin are more effective and less disfiguring when found early. We should point out that it is to replace doctors because no machine can yet replace the human input on analysis and intuition. Researches in European Society of Medical Oncology have shown for the first time that form of AI or ML is better than experienced dermatologists. In this a brief description of the system and the implementation methodology is presented.

IX. Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are heartily thankful to our internal guide Dr. Swapna Borde for noble guidance and moral support. Their valuable suggestion timely advise, inspired us towards sustain efforts for our project work.

We would like to express our gratitude towards our parents and members of Vidyavardhini's College of Engineering & Technology for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciation also goes to our colleagues in developing the project and people who have willingly helped us out with their abilities.

X. References.

[1] Shamsul Arifin, M., Golam Kibria, M., Firoze, A., Ashraful Amini, M., & Hong Yan. (2012). Dermatological disease diagnosis using color-skin images. 2012 International Conference on Machine Learning and Cybernetics. doi:10.1109/icmlc.2012.6359626).

[2] Jana, E., Subban, R., & Saraswathi, S. (2017). Research on Skin Cancer Cell Detection Using Image Processing. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). doi:10.1109/iccic.2017.8524554.

[3] Mhaske, H. R., & Phalke, D. A. (2013). Melanoma skin cancer detection and classification based on supervised and unsupervised learning. 2013 International Conference on Circuits, Controls and Communications (CCUBE). doi:10.1109/ccube.2013.6718539.

[4] Alfed, N., Khelifi, F., Bouridane, A., & Seker, H. (2015). Pigment network-based skin cancer detection. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). doi:10.1109/embc.2015.7320056.

[5] Lau, H. T., & Al-Jumaily, A. (2009). Automatically Early Detection of Skin Cancer: Study Based on Nueral Netwok Classification. 2009 International Conference of Soft Computing and Pattern Recognition. doi:10.1109/socpar.2009.80.

[6] Dubal, P., Bhatt, S., Joglekar, C., & Patii, S. (2017). Skin cancer detection and classification. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI). doi:10.1109/iceei.2017.8312419

Appendix C

Tools Used

- **Overleaf:** Overleaf is free to use. You Can Create, Edit and Share your projects with a signup method. Overleaf is a real-time editor used to research papers and projects. is a cloud-based LaTeX editor used for writing, editing, and publishing scientific documents.Overleaf can be accessed by multiple users at a time.
- **Jupyter Notebook:** Jupyter notebook is a web based application for a creating and sharing any documents. Jupyter notebook is mostly used in Python programming language-related project. Jupyter Notebook can support programming languages such as R and python.
- **Android Studio:** Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

- **Diagrams.Net:** Diagrams net is free online diagram software.It can be used for making flowchart,process diagram,DFD diagram,UML diagram and network diagram.

Appendix D

Papers Published/Certificates

- Title: Common Skin Disease Diagnosis and Prediction: A Review
- Publish date : February 2023
- Published In: IRJET journal Volume 10 Issue 02
- Link: https://www.irjet.net/volume10-issue02
- Serial No: 111
- Impact: 8.226


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
Atharva P. Gurav
In recognition the publication of the manuscript entitled
Common Skin Disease Diagnosis and Prediction: A Review
published in our Journal Volume 10 Issue 2 February 2023
Editor in Chief
E-mail : editor@irjet.net
Impact Factor : 8.226
www.irjet.net


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
Deep R. Khadke
In recognition the publication of the manuscript entitled
Common Skin Disease Diagnosis and Prediction: A Review
published in our Journal Volume 10 Issue 2 February 2023
Editor in Chief
E-mail : editor@irjet.net
Impact Factor : 8.226
www.irjet.net


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
Samruddhi M. Gaikwad
In recognition the publication of the manuscript entitled
Common Skin Disease Diagnosis and Prediction: A Review
published in our Journal Volume 10 Issue 2 February 2023
Editor in Chief
E-mail : editor@irjet.net
Impact Factor : 8.226
www.irjet.net


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
Ruchika K. Gaikwad
In recognition the publication of the manuscript entitled
Common Skin Disease Diagnosis and Prediction: A Review
published in our Journal Volume 10 Issue 2 February 2023
Editor in Chief
E-mail : editor@irjet.net
Impact Factor : 8.226
www.irjet.net