 **SKIN CANCER PREDICTION WITH**



**DEEP NEURAL NETWORK**

**PROJECT** **REPORT**

***Submitted*** ***By***

**PADMAVEERASHREE. L**

**PRASANTH. A**

**VIGNESH. R**

***in*** ***partial*** ***fulfillment*** ***for*** ***the*** ***award*** ***of*** ***the*** ***degree*** ***of***

**BACHELOR** **OF** **ENGINEERING** **IN**

**COMPUTER** **SCIENCE** **AND** **ENGINEERING**

**KNOWLEDGE** **INSTITUTE** **OF** **TECHNOLOGY,** **SALEM-637 504**

**ANNA** **UNIVERSITY::CHENNAI** **600** **025**

**APRIL-MAY** **2021**

 **SKIN CANCER PREDICTION WITH**



**DEEP NEURAL NETWORK**

**PROJECT** **REPORT**

***Submitted*** ***By***

**PADMAVEERASHREE. L**

**PRASANTH. A**

**VIGNESH. R**

***in*** ***partial*** ***fulfillment*** ***for*** ***the*** ***award*** ***of*** ***the*** ***degree*** ***of***

**BACHELOR** **OF** **ENGINEERING** **IN**

**COMPUTER** **SCIENCE** **AND** **ENGINEERING**

**KNOWLEDGE** **INSTITUTE** **OF** **TECHNOLOGY,** **SALEM-637 504**

**ANNA** **UNIVERSITY::CHENNAI** **600** **025**

**APRIL-MAY** **2021**

**ANNA** **UNIVERSITY::CHENNAI** **600** **025** **BONAFIDE** **CERTIFICATE**

Certified that this project report titled **“SKIN CANCER PREDICTION WITH DEEP NEURAL NETWORK”** is the bonafide work of **“PADMAVEERASHREE.L(611217104057), PRASANTH.A** **(611217104065), VIGNESH.R (611217104119)”** who carried out the project work under my supervision.

**SIGNATURE**

Dr. V. KUMAR M.E., Ph.D., **HEAD** **OF** **THE** **DEPARTMENT** **PROFESSOR**

Department of Computer Science and Engineering,

Knowledge Institute of Technology, Kakapalayam,

Salem - 637 504.

**SIGNATURE**

Mr. K. RAVIKUMAR M.E,

**SUPERVISOR**

**ASSOCIATE PROFESSOR**

Department of Computer Science and Engineering,

Knowledge Institute of Technology, Kakapalayam,

Salem - 637 504.

Submitted for the Project Viva-Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

---------------------------

**INTERNAL EXAMINER**

---------------------------

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

At the outset, we express our heartfelt gratitude to **GOD,** who has been our strength to bring this project to light.

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to our beloved President **Mr. C. Balakrishnan**, who has provided all the facilities to us.

We would like to convey our sincere thanks to our beloved Principal **Dr.** **PSS.** **Srinivasan,** who forward us to do our project and offers adequate duration to complete our project.

We express our sincere thanks to **Dr.** **V.** **Kumar**, Head of the Department, Department of Computer Science and Engineering, for fostering the excellent academic climate in the department.

We express our pronounced sense of thanks with deepest respect and gratitude to our internal guide **Mr. K. Ravikumar,** Associate Professor, Department of Computer Science and Engineering, for his valuable and precious guidance and for having amicable relation.

With deep sense of gratitude, we extend our earnest and sincere thanks to our project coordinators **Mr. T. Karthikeyan,** Assistant Professor, **Dr. S. Sakthivel**, Associate Professor **and** **Mrs. R. Saranya,** Assistant Professor, Department of Computer Science and Engineering for their kind guidance and encouragement during this project.

We would also like to express our thanks to all the staff members of our department, friends and students who helped us directly and indirectly in all aspects of the project work to get completed successfully.

. **TABLE** **OF** **CONTENTS**

**CHAPTER TITLE PAGE**

**NO NO**

**ABSTRACT I**

**LIST** **OF** **FIGURES** **II**

**LIST** **OF** **ABBREVIATIONS** **III**

**1** **INTRODUCTION** **1**

1.1 Data Science 1

1.2 Machine Learning 3

1.3 Deep Learning 4

1.4 Image processing & Overview 4

**2** **LITERATURE** **SURVEY** **5**

**3** **SYSTEM** **ANALYSIS** **13**

3.1 Existing System 13

3.2 Proposed System 14

**4** **SYSTEM** **SPECIFICATION** 15

4.1 Hardware Requirements 15

4.2 Software Requirement 15

4.3 Tools Requirements 15

4.4 Language Description 16

4.5 Tool Description 18

**5** **SYSTEM** **STUDY** **22**

5.1 FEASIBILITY STUDY 22 5.1.1Technical Feasibility 22

5.1.2Economic Feasibility 23

5.1.3 Behavioural Feasibility 23

**6** **SYSTEM** **DESIGN** **25**

6.1 SYSTEM ARCHITECTURE 25

**7** **MODULE** **DESCRIPTION** **27**

7.1 MODULES

7.1.1 Data Visualization 27

7.1.2 Model Definition 28

7.1.3 Training Phase 30

7.1.4 Testing Phase 30

7.1.5 Deployment 31

**8** **SYSTEM** **TESTING** **32**

8.1 TYPES OF TESTS 32

8.1.1 Unit Testing 32

8.1.2 Integration Testing 33

8.1.3 Validation Testing 33

8.1.4 Output Testing 34

8.1.5 Performance Testing 34

8.1.6 System Testing 34

8.1.7 User acceptance Testing 34

**9** **SYSTEM** **IMPLEMENTATION** **35**

**10** **CONCLUSION** **36**

10.1 Conclusion 36

10.2 Future Enhancement 36

**11** **APPENDIX** **A1**

A.1 Source Code A1

A.2 Screenshot A10

**12** **REFERENCES** **R1**

ABSTRACT

**ABSTRACT**

The medical industry is advancing with the innovation of new technologies; newer healthcare technology and treatment procedures are being developed. Biotechnology is the base for all these advancement in technology. With the advent of several pollutants, cosmetics and chemicals into our day to day lives, the health of individuals has been deteriorating every day. These not only effects physical or mental health, but also change our lifestyles. Recently, machine learning techniques have become a trend to deal with the task. Deep learning, in particular Convolutional Neural Network (CNN), have been achieving remarkable results in this field. This project work concentrates on identification of skin cancer, caused by one of the above-mentioned conditions. The main objective of our project is to make easy detection of skin cancer. For that, we are creating a deep learning model to predict whether the skin cancer present or not.

**I**

**FIGURE** **NO.**

2.1.1

2.3.1

3.2.1

6.1.1

7.1.1.2

7.1.4.2

8.1.1.1

**NAME** **OF** **FIGURE**

MOBILE NET ARCHITECTURE USED IN THE CURRENT STUDY FOR THE CLASSIFICATION OF SKIN LESION IMAGE AMONG SEVEN SKIN CANCER TYPES.

COMPONENTS OF METHODOLOGY: PRE-PROCESSING

THE WORKFLOW OF THE SYSTEM

SYSTEM ARCHITECTURE

ARCHITECTURE OF PROPOSED SYSTEM

TRAINING MODEL

PHASES OF TESTING

II**PAGE** **NO.**

7

10

14

26

29

31

33

**ABBREVIATION**

DL

ML

CNN

ResNet

NLP

SVM

ReLU

**LIST OF ABBREVIATIONS**

**EXPANSION**

DEEP LEARNING

MACHINE LEARNING

CONVOLUTION NEURAL NETWORK

RESIDUAL NETWORK

NATURAL LANGUAGE PROCESSING

SUPPORT VECTOR MACHINE

RECTIFIED LINEAR UNIT

III

INTRODUCTION

**CHAPTER** **–** **1**

**INTRODUCTION**

* 1. **DATA SCIENCE**

Data science is a multidisciplinary blend of **data inference, algorithm development, and technology** in order to solve analytically complex problems. The main part of data science is all about data. Data science is ultimately about using this data in creative ways to generate business value.

### DATA SCIENCE – DISCOVERY OF DATA INSIGHTS

This aspect of data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviors, trends, and inferences. It's about surfacing hidden insight that can help enable companies to make smarter business decisions. For example:

Netflix data mines movie viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce.

Target identifies what is major customer segments within it are base and the unique shopping behaviors within those segments, which helps to guide messaging to different market audiences.

How do data scientists mine out insights? It starts with data exploration. When given a challenging question, data scientists become detectives. They investigate leads and try to understand pattern or characteristics within the data. This requires a knowledge of analytical creativity.

**1**

Then as needed, data scientists may apply quantitative technique in order to get a level deeper – e.g. inferential models, segmentation analysis, time series forecasting, synthetic control experiments, etc. The intent is to scientifically piece together a forensic view of what the data is really saying.

This data-driven insight is central to providing strategic guidance. In this sense, data scientists act as consultants, guiding business stakeholders on how to act on findings.

### 

### 1.1.2 DATA SCIENCE – DEVELOPMENT OF PRODUCT

A "data product" is a technical asset that: It utilizes data as input and processes that data to return algorithm-generated results. The classic example of a data product is a recommendation engine, which ingests user data, and makes personalized recommendations based on that data. Here are some examples of data products:

Amazon's recommendation engines suggest items for you to buy, determined by their algorithms. Netflix recommends movies to you. Spot the recommends music to you.

Gmail's spam filter is data product – an algorithm behind the scenes processes incoming mail and determines if a message is junk or not.

Computer vision used for self-driving cars is also data product – machine learning algorithms are able to recognize traffic lights, other cars on the road, pedestrians, etc. This is different from the "data insights" section above, where the outcome to that is to perhaps provide advice to an executive to make a smarter business decision. In contrast, a data product is based on the technical functionality that encapsulates an algorithm, and is designed to integrate directly into core applications.

**2**

Respective examples of applications that incorporate data product behind the scenes: Amazon's homepage, Gmail's inbox, and autonomous driving software.

Data scientists play a central role in developing data product. This involves building out algorithms, as well as testing, refinement, and technical deployment into production systems. In this sense, data scientists serve as technical developers, building assets that can be leveraged at wide scale.

### 1.2 MACHINE LEARNING

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

**3**

**1.3 DEEP LEARNING**

Deep learning (DL) is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

**1.4** **IMAGE** **PROCESSING & OVERVIEW**

Image processing is a way to perform such procedures on a photo, to produce an optimized image or to retrieve useful information from the image. It is a form of processing of the signal in which the input is in the format of an image and the output can be an image or features of the image. Image processing nowadays is one of the fast-growing technologies. It is also a core field of research in engineering and informatics.

The foregoing are the three stages of the image processing process:

• Image extraction by way of photo retrieval tools;

• Photo review and manipulation;

• Production where an image or video produced on an image analysis can be changed.

**4**

Our project aims to predict if the person is affected by skin cancer or not. The prediction is done by using deep learning algorithms. We use ResNet152V2 pretrained model to predict.

**5**

LITERATURE SURVEY

**CHAPTER** **–** **2**

**LITERATURE** **REVIEW**

**2.1 SKIN LESION ANALYZER: AN EFFICIENT SEVEN-WAY MULTICLASS SKIN CANCER CLASSIFICATION USING MOBILENET [Saket S. Chaturvedi, Kajol Gupta1, and Prakash S. Prasad]**

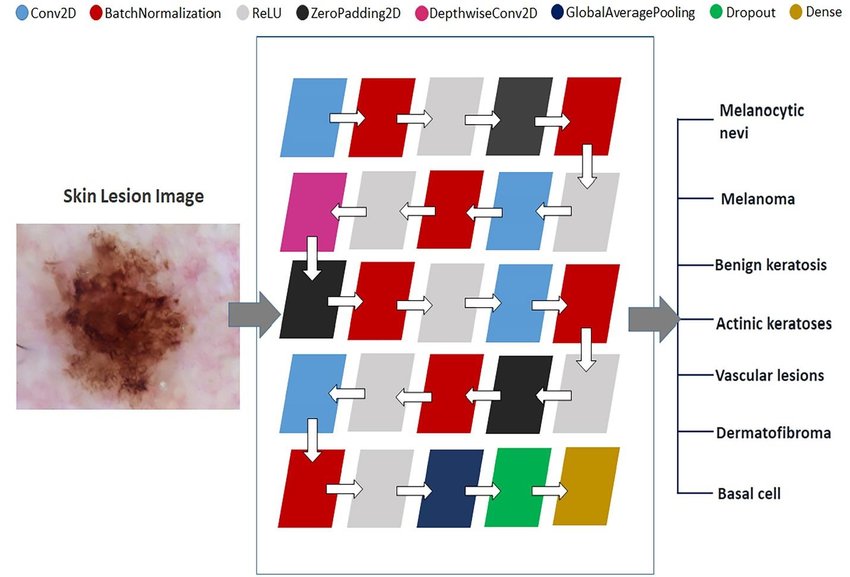
Skin Cancer is an emerging global health problem considering the increasing prevalence of harmful ultraviolet rays in the earth’s environment. The researchers had discovered a further 10 percent depletion of the ozone layer will intensify the problem of skin cancer with an additional 300,000 non-melanoma and 4,500 melanoma cases each year. Currently, every year 123,000 melanomas and 30,00,000 non-melanoma cases are recorded worldwide. The recent study on the prevention of skin cancer reports 90 percent of non-melanoma and 86 percent of melanoma cases induced by excessive exposure of ultraviolet rays. The UV radiation detriments the DNA present at the inner layers of skin, triggering the uncontrolled growth of skin cells, which may even emerge as a skin cancer.

We utilized a MobileNet convolutional neural network pretrained on approximately 12,80,000 images from 2014 ImageNet Challenge and finetuned on HAM10000 dataset which contain 10015 dermoscopy images employing transfer learning . The MobileNet model classified skin lesion image with performance better or comparable to expert dermatologists for seven classes. We also conducted data-analysis on the dermoscopy images of skin cancer from HAM10000 dataset to uncover the relation of skin cancer with several parameters to strengthen the understanding of skin cancer.

The pre-processing of skin lesion images was done by using Keras ImageDataGenerator[1]. The 57 null Age entries in the dataset were filled using the 12 mean filling method. The Dermoscopy images in the dataset were downscaled to 224X224 pixel resolution from 600X450 pixel resolution to make images compatible with the MobileNet model.

**6**

The 10015 images in the dataset were split into the training set (9077 images) and validation set (938 images). The dataset images with no duplication in training data were selected for the validation set so that the authenticity in the validation process can be maintained.



**FIG** **2.1.1** **MOBILE NET ARCHITECTURE USED IN THE CURRENT STUDY FOR THE CLASSIFICATION OF SKIN LESION IMAGE AMONG SEVEN SKIN CANCER TYPES.**

The MobileNet model is ideal for mobile and embedded vision applications as they have lightweight DNN architecture. We used MobileNet convolutional neural network pretrained on 12,80,000 images containing 1,000 object classes from the 2014 ImageNet Challenge. The 25 layered MobileNet architecture was constructed for the current study, which employs four Conv2D layers, seven Batch Normalization layers, seven ReLU layers[1], three ZeroPadding2D layers, and single DepthwiseConv2D, Global Average Pooling, Dropout, and Dense layers. The training of the model was done on a training set of 38,569 images using Transfer Learning with batch size and epochs as 10 and 50 respectively.

**7**

**2.2 SKIN CANCER PREDICTION USING IMAGE PROCESSING AND DEEP LEARNING [PROF. SHASHANK BHOLANE1, SHUBHAM PATIL2, GAURAV RAJPUT3, SWAPNIL PATIL4, SANKET GUNJALKAR @2020]**

Skin cancer is nothing however the abnormal growth of skin cells — most frequently develops on skin exposed to the sun. however, this common type of cancer may occur on areas of your skin not normally exposed to daylight. There are 3 major varieties of carcinoma basal cell malignant neoplastic disease, epithelial cell malignant neoplastic disease and skin cancer. You’ll scale back your risk of carcinoma by limiting or avoiding exposure to ultraviolet (UV) radiation[2]. Observing the skin for dubious changes will facilitate discover carcinoma at its earliest stages. Early detection of carcinoma offers you the best likelihood for fortunate carcinoma treatment.

The combination of visual inspection and dermoscopic images ultimately results in an absolute melanoma detection accuracy of 75-84 percent by dermatologists. CNN methods used to classify skin cancers are presented. A basic requirement for the successful training of deep CNN models is that sufficient training data labelled with the classes are available.

Various methods of every stage is properly stated in this paper. Because of the data constraints algorithms like SVM, Bayesian Classifiers or other classification techniques are not very effective.

A. Melanoma

B. Basal Cell Carcinoma

C. Squamous Cell Carcinoma

D. Merkel Cell Cancer

They are the various strategies for the identification and classification of skin cancer moles like convolutional neural network, transfer learning etc. Discussed the basic concept of skin cancer detection and various types of skin cancer.

**2.3 MELANOMA SKIN CANCER DETECTION USING IMAGE PROCESSING AND MACHINE LEARNING [VIJAYALAKSHMI M M @2019]**

Methodology Our model is designed in 3 phases as follows: A. Phase1 – the first model involves collection of dataset, the images are collected from ISIC dataset (International Skin Imaging Collaboration) Phase 1 also involves the pre-processing of the images where hair removal, glare removal and shading removal are done B. Removal of these parameters helps us to identify the texture, color, size and shape like parameters in an efficient way. C. Phase2- this phase consists of the segmentation and feature extraction, segmentation is explored via three methods a. Otsu segmentation method b. Modified Otsu segmentation method c. water shed segmentation method. Feature are extracted for color, shape, size and texture. D. Phase 3- this is the most important phase of our model, this phase involves designing of the model and training. Our model was trained for Back Propagation Algorithm (Neural Networks), SVM (Support Vector Machine), and CNN (Convolutional Neural Networks) on the dataset that was collected in the phase1, the model after training was tested for the accurate output.

In the neural Networks we have used the Back Propagation Algorithm. The Back Propagation is a supervised learning algorithm, for training the multi-layer perceptron’s. while designing the neural networks we initialize the weights with some random values as we do not know what exactly the weight can be, so we first give some random weight if the model provides an error with large values. so, we need to need to change the values to somehow minimize the error value. To generalize this, we can just say

Calculate the error – How far is your model output from the actual output

**9**

Minimum Error – Check whether the error is minimized or not.

Update the parameters – If the error is huge then, update the parameters (weights and biases). After that again check the error. Repeat the process until the error becomes minimum.

Model is ready to make a prediction [3]– Once the error becomes minimum, you can feed some inputs to your model and it will produce the output Designing the Model: In our model we have used 3 different methods i.e. Neural Networks, Support Vector Machine and Convolutional Neural Networks to find the efficient detection and classification of the melanoma skin cancer into Malignant and benign skin cancers. The data that is pre-processed is followed by segmentation and feature extraction these extracted feature images are then passed into Neural Networks and Support Vector Machine to classify the images into malignant and benign and to predict the exact accuracy.

Segmentation

Feature extraction

Feature vectorization

Pre-processing

Load pre-trained /created classifier

Load classifier

Database

**FIG** **2.3.1 COMPONENTS OF METHODOLOGY: PRE-PROCESSING**

**10**

CNNs are neural networks with a specific architecture that have been shown to be very powerful in areas such as image recognition and classification. CNNs have been demonstrated to identify faces, objects, and traffic signs better than humans and therefore can be found in robots and self-driving cars. CNNs are a supervised learning method and are therefore trained using data labeled with the respective classes. Essentially, CNNs learn the relationship between the input objects and the class labels and comprise two components: the hidden layers in which the features are extracted and, at the end of the processing, the fully connected layers that are used for the actual classification task. Unlike regular neural networks, the hidden layers of a CNN have a specific architecture.

**2.4 SKIN CANCER DETECTION USING MACHINE LEARNING ALGORITHM [SANJANA M 1 , DR. V. HANUMAN KUMAR @2018]**

Image handling systems give a productive instrument to characterize the malignancy from the images. Different creators have utilized diverse approaches to consolidate these innovations to accomplish better end. Different works have been done in the location of Skin malignancy utilizing picture handling and mix of neural system. Paper proposes image segmentation scheme based on an algorithm, Support vector Machine (SVM) and snake active counter[4]. To help finding the appropriate parameters for snake algorithm, SVM is used. The initial curve and the snake parameters must be chosen appropriately to perform the snake algorithm efficiently. Hence, to choose the appropriate initial curve and parameters we apply Support vector Machine (SVM). The initial curve is predicted to be of the following shapes: curve, eclipse and rectangle. To reduce the complexity in the implementation of SVM model, and to keep up the SVM implementation without any degradation, these shapes are chosen. Based on the results of the SVM, the attributes can fit in the image. The images used as testing set were chosen as the use for template circle and to determine the accuracy of the edges.

**11**

The experimental results show that the Snake algorithm is the basis to determine the edge, in comparison to the edge found by the experts.

The segmentation technique applied by the algorithm found the edges, which are closer to the results determine by the experts. But the undesired areas are also segmented. To obtain accurate results of segmentation, better classification or the refinements of the algorithm is required.

There are various advantages of utilizing relapse investigation. They are as per the following:

• It shows the noteworthy connections between ward variable and free factor.

• It shows the quality of effect of numerous autonomous factors on a reliant variable.

**12**

 SYSTEM ANALYSIS

**CHAPTER** **–** **3**

**SYSTEM** **ANALYSIS**

**3.1** **EXISTING** **SYSTEM**

The existing system uses Deep Neural network and transfer learning to predict the skin cancer in an image. The system used pre-trained model MobileNet to predict the skin cancer. The MobileNet model is ideal for mobile and embedded vision applications as they have lightweight DNN architecture. The MobileNet model consist was trained on 12,80,000 images containing 1,000 object classes from the 2014 ImageNet Challenge. The dataset was obtained from HAM1000 database. The dataset consists of a total of 10015 dermoscopy images, which includes 6705 Melanocytic nevi images, 1113 Melanoma images, 1099 Benign keratosis images, 514 Basal cell carcinoma images, 327 Actinic keratosis images, 142 Vascular images and 115 Dermatofibroma images with 600X450 pixels resolution which will be downscaled to 224X224 pixels resolution for mobilenet image compatibility for training and testing the data.

HAM10000 dataset has an unbalance distribution of images among the seven classes. Data Augmentation brings an opportunity to rebalance the classes in the dataset, alleviating other minority classes. Data Augmentation is an effective means to expand the size of training data by randomly modifying several parameters of training data images like rotation range, zoom range, horizontal and vertical flip, fill\_mode, etc. We conducted data augmentation of minority classes in the dataset: Melanoma, Benign Keratosis, Basal Cell Carcinoma, Actinic Keratosis, vascular lesion, and dermatofibroma to generate approximately 6000 images in each class giving a total of 38,569 images in the training set.

The final model will contain the MobileNet model. The output of the model will be given to a convolutional neural network and a set of Batch normalization and Dropout is applied on the model.

**13**

The output will be finally obtained from the output layer which is a softmax function. The trained model was then deployed as a web application.

**PROBLEM IDENTIFICATION**

* The average accuracy of the model was 83%. So, there’s a need for improving the accuracy.
* Usage of HAM1000 dataset requires downscaling the resolution of the image dataset. So, there’s a need for another dataset which does not require downscaling.

**3.2** **PROPOSED** **SYSTEM**

This system aims to reduce the manual work in getting information from the images. It uses neural networks to gain insights from the image data on its own and segments the image into multiple slices and gives them as input to the neural network. The workflow of the system is shown below

GATHERING DATA

LOADING THE DATA

MODEL CREATION

PREPROCESSING

TRAINING & TESTING

SAVE THE MODEL

**FIG** **3.2.1** **THE** **WORKFLOW** **OF** **THE** **SYSTEM**

**14**

`

SYSTEM SPECIFICATION

**CHAPTER** **–** **4**

**SYSTEM** **SPECIFICATIONS**

**4.1** **HARDWARE** **REQUIREMENTS**

Processer

RAM

Hard Disk

: Intel i3-5005U

: 2 GB

: 1 GB

**4.2** **SOFTWARE** **REQUIREMENTS**

Operating System

Technology

Language

Platform used

: Windows 10

: Deep Learning

: Python

: Google Colab – Environment, Browser, Text editor

**4.3 TOOLS REQUIREMENT**

Software Applications : Jupyter Notebook

Environment : Google Colab

**15**

**4.4** **LANGUAGE** **DESCRIPTION**

Python is a modular language, supporting multiple paradigms of programming such as object and functional programming while having a robust library. Python encourages modularity and reuse of code. Most syntax in Python is derived from C since Python is written in C.

However, Python indents code for structuring code blocks. Python is an

interpreter that translates the text that you type in a language that can

be interpreted by the processor of your machine. Python is easy to learn and execute. It is an open-source language.

Python is a scripting language, such as Ruby and Perl, which is often used for web apps and interactive material. Python can be used to build a variety of applications such as Web applications, Graphical UI applications, software development apps, physics and mathematical apps, network design, gaming and 3D apps and other business applications.

The most rapid way to debug is by inserting some print commands into the code. This simple approach is the quick edit-test-debug loop. The interpreter can print the pile trace if the exception is not found. Python is Cross-Platform. Compilation of program is not needed, this is PERL-like. And PHP Python is collaborative − A Python prompt to write the programs to the interpreter.

**PANDAS**

***Pandas***is an open source, BSD-licensed library providing high- performance, easy-to-use data structures and data analysis tools for the Python programming language. Python has long been great for data mugging and preparation, but less so for data analysis and modelling. *Panda* helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

**16**

**NUMPY**

It is a package for numbers, it is a library that contains strong multi-dimensional array objects used to store homogenous data. This is useful in linear algebra, random, etc. NumPy is a Python extension framework. It offers fast and effective operations with uniform data arrays.

**MATPLOTLIB**

Matplotlib is an impressive library of 2D plots. Based on NumPy arrays,

Matplotlib is a cross-platform data visualization library. It is composed of several plots such as line, bar, histogram, etc. Matplotlib is a Python programming language plot library and its NumPy extension of numeric mathematics. It offers an object-based API to incorporate plots into programs using GUI toolboxes for general purposes such as Tkinter, wxPython, Qt or GTK+.

**KERAS**

Keras is an open-access library based on Python and is a high-grade API for Neural Networks. It's the best way to combine easily with R, Theano and so on. This enables deep neural networks for accelerated exploration and is user-friendly, flexible and adaptable. It supports CNN's and RNN's and works on CPUs or GPUs seamlessly.

**TENSORFLOW**

Tensor flow is an open source software library for numerical computation using data-flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. It [reached version 1.0](https://research.googleblog.com/2017/02/announcing-tensorflow-10.html)  in February 2017, and has continued rapid development, with 21,000+ commits thus far, many from outside contributors.

**17**

This article introduces TensorFlow, its open source community and ecosystem, and highlights some interesting TensorFlow open sourced models.

TensorFlow is cross-platform. It runs on nearly everything: GPUs and CPUs - including mobile and embedded platforms—and even tensor processing units (TPUs), which are specialized hardware to do tensor math on. They aren't widely available yet, but we have recently launched an alpha program.

**HTML**

Hypertext Markup Language (HTML) is the standard [markup](https://en.wikipedia.org/wiki/Markup_language) [language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser). It can be assisted by technologies such as [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting](https://en.wikipedia.org/wiki/Scripting_language) [languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript). HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items.

**4.5** **TOOL** **DESCRIPTION**

Jupyter Notebook is an open-source software application for developing and distributing a live file, a visualization, and narrative text paper. Data purification and processing, calculation of quantities, mathematical modeling, computer analysis, software testing, etc.

Jupyter Notebook is in the IPython software spin-off and has a Notebook Campaign for IPython**.** The system is operating in tandem with the top of the screen, writing, browsing, loading, shifting, button, widgets.

**18**

**Notebook** **Documents**

All material, involves code inputs and outputs, informative language, mathematics, photographs and rich media representations of objects, which can be included in the Web application. In one single text, a Notebook combines code and its performance. Notebook notes providing a summary of the research and results in the data processing in real-time, both rich and code components, connections, code-mix equation, and text items.

 **Code** **cells**

Functional computer input and output in the kernel

 **Markdown** **cells**

Embedded LaTeX calculations story text

 **Raw** **cells:**

The unformatted text used, without alteration, like notebooks with nbconvert are translated to different formats

**Structure** **of** **a** **Notebook** **Document**

The notebook uses a cell chain. The Shift-Entre or the Play or Cell Run toolbar button on the menu bar may be used to perform a cell which is a multi-line input field for texts. The efficiency of a cell depends on the nature of the cell. There are three cell types: coding cells, labeling cells, and simple cells. The cell starts becoming a computer unit, but the code may be modified through a toolbar dropdown (which will initially be "Text") or keyboard shortcuts.

**Plotting**

A main feature of the Jupyter notebook is the ability to imagine images that generate running codes. That is achieved by the IPython kernel, which has been built to operate together with the matplotlib plotting library. The kernel function is the basic incorporation of the plotting library.

**19**

**Browser** **Compatibility**

The Jupyter Notebookaimstosupportthe latestversionsof thesebrowsers:

* Chrome
* Safari
* Firefox

The Jupyter Notebook is built to help the up-to-date Opera and Edge models, but if they don't, please use the one plugin that they do. Safari with the API is not supposed to operate with an untrusted credential (Web-Sockets do not operate).

**Structure** **of** **a** **Notebook** **Document**

The notebook uses a cell chain. The Shift-Entre or the Play or Cell Run toolbar button on the menu bar may be used to perform a cell which is a multi-line input field for texts. The efficiency of a cell depends on the nature of the cell. There are three cell types: coding cells, labeling cells, and simple cells. The cell starts becoming a computer unit, but the code may be modified through a toolbar dropdown (which will initially be "Text") or keyboard shortcuts.

**Plotting**

A main feature of the Jupyter notebook is the ability to imagine images that generate running codes. That is achieved by the IPython kernel, which has been built to operate together with the matplotlib plotting library. The kernel function is the basic incorporation of the plotting library.

**GOOGLE COLAB**

Colaboratory   is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

**20**

### FLASK

Flask is a lightweight web frame of Python. It provides the user with libraries, modules and tools to help build Web-Applications such as a blog or wiki. Flask, unlike Django does not depend on other libraries and is hence termed as a micro framework. Flask is considered more [Pythonic](http://blog.startifact.com/posts/older/what-is-pythonic.html) than the [Django](https://www.fullstackpython.com/django.html) web framework because in common situations the equivalent Flask web application is more explicit. Flask provides the most flexibility in terms of customization. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running. Flask was also written several years after Django and therefore learned from the Python community's reactions as the framework evolved.

Jokull Solberg wrote a great piece articulating to this effect in his experience switching between Flask and Django. Flask application, it doesn't do anything interesting. One interesting thing a web application can do is persist user data, but it needs the help of and connection to a database. Flask is very much a "do it yourself" web framework

**21**

 SYSTEM STUDY

**CHAPTER-5**

**SYSTEM** **STUDY**

**5.1** **FEASIBILITY** **STUDY**

Feasibility study is the evaluation of system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Months or years of effort, thousands and millions of dollars, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition of phase.

Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. During product engineering, however, we concentrate our attention on primary areas of interest.

In this project, there are three key considerations involved in the feasibility analysis are

* + - Technical Feasibility
    - Economical Feasibility
    - Behavioral Feasibility

**5.1.1** **TECHNICAL FEASIBILITY**

Technical feasibility is the need of hardware and software, which are needed to implement the proposed system in the organization. Technical requirements are to be fulfilled to make the proposed system work. This should be necessarily predetermined so as to make the system more competent.

**22**

Technical feasibility is the most difficult area to assess at the stage of the system development process. Because objectives, functions and performance are somewhat hazy, anything seems possible if the right assumptions are made.

In this project, the hardware and software that we use are open source and provide flexibility and agility for enterprise. Python has clean object- oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity.

**5.1.2** **ECONOMICAL FEASIBILITY**

Economical feasibility deals with the analysis of cost against benefits i.e. whether the benefits to be enjoyed due to the new system are worthy, when compared to the costs to be spent on the system.

Economic justification is generally the “bottom-line” consideration for most system, long-term corporate income strategies, impact on other profit centre’s or products, cost of the resources needed for development, and potential market growth. This project is based on web and resources available in web. Using resources from the web does not indulge much in cost.

Hence this project was economically justified for development in this organization. Especially in the present scenario, where the objective is towards compatibility, reduced cost is weighed against the ultimate income or benefit derived from the developed.

**5.1.3** **BEHAVIOURAL FEASIBILITY**

Behavioral feasibility speaks about how a strong reaction the programmer is likely to have toward or against the development of the system.

**23**

In this project, the programmers work on Python, which helps the programmers to do coding in fewer steps as compared to Java or C++. It has a comprehensive and large standard library that has automatic memory management and dynamic features.

The language has clean object-oriented designs that increase two to tenfold of programmer’s productivity while using the languages like Java, VB, Perl, C, C++ and C#. The programmers of big companies use Python as it has created a mark for itself in the software development. Since the programmers are well exposed to the system, it will be feasible for them to work on. Therefore, the system to be computerized is also behaviorally feasible.

**24**

 SYSTEM DESIGN

**CHAPTER** **–** **6**

**SYSTEM** **DESIGN**

System architecture is the process that satisfies certain specifications

across a collection of device specifications identified to a specification. The method fills the distance between the issue region and the current structure in a safe manner. In the cycle the design of the device is split into many smaller sub-activities, operating together to accomplish the key purpose of the program.

**6.1** **SYSTEM** **ARCHITECTURE**

The proposed system is based on the image data that is used. The image data is from kaggle (Skin cancer: Malignant vs Benign images). The images are visualized to get an idea about their attributes. Finally, the images are made into patches and separated as training and testing set and finally fed into the model. It is a neural network based on CNN’s and the work flow contains the following modules:

1. Data Visualization

2. Model Definition

3. Training Phase

4. Testing Phase

Data collection is the process of measuring and gathering the data, or any variables in a systemized and well-established manner. It enables the collector to answer or test hypothesis and evaluate the outcomes of the collection.

**25**

SKIN IMAGE DATASET

GENERATING DATA PACHES

CREATING NEURAL NETWORK

MODEL TRAINING

MODEL TESTING

(PREDICTION)

**FIG** **–** **6.1.1** **SYSTEM** **ARCHITECTURE**

**26**

MODULE DESCRIPTION

**CHAPTER** **–** **7**

**MODULE** **DESCRIPTION**

A module is a set of source files, which provides settings that allow developers to break the project into distinct functional units. It makes the job simple to break down a project into modules and also helps us understand the essential role of eachmodule. Four Modules are described for theimplementation of the proposed framework.

**7.1** **MODULES**

**1.** Data visualization

**2.** Model definition

**3.** Training phase

**4.** Testing phase

**5.**Deployement

**7.1.1 DATA** **VISUALIZATION**

This module helps us in gaining valuable information such as the format of the data, dimensions of the image, number of slices present in each image. The observations that are got from this module play a vital role in the Modeldefinition phase. The library used for visualization is “Matplotlib”.

Once the images were loaded, data augmentation will be performed on the dataset to increase the number of data and the accuracy of the model by zooming in and out, rotating, skewing, flipping the images.

**OBSERVATIONS:**

|  |  |
| --- | --- |
| FORMAT | PNG |
| DIMENSIONS | 224X224 PIXEL |

**27**

**7.1.2** **MODEL** **DEFINITION**

This model is based on a Two-Path CNN, Cascaded CNN’s. These CNN’s are used to reduce the image into patches and finally make the model learn from them.

The Two-Path CNN consists of 9 Convolution Layers and contains 8,93,989 trainable parameters. The output shape of this CNN is (33,33,4). This Architecture is embedded within an Input Cascade CNN, MFC Cascade CNN which has an output of (65,65,4), (53,53,4) and has a total of 36 Convolution Layers in it. With a total of 20,87,454 parameters in the Cascade CNN, it is used with the previous Two-Path CNN. And we used ResNet152v2 pretrained model.

**ResNet152v2**

This is an Keras implementation of ResNet-152 with ImageNet pre-trained weights. The implementation supports both Thaenos  and TensorFlow backends

**28**

----

CNN &Pooling layers

ResNet15v2

Batch norm

Batch norm

**…..**

**Input**

**output**

…

…

…

…

…

…

…

…

**1024 neurons 512 neurons 128 neurons 4 neurons**

**FIG** **7.1.1.2 ARCHITECTURE OF PROPOSED SYSTEM**

**29**

**7.1.3** **TRAINING** **PHASE**

In this phase, the model is trained using the SKIN CANCER IMAGE Dataset. The model is run in Google Colab. The model is trained for only 3 patient's data since the dataset is very large. The Training follows the following steps

1. The Images are converted to Numpy Arrays.

2. The Arrays are given to the Neural Network.

3. Initial Phase is started, Weights for all the nodes are allocated.

4. The training continues for the next set of data, the weights of the nodes are altered based on the new and old inputs.

5. After all the training data is exhausted the weights are aggregated and saved to the tensors.

**7.1.4** **TESTING** **PHASE**

In this phase, the model is tested for its accuracy. The model is giving an accuracy of an average of 85% for testing images. The mode at present can only give the prediction of the cancer and not the segmented image.

1. The calculated node weights are used for prediction.
2. The accuracy is caluclated using the inbuilt function.

3. If the score is very less then, training is done once more to get good score.

**30**



**FIG** **7.1.4.2** **TRAINING MODEL**

**7.1.5 DEPLOYMENT**

**Deployment** is the method by which you integrate a **machine learning** model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the **machine learning** life cycle and can be one of the most cumbersome. In our project, we deploy our deep learning model into a web page. By using flask we converted our model into web application. By uploading the image, the model can predict that the user having skin cancer or not.

**31**

 SYSTEM TESTING

**CHAPTER** **–** **8**

**SYSTEM** **TESTING**

The process of testing an integrated hardware and software system is an implementation process that helps ensure that the system works correctly and consistently before the start of a live operation. Testing is crucial to the success of the system. System Testing is a logical assumption that the goal would be accomplished if all parts of the device are correct.

**8.1** **TYPES** **OF** **TESTS**

* + - Unit Testing
    - Integration Testing
    - Validation Testing
    - Output Testing
    - Performance Testing
    - System Testing
    - User acceptance Testing

**8.1.1** **UNIT** **TESTING:**

Unit testing is a productivity tool for measuring individual operating systems. The goal is to evaluate the performance of each control machine. For the smallest part of any system, an element may be evaluated. Inputs and outputs are typically one or two. A single program, process, system, etc. may be a unit for procedural programming. For analytics, the smallest kind of test structures, triggers, stubs, and false artifacts may be used for in-unit tests that may be a simple / hood, an abstraction level, or an extract/kid's class.

**32**

Unit



Unit

User Acceptance Validation

Output

User Acceptance Validation

Output

**FIG** **8.1.1.1** **PHASES** **OF** **TESTING**

**8.1.2 INTEGRATION** **TESTING:**

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets are conducted. The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step, all the errors uncovered are corrected for the next testing steps.

**8.1.3** **VALIDATION** **TESTING:**

Validation ensures that the checked and assembled program follows consumer and model requirements. The logic of system specifications should also be closely monitored. A program will check all essential functions. The whole program, including the document, should be reviewed. There are available methods for functional and non-functional testing. The norm is carefully monitored. It monitors the app habits of the consumer.

**33**

**8.1.4** **OUTPUT** **TESTING:**

* Determine automated functions to be done.
* Based on function parameters, provide data input.
* Assess performance under mission parameters.
* The test case was working. Go forward.
* Equate results that are actual and anticipated.

**8.1.5** **PERFORMANCE** **TESTING**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. It requires both hardware and software instrumentation. It is often necessary to measure resource utilization in an exacting fashion.

**8.1.6** **SYSTEM** **TESTING**

A system testing does not test the software but rather the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specifications, and system documentation.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

**8.1.7** **USER** **ACCEPTANCE** **TESTING**

The User Acceptance Test (UAT) is defined as checking the user or device to obtain whether or not it is approved by the program or not. This will be the last test after completion of the functional system and regression tests. This check serves primarily to verify the program in compliance with the business requirements. The end-user who are conscious of the business requirements carry out this confirmation.

**34**

 SYSTEM IMPLEMENTATION

**CHAPTER** **-** **9**

**SYSTEM** **IMPLEMENTATION**

System implementation is the important stage of project when the theoretical design is tuned into practical system. After proper testing and validation, system implementation should be done. System implementation includes all those activities that take place to convert an old system to the new one. The new system may be different. Replacing an existing manual or automated system may be a major modification to an existing system.

The implementation of our proposed solution begins with launching of Jupyter notebook leads to importing of certain necessary packages such as pandas, numpy, keras, Tensorflow etc.

The two-way architecture of CNN applies to local and global functionality. The six-execution stage involves data collection, data analysis, preprocessing, model creation, training, testing. Using CNN max-pooling and also by using ResNet15V2 pretrained model improves training and test speed. To be efficient in an implementation phase a lot of activities have to be performed across various departments.

After importing all the packages, various machine learning is implemented for identifying an algorithm with high accuracy. The algorithm which is found to be more accurate is embedded with GUI (Graphical User Interface) backend for database connectivity.

**35**

 CONCLUSION

**CHAPTER** **-** **10**

**CONCLUSION**

**10.1** **CONCLUSION**

We also provided an automated augmentation of skin images dependent on deep neural networks in this article.

We looked at various architectures and analyzed their efficiency effects.

The new two-track design (which can model both the local information and the global environment) is used for high efficiency and the modeling of local mark dependencies through the stacking of two CNNs and ResNet15V2 model.

Learning is focused on a two-step process which, when the label distribution becomes unbalanced, helps one to train CNNs efficiently.

The resulting segmentation method is very quick and 90 percent effective.

**10.2** **FUTURE** **ENHANCEMENT**

In future research it is important to better model prior information by more suitable geometric modeling and closeness which codes classification errors in those areas more accurately.

**36**

 APPENDIX

**CHAPTER** **–** **11**

**APPENDIX**

**A.1 SOURCE CODE**

**MODEL.IPYNB**

**IMPORTING THE PACKAGES**

import tensorflow as tf

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

**SET THE TRAINING AND TESTING DATA LOCATION**

train\_data\_path='skin-cancer-dataset/train'

test\_data\_path='skin-cancer-dataset/test'

**CREATING IMAGE DATA GENERATOR**

def get\_image\_data\_generator(path):

    data\_generator=tf.keras.preprocessing.image.ImageDataGenerator(

        rescale=1.0/255,

        rotation\_range=40,

        width\_shift\_range=0.2,

        height\_shift\_range=0.2,

        shear\_range=0.2,

        zoom\_range=0.4,

        horizontal\_flip=True,

        fill\_mode='nearest'

    )

    generator=data\_generator.flow\_from\_directory(

        path,

        target\_size=(640,480),

        batch\_size=32,

        class\_mode="binary"

    )

    return generator

**LOAD THE TRAINING DATA**

train\_datagen=get\_image\_data\_generator(train\_data\_path)

**A1**

**LOAD THE TESTING DATA**

test\_datagen=get\_image\_data\_generator(test\_data\_path)

**CREATING THE MODEL**

def get\_model():

    # Model

    pretrained=tf.keras.applications.ResNet152V2(weights='imagenet',input\_shape=(224,224,3),include\_top=False)

    for layer in pretrained.layers:

        pretrained.trainable=False

    model=pretrained.output

    model=tf.keras.layers.Conv2D(16, (2, 2),activation='relu')(model)

    model=tf.keras.layers.MaxPooling2D((2, 2))(model)

    model=tf.keras.layers.Conv2D(32, (2, 2), activation='relu')(model)

    model=tf.keras.layers.MaxPooling2D((2, 2))(model)

    model=tf.keras.layers.GlobalAveragePooling2D()(model)

    model=tf.keras.layers.Dense(1024,activation='relu')(model)

    model=tf.keras.layers.BatchNormalization()(model)

    model=tf.keras.layers.Dense(512,activation='relu')(model)

    model=tf.keras.layers.BatchNormalization()(model)

    model=tf.keras.layers.Dense(128,activation='relu')(model)

    model=tf.keras.layers.Dense(32,activation='relu')(model)

    model=tf.keras.layers.Dense(8,activation='relu')(model)

    model=tf.keras.layers.Dense(4,activation='relu')(model)

    model=tf.keras.layers.Dense(1,activation='sigmoid')(model)

    model=tf.keras.models.Model(inputs=pretrained.input,outputs=model)

    return model

model=get\_model()

**GET THE SUMMARY OF THE MODEL**

model.summary()

**COMPILE THE MODEL**

model.compile(

    optimizer=tf.keras.optimizers.RMSprop(lr=0.0001),

    loss='binary\_crossentropy',

    metrics=['acc']

**) A2**

**TRAINING THE MODEL**

model\_training=model.fit\_generator(

    train\_datagen,

    steps\_per\_epoch=64,

    epochs=15,

    validation\_data=test\_datagen,

    validation\_steps=16

)

**SAVE THE TRAINED MODEL**

tf.keras.models.save\_model(model,"drive/MyDrive/model/SkinCancer\_Model")

**DATA VISUALIZATION**

epochs=list(range(1,16))

plt.title("Accuracy")

plt.xlabel("EPOCHS")

plt.ylabel("Accuracy")

plt.plot(epochs,history.history['accuracy'])

plt.show()

plt.title("Validation Accuracy")

plt.xlabel("EPOCHS")

plt.ylabel("Validation Accuracy")

plt.plot(epochs,history.history['val\_accuracy'])

plt.show()

plt.title("loss")

plt.xlabel("EPOCHS")

plt.ylabel("Loss")

plt.plot(epochs,history.history['loss'])

plt.show()

plt.title("Validation Loss")

plt.xlabel("EPOCHS")

plt.ylabel("Validation Loss")

plt.plot(epochs,history.history['val\_loss'])

plt.show()

**A3**

**PREDICTOR.PY**

import tensorflow as tf

model=tf.keras.models.load\_model("Models/SkinCancer\_Model")

def predict(image):

img\_path="static/Images/"+image

img = tf.keras.preprocessing.image.load\_img(

img\_path, target\_size=(224, 224)

)

img\_array = tf.keras.preprocessing.image.img\_to\_array(img)

img\_array = tf.expand\_dims(img\_array, 0)

if(model.predict(img\_array)[0][0] == 0):

return "Result: Negative"

else:

return "Result: Positive"

**A4**

**BACKEND.PY**

from flask import Flask, render\_template, request

from flask\_cors import CORS

from flask\_uploads import UploadSet, IMAGES,configure\_uploads

from Predictor import predict

app=Flask(\_\_name\_\_)

CORS(app)

photos=UploadSet("photos",IMAGES)

app.config['UPLOADED\_PHOTOS\_DEST']="static/Images"

configure\_uploads(app,photos)

@app.route("/",methods=['GET','POST'])

def index():

return render\_template("index.html")

@app.route('/upload', methods=['GET', 'POST'])

def upload():

if request.method == 'POST' and 'photo' in request.files:

filename = photos.save(request.files['photo'])

res=predict(filename)

return render\_template("index.html",result=res)

return render\_template('index.html')

if(\_\_name\_\_=="\_\_main\_\_"):

app.run(port=5500,debug=True)

**A5**

**INDEX.HTML**

<!DOCTYPE html>

<html>

<head>

<title>Breast Cancer Predictor</title>

<link rel="stylesheet" href={{url\_for('static',filename='styles/styles.css')}} />

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css2?family=Patua+One&display=swap" rel="stylesheet">

</head>

<body>

<p class="title">Skin Cancer Predictor</p><hr />

<form action="upload" enctype="multipart/form-data" method="POST">

<center>

<div class="upload-btn-wrapper">

<button class="btn">Upload the Image</button>

<input type="file" name="photo" />

</div> <br/>

<input type="submit" name="submit" class="submit-btn">

</center>

</form>

<h2 class="result">{{result}}</h2>

{% if result == "Result: Negative" %}

<h2 class="result">You are Safe</h2>

{% elif result == "Result: Positive" %}

**A6**

<h2 class="result">You might have Skin cancer. Please consult a Doctor</h2>

{% else %}

<h2 class="result"></h2>

{% endif %}

</body>

</html>

**STYLES.CSS**

body {

background: #B5BF9F;

}

.title {

font-family: Patua One;

text-align: center;

font-size: 50px;

margin: 0px 0px;

}

.upload-btn-wrapper {

position: relative;

overflow: hidden;

display: inline-block;

margin-top: 100px;

}

.btn {

border: 2px solid gray;

color: gray; **A7**

background-color: white;

padding: 8px 20px;

border-radius: 8px;

font-size: 20px;

font-weight: bold;

}

.upload-btn-wrapper input[type=file] {

font-size: 100px;

position: absolute;

left: 0;

top: 0;

opacity: 0;

}

.submit-btn {

background-color: #4CAF50; /\* Green \*/

color: white;

padding: 10px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

transition-duration: 0.4s;

cursor: pointer;

**A8**

background-color: white;

color: black;

border: 2px solid #4CAF50;

border-radius: 7px;

}

submit-btn:hover {

background-color: #4CAF50;

color: white;

}

.result {

text-align: center;

color: #0F1527;

}

**A9**

 SCREENSHOT

**A.2 Screenshot**

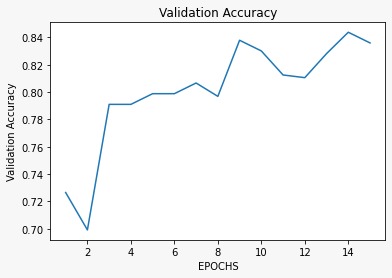
**1.WEB PAGE -INPUT AND RESULT.**

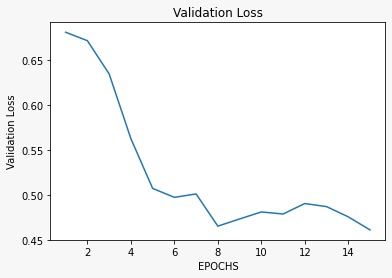




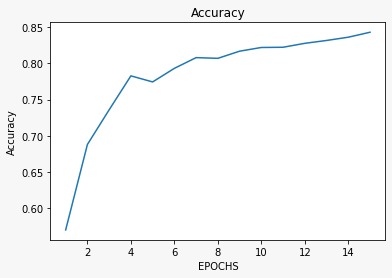
**A10**

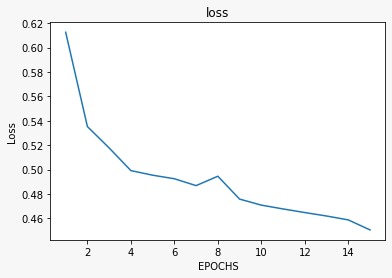
**2. MODEL ACCURACY vs EPOCH GRAPH**

****

****

**A11**

****

****

**A12**

**REFERENCE**

[1]Saket S. Chaturvedi, Kajol Gupta1, and Prakash S. Prasad “Skin Lesion Analyser: An Efficient Seven-Way MultiClass Skin Cancer Classification Using MobileNet”, Department of Computer Science & Engineering, Priyadarshini Institute of Engineering & Technology, Nagpur-440019.

[2] prof. shashank bholane1, shubham patil2, gaurav rajput3, swapnil patil4, sanket gunjalkar @2020 “Skin cancer prediction using image processing and deep learning”. 1Assistant Professor, Department of Computer Engineering, Sinhgad College of Engineering, Pune,Maharashtra, India

2,3,4,5B.E(Computer Engineering), Sinhgad College of Engineering, Pune, Maharashtra, India.

[3] Vijayalakshmi mm,2019 “Melanoma skin cancer detection using image processing and machine learning”, Assistant Professor, Department of Information Science & Engineering GSSSIETW, Mysuru, Karnataka, India.

[4] Sanjana M1, Dr. V. Hanuman Kumar 2 “Skin Cancer Detection Using Machine Learning Algorithm” Department of Computer Science1,2, New Horizon College of Engineering 1,2 Bangalore-560103, Karnataka, India.

**R1**