**AI DERMASSISTANT**

**Automated Skin Cancer Classification Web Based System**



Final Year Project Report

By

**Sanjinee**

In Partial Fulfillment

Of the Requirements for the degree

Bachelor of Science (CS)

Sukkur IBA University,

Sukkur, Sindh Pakistan

(2022)

**DEPARTMENT OF COMPUTER SCIENCE**



**CERTIFICATE**

By signing below, I attest that the project work "**AI DermAssistant (Skin Cancer Classification Web Based App)**" that I submitted to the **Sukkur IBA University Kandhkot Campus** is a record of the original work that I undertook under the guidance of **Mr. Syed Muzamil Hussain Shah**, Lecturer Computer Science, Sukkur IBA University Kandhkot Campus, and **Prof. Dr. Ghulam Mujtaba Shaikh** Professor Computer Science, Sukkur IBA University. The project work that is submitted here partially satisfies the requirements for the Bachelor of Computer Science degree. The findings presented in this study have not been distributed to any other universities or institutes for the purpose of conferring degrees or diplomas.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor** **Co-Supervisor**

Mr. Syed Muzamil Hussain Shah Prof. Dr. [Ghulam Mujtaba Shaikh](mailto:mujtaba@iba-suk.edu.pk)

**External Examiner**

Prof. Dr. Samina Rajper

*Date: 05-09-2022*

**DECLARATION**

I hereby declare that this project report entitled “**AI DERMASSISTANT (Automated Skin Cancer Classification Web Based System)**” submitted to the “**DEPARTMENT OF COMPUTER SCIENCE”**, is a record of an original work done by me under the guidance of Supervisor Muzamil Hussain and that no part has been plagiarized without citations. Also, this project work is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

| **Team Members** | **Signature** |
| --- | --- |
| Sanjinee | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Supervisor:** | **Signature** |
| Syed Muzamil Hussain Shah | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Date:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| **Place:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

**DEDICATION**

I dedicate this to Allah almighty, who has been my strength throughout the project. I also dedicate this to my beloved family who tried hard to provide me with the best education and resources, without their belief it would not have been possible.

I also dedicate this piece of work to my supervisor Sir Muzamil. It was his unwavering support and guidance that led to the completion of the project.

**ACKNOWLEDGEMENTS**

I would like to thank all the people who have helped me along the way and contributed to this research. I am especially grateful to my Supervisor Muzamil Hussain for motivating me to achieve this milestone on time. I also thank the Department of Computer Science for empowering me and arranging FYP activities. Finally, I would like to pay Tribute to the Visionary “Late Professor Nisar Ahmed Siddiqui” for lighting the candle in many students’ lives. I am indebted to all his services. I have learned how to pursue research problems with intellectual rigor and how to critically evaluate my work. Over the last two years I have had the privilege of working with a variety of people who have made my time at Sukkur IBA University, Kandhkot Campus an enjoyable and intellectually stimulating experience.

Finally, I would like to acknowledge the endless love of my family; who have been a constant source of support and who have provided guidance, love and encouragement throughout my life.

Contents

[ABSTRACT](#_heading=h.44sinio) 12

[INTRODUCTION](#_heading=h.2jxsxqh) 13

[1.1 Background](#_heading=h.z337ya) 13

[1.2 Project Goals](#_heading=h.hjsk9l7qw3pm) 13

[1.3 Project Scope](#_heading=h.g0bhpxhssrv0) 14

[1.4 Not In Scope](#_heading=h.s9nn7ruj6ux5) 14

[1.5 Project Objectives](#_heading=h.a99mluddeocm) 14

[LITERATURE REVIEW](#_heading=h.3j2qqm3) 15

[PROBLEM DEFINITION](#_heading=h.1y810tw) 21

[METHODOLOGY](#_heading=h.4i7ojhp) 22

[DETAILED DESIGN AND ARCHITECTURE](#_heading=h.2xcytpi) 24

[5.1 ARCHITECTURE DESIGN APPROACH](#_heading=h.1ci93xb) 24

[5.2 ARCHITECTURE DESIGN](#_heading=h.3whwml4) 25

[5.3 SEQUENCE DIAGRAM](#_heading=h.qkj03e11hw2v) 27

[5.4 ACTIVITY DIAGRAM](#_heading=h.xl6hy5toikd8) 28

[5.5 FUNCTIONAL HIERARCHY](#_heading=h.fswq7ppb1y69) 29

[5.6 CLASS DIAGRAM](#_heading=h.6psidm6kl9zz) 30

[5.7 USE CASE DIAGRAM](#_heading=h.h834q7a3th8c) 31

[IMPLEMENTATION AND TESTING](#_heading=h.rqsopsogf7w9) 32

[6.1 IMPLEMENTATION](#_heading=h.147n2zr) 32

[6.1.1 Frontend (Web Application)](#_heading=h.3o7alnk) 32

[6.1.2 Backend](#_heading=h.23ckvvd) 32

[6.2 TESTING](#_heading=h.ihv636) 33

[RESULTS AND DISCUSSION](#_heading=h.4f1mdlm) 38

[7.1 BACKEND MODEL](#_heading=h.8oxpmp3nntqs) 38

[7.2 AI DERMASSISTANT WEB BASED SCREENS](#_heading=h.35blwfevu2am) 50

[CONCLUSION AND FUTURE WORK](#_heading=h.1iy1namritb3) 56

[8.1. CONCLUSION](#_heading=h.3q5sasy) 56

[8.2. RECOMMENDATIONS AND FUTURE DIRECTIONS](#_heading=h.25b2l0r) 56

[REFERENCES](#_heading=h.kgcv8k) 57

**LIST OF FIGURES**

[Figure 1 Architecture Diagram](#_heading=h.h2333er3yxom) 25

[Figure 2 Sequence Diagram](#_heading=h.lekj7yrmmhfa) 27

[Figure 3 Database Diagram](#_heading=h.5tuhx6ngxb8f) 28

[Figure 4 Activity Diagram Frontend](#_heading=h.5tuhx6ngxb8f) *29*

Figure 5 Functional Hierarchy 29

[Figure 6 Class Diagram](#_heading=h.9f3854pjxho4) 30

[Figure 7 Use Case Diagram](#_heading=h.cythkm54uzbs) 31

[Figure 8 validation dataframe and train dataframe](#_heading=h.zdpoahaztlaa) 44

[Figure 9 model initialization and transforms](#_heading=h.3eogtvqquonn) 44

[Figure 10 train function](#_heading=h.4ksrzauu7u1) 45

[Figure 11 validate function](#_heading=h.efwoogupgzil) 45

[Figure 12 train iteration](#_heading=h.s7lp51j05n39) 46

[Figure 13 Classification report on ResNet model ISIC 2019 (9 classes)](#_heading=h.kljpa0b0l2om) 46

[Figure 14 Classification report on DenseNet model ISIC 2019 (9 classes)](#_heading=h.gyghrykjjdan) 47

[Figure 15 Classification report on ResNet model ISIC 2019 (7 classes)](#_heading=h.ncss5fsklntp) 47

[Figure 16 Classification report on DenseNet model ISIC 2019 (7 classes)](#_heading=h.f3ua04uv5m5t) 48

Figure 17 Classification report on retrained ResNet model  *48*

[Figure 18 Classification report on retrained DenseNet model](#_heading=h.ahp8c66gajlw) 49

[Figure 19 Register Screen](#_heading=h.pn9yb9i4l8k2) 50

[Figure 20 Login Screen](#_heading=h.qd53534i8b5o) 50

# Figure 21 Dashboard Statistics Screen 51

[Figure 22 Dashboard graphs](#_heading=h.6g29awybfdd6) 51

[Figure 23 Dashboard Screen Data](#_heading=h.k0wrjdphaf0w) 52

[Figure 24 Dashboard graphs](#_heading=h.9j00y9wvcff2) 52

[Figure 25 Cancer Classification Test Screen Contributors Section](#_heading=h.t8nar31cmapq) 53

[Figure 26 Cancer Classification Test Screen Technologies Used Section](#_heading=h.gtvawl74l7tc) 53

[Figure 27 Author Screen](#_heading=h.vmnl0c1p9lo8) 54

[Figure 28 Contact Us Screen](#_heading=h.et0fmd2bf9lj) 54

[Figure 29 Image Gallery](#_heading=h.pvzwsfrgixh7) 55

**LIST OF TABLES**

[Table 1 ANN based Skin Cancer Classification](#_heading=h.d0j9b1x10gr6) 18

[Table 2 CNN based Skin Cancer Classification](#_heading=h.xkr4h88f5e4a) 21

[Table 3 Class Diagram](#_heading=h.z0kf9ttzvwz) 31

[Table 4 Test Case 1](#_heading=h.rbdgrdmfnedl) 34

[Table 5 Test Case 2](#_heading=h.3gf75wz6or0b) 36

[Table 6 Test Case 3](#_heading=h.ra6h34r0zlsx) 37

[Table 7 ResNet on ISIC 2019 Dataset](#_heading=h.i5hheiv0suur) 40

[Table 8 ResNet on ISIC 2019 Dataset](#_heading=h.vtl8li30vd1z) 40

[Table 9 Retrained ResNet on ISIC 2019 and HAM10000 Dataset](#_heading=h.k1xez6u4jfvb) 41

[Table 10 ResNet on HAM10000 Dataset](#_heading=h.9gec487uuz0i) 42

[Table 11 DenseNet on ISIC 2019 Dataset](#_heading=h.7nbrdvxhcwqg) 42

[Table 12 DenseNet on ISIC 2019 Dataset](#_heading=h.kj82jikbjps1) 43

[Table 13 Retrained DenseNet on ISIC 2019 and HAM10000 Dataset](#_heading=h.of0rvhpfkob4) 43

[Table 14 DenseNet on HAM10000 Dataset](#_heading=h.4h8zsv6plu5o) 44

[Table 15 Classification Results of Retrained models](#_heading=h.qg13ylarfrzh) 44

# **ABSTRACT**

Since the last decade, Cancer has been a severe disease that people are most afraid of. The American Academy of Dermatology Association states that, in the U.S., more than 9,500 people are diagnosed with skin cancer every day. More than two people die of the disease every hour. More than 5.4 million cases of nonmelanoma skin cancer were treated in over 3.3 million people in the U.S. According to the National Library of Medicine National Center for Biotechnology Information, the mean annual cost per patient using Medicare for all cancers was $17,094. Also, the overall process of identifying the cancer is time consuming.

Therefore, the idea is to develop such a product that leverages the latest AI trends at backend and is accessible to the end-users within the reach of a few seconds to get their skin lesion type identified. Among those trends:

* Convolutional neural network is used for feature extraction
* Retraining the neural network using different available datasets like ISIC 2019, HAM10000 etc is done. Because there were not sufficient numbers of images available in the dataset, a kind of ensemble learning is implemented in which an optimum model is achieved by training the same base model on two different datasets.
* ViT (Vision Transformers) are used for improved computation efficiency and accuracy.

Ultimately, this project would open the doors of the latest computing technologies to be used for the field of medical healthcare sciences for further research purposes.

***Chapter 1***

# **INTRODUCTION**

## **1.1 Background**

'AI will not replace doctors; instead it will augment them, enabling doctors to practice better medicines with greater accuracy' is what AI DermAssistant is all about. This end product is supposed to assist doctors to have an idea of the Skin lesion and it would make them able to identify the type of skin lesion within a few seconds unlike the traditional methods of identification of lesions which are unaffordable, time consuming and costly. The Biopsy (process of testing skin lesions) costs approximately in the range of $1000 to $5000, and it takes 2 to 3 days for patients to get their results of the reports. It is the reason the project would be presented as an open source web based platform that everyone can use. The core aim of this project is to provide accessibility to the end user may it be any patient or any consultant. It does not require patients to provide any amount of money. Instead what they need to do is to take a picture of the skin lesion and upload it to the web application. Then the web application would provide them the classification results.

The proposed solution (AI DermAssistant Web Application) is structured in a way that at backend it has a deep learning model deployed. This model would take the image and but before that preprocessing of the image would be done over the image. The image would be given to the model and the model would present the classification result to the end users within seconds.

## **1.2 Project Goals**

**Accessibility:** A common person can have the access of an application using which he can classify Skin Cancer

**Time Saving:** Time can be saved due to autonomous classification of Skin Cancer

**Rapid Results:** The proposed idea can provide rapid results to the end user unlike the traditional methods

**Training on a variety of datasets:** While training, 2 datasets HAM10000 and ISIC 2019 are used for the classification

## **1.3 Project Scope**

As a Minimum Viable Product, I have presented a product that would classify the skin cancer type, for that user will have to get himself registered, then the user will be redirected to the login page. From the login page, the user will get redirected to the dashboard once the credentials get verified. From the dashboard, the user can upload the image and choose different model architectures and submit his response. At submitting the user response, the user gets the result of Skin Cancer Classification. After that, the user can save the image to the file handling system of the server. The project classifies only seven classes of Skin Cancer currently.

## **1.4 Not In Scope**

* This project does not store the classification result to the database. Rather it stores the image to the File Handling System of Django.
* This project does not allow to capture images at the runtime, it requires the user to have the images in the device using which he access the web interface

## **1.5 Project Objectives**

I am over to developing a system that provides accessibility to users for classifying skin lesions as 7 different types of skin cancer, and telling about the model used. The main objective is to get images as input and apply different techniques of image processing and deep learning to identify Skin Cancer. So for that, the input image would be classified using the dashboard of this application and the image can also be saved to some storage. The system is and would be thoroughly tested throughout the development cycle.

**Chapter Summary**

This chapter mainly discussed the background of the project, the goal that must be achieved. And finally, the benefits achieved by the users of the web application.

***Chapter 2***

# **LITERATURE REVIEW**

The work in this paper is focused on providing a solution to the end users for the cause of the classification of skin cancer. This research covers the many aspects that have an impact on the classification of skin cancer that leads the patient to many unbearable, and time consuming test methods and for that doctors have to wait for the results of the reports till days, only then he could provide the consultation to the patients. Apart from that, the testing centers have to face the shortage in the number of available healthcare professionals to diagnose and provide consultation for the test results. Therefore such a system must exist to allow both the patients and doctors to classify the type of skin lesion. Such a solution can be made possible using the latest machine learning and deep learning trends(model architecture) like: ResNet, DenseNet, Inception, Vgg, SVM, SVM along with CNN extracted features etc. For the better understanding of the already existing research, I performed a search on google scholar and IEEE Xplore, and found out many research papers related to my final year project. At last, I highlighted some of the research papers related to my research findings. Below I have given the research findings of those research papers based on Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) Based Classification. ANN is employed in order to categorize the extracted features. The amount of input photos determines how many hidden layers there are in an ANN. The layer of inputs of the dataset is connected to the ANN with the hidden layer. A supervised or unsupervised learning method may be used to process a labeled or unlabeled dataset, respectively. A neural network learns the weights existing at each network connection or link using a backpropagation or feed-forward architecture. The underlying dataset is used by both systems using a different pattern. Neural networks based on feed-forward design only transmit data in one way. Only the input layer and the output layer experience data flow.

Xie et al. [[1](#_heading=h.kgcv8k)] proposes a paradigm for classifying skin lesions that divides lesions into benign and malignant categories. Three phases made up the planned system's operation.First of all, lesions were extracted from photos using a self-generating neural network. Features including the tumor boundary, texture, and color details were extracted in the second phase. A total of 57 features were retrieved by the algorithm, including 7 unique features that describe lesion borders. The features' dimensionality was decreased using principal component analysis (PCA), which allowed the best set of features to be chosen. At the end, an ensemble neural network model was used to make classification for lesions. The model performance and training accuracy was improved by introducing backpropagation to the neural network. Furthermore, the classification results of machine learning algorithms like SVM, KNN, random forest, Adaboot, etc were observed. With an accuracy of 91.11, the proposed model performed 7.5% better in terms of sensitivity than other classifiers.

On the basis of backpropagation ANN, an automated model for diagnosing skin cancer was suggested [[2](#_heading=h.kgcv8k)]. For feature extraction of the images, this system used a 2D wavelet transform approach. By the proposed ANN model, the input images were classified in two main classes as either cancerous or noncancerous.

Choudhari and Biday [[3](#_heading=h.kgcv8k)] suggested the ANN-based skin cancer diagnosis model. Using a maximum entropy thresholding method, images were segmented. The distinctive features of skin lesions were extracted using a gray-level co-occurrence matrix (GLCM). With an accuracy level of 86.66%, a feed-forward neural network segmented the lesion images into either malignant or benign classes of skin cancer.

Aswin et al. presented a genetic algorithm and Artificial neural network based technique to identify skin cancer [[4](#_heading=h.kgcv8k)]. With the help of the region of interest (ROI) extraction technique and the medical imaging programme Dull-Rozar, images were preprocessed to remove hair. Additionally, the segmented images' distinctive features were extracted using the GLCM approach. The classification of lesion images into classes for malignant and noncancerous lesions was then done using a hybrid ANN and GA classifier. The proposed approach scored 88% in overall accuracy.

This table shows the technical details of different skin cancer detection systems.

# *Table 1 ANN based Skin Cancer Classification*

| Ref | Skin Cancer Diagnosis | Implementation | Dataset | Description | Results(%)  Accuracy |
| --- | --- | --- | --- | --- | --- |
| [[1](#_heading=h.kgcv8k)] | Melanoma/non melanoma | ANN with Levenberg–Marquardt (LM), resilient backpropagation (RBP), and scaled conjugate gradient (GCG) learning algorithms | 135 lesion images | Combination of multiple classifiers to avoid the misclassification | Accuracy (SCG:91.9, LM: 95.1, RBP:88.1) |
| [[3](#_heading=h.kgcv8k)] | Malignant/non Malignant | ANN with backpropagation algorithm | 30 cancerous/noncancerous images | RGB color features and GLCM techniques for feature extraction | 86.66% |
| [[2](#_heading=h.kgcv8k)] | Melanoma/Non Melanoma | ANN with backpropagation algorithm | 90 dermoscopic images | maximum entropy for thresholding, and gray level co-occurrence matrix for features extraction | 86.66% |
| [[5](#_heading=h.kgcv8k)] | Malignant/benign | Feed-forward ANN with the backpropagation training algorithm | 326 lesion images | Color and shape characteristics of the tumor were used as discriminant features for classification | 80% |
| [[7](#_heading=h.kgcv8k)] | Malignant/non malignant | ANN with backpropagation algorithm | 31 dermoscopic images | 2D-wavelet transform for feature extraction and thresholding for segmentation | – |
| [[6](#_heading=h.kgcv8k)] | Malignant/non Malignant | Backpropagation neural network as NN classifier | 448 mixed-type images | ROI and SRM for segmentation | 70.4% |

An important kind of deep neural network that is successfully applied in computer vision is the convolution neural network. It is employed for image classification, object detection, and image recognition. CNN is an excellent mechanism for gathering and learning data because it uses simpler characteristics like curves and edges to create more complex features like shapes and corners. Convolution, nonlinear pooling, and fully connected layers make up CNN's hidden layers. Multiple convolutional layers may be present in a CNN, which may be followed by a number of fully linked layers. Convolution layers, pooling layers, and fully-connected layers are the three primary types of layers used in CNN.

A multi-scale CNN employing an inception v3 deep neural network trained on an ImageNet dataset was presented by DeVries and Ramachandram [[8](#_heading=h.kgcv8k)]. For the categorization of the lesion images of skin cancer, pre-trained inception-v3 model was tuned on two resolution scales, coarse scale and finer scale. The coarse-scale was employed to record the lesion's general contextual information as well as its shape features. The finer scale, in contrast, collects textural information about lesions to distinguish between various skin lesion types.

An extremely deep CNN was suggested for melanoma detection by Lequan et al. [[9](#_heading=h.kgcv8k)]. To enhance performance, a fully convolutional residual network (FCRN) with 16 residual blocks was applied to the segmentation process. The suggested method classified data by averaging SVM and softmax classifiers. It demonstrated 82.8% accuracy without segmentation and 85.5% accuracy for melanoma classification.

Mahbod et al. [[10](#_heading=h.kgcv8k)] planned a method to find out the characteristics from various pre-trained deep Convolutional neural networks for classification of skin lesions. For the purpose of generating deep features, Alexnet, ResNet-18 and VGG16 are used, then for the purpose of classification of multi classes, generated features were trained on SVM. At the end, the results of the SVM are used to confirm the classes of skin lesions. This system had an accuracy of 97.55% on ISIC 2017 dataset and 83.83% space beneath the curve (AUC) performance for keratosis (SK) and malignant melanoma classification.

For the classification of 12 different types of skin lesions, a deep CNN architecture based on pre-trained ResNet-152 was proposed [[11](#_heading=h.kgcv8k)]. It was first trained using 3797 images of lesions, but later, 29 instances of augmentation were applied depending on changes in lighting situations and scale. For the classification of hemangioma lesions, pyogenic granuloma (PG) lesions, and intraepithelial carcinoma (IC) skin lesions, the proposed technique produced an accuracy value of 0.99.

Dorj et al. [[12](#_heading=h.kgcv8k)] suggested a method for classifying photos of four different kinds of skin lesions. After features were extracted using a pre-trained deep CNN named AlexNet, an error-correcting output coding SVM was utilized as a classifier. For SCC, actinic keratosis (AK), and BCC, the proposed system produced the highest scores of the average sensitivity, specificity, and accuracy: 95.1%, 98.9%, and 94.17%, respectively.

# *Table 2 CNN based Skin Cancer Classification*

| Ref | Skin Cancer Diagnoses | Classifier and Training Algorithm | Dataset | Description | Results(%) |
| --- | --- | --- | --- | --- | --- |
| [[8](#_heading=h.kgcv8k)] | Melanoma/SK | Deep multi-scale CNN | ISIC dataset | The proposed model used the Inception-v3 model, which was trained on the ImageNet. | Accuracy (90.3), AUC (94.3) |
| [[9](#_heading=h.kgcv8k)] | Benign/malignant | A very deep residual CNN and FCRN | ISIC 2016 database | FCRN with a multi-scale contextual information integration technique was proposed for accurate lesions segmentation | Accuracy (94.9), sensitivity (91.1), specificity (95.7), Jaccard index (82.9), dice coefficient (89.7) |
| [[10](#_heading=h.kgcv8k)] | Malignant melanoma/SK | SVM classification with features extracted AlexNet, ResNet-18, and VGG16 | ISIC dataset | SVM scores were mapped to probabilities with logistic regression function for evaluation | Average AUC (90.69) |
| [[11](#_heading=h.kgcv8k)] | Malignant melanoma and BC carcinoma | CNN with Res-Net 152 architecture | The first dataset has 170 images the second has 1300 images | Augmentor Python library for augmentation. | AUC (melanoma: 96, BCC: 91) |
| [[12](#_heading=h.kgcv8k)] | BCC/SCC/melanoma/AK | SVM with deep CNN | 3753 dermoscopic images | Pertained to deep CNN and AlexNet for features extraction | Accuracy (SCC: 95.1, AK: 98.9, BCC: 94.17) |

***Chapter 3***

# **PROBLEM DEFINITION**

The world is moving at such a fast pace that almost everybody is occupied with some business. Almost everything is digitalized, may it be any sector like: Agriculture, Academic and even healthcare itself. Robots are operating on humans in the operation rooms on the commands of some specialists. How can we not implement some identification methodology to this grim skin disease which is so severe. If we do not introduce a product like this in the market, we would lag behind following the traditional methods and we would never be able to get advanced.

If such a system does not exist, there would be required 2 to 3 working days to get your results of reports of biopsy. Not only that, but the cost of biopsy and other methods is quite expensive as well. Apart from that there is an increasing need of healthcare professionals e.g. doctors, nurses, physicians. WHO’s global strategy on Human Resource for Health reports the shortage of 9.9 million physicians, nurses, and doctors, etc.

Besides, the product is related to medical sciences and cancer, this requires the research to be significant and efficient. Else, the lives of humans would be at considerable risk. This being the reason, the model of the backend must be extensively trained on more than abundant images so that the model can accurately and significantly contribute to classify the type of skin cancer.

***Chapter 4***

# **METHODOLOGY**

For the development of the project AI DermAssistant, Agile Development Methodology is used. This methodology allows to split down the large development projects into smaller sprints in which the different chunks of the project are developed based on their importance. Along with that, different tools and approaches were required to plan, analyze, design, implement and test the project that are as follows:

**4.1 PLANNING**

In this phase, project requirements, functionalities and responsibilities were analyzed. For this purpose, the SCRUM meeting was held with the supervisor every Tuesday to discuss the progress and updates.

**4.2 DESIGN**

After gathering the requirements and planning a mind map, designing was the next phase. In this phase, I designed the class diagram, system diagram, architecture design of the project and use-case diagram. The following tool is used in this phase:

1. **app.creately.com:** It was used for designing flow charts and other diagrams such as use case, entity relationship diagram, database diagram and class diagram.
2. **Paper Prototypes**: It was used to design the interface of the application (AI DermAssistant).
3. **canva.com:** It was also used to design the Interface of the web application.

**4.3 IMPLEMENTATION**

Next to designing was the creation of the design diagrams to software components. The tools and technology required in this phase are the following:

1. **Deep Learning:** This AI based mechanism is used because it allows feature extraction automatically by using convolutional neural networks. If it is not used, manual feature extraction would have to be done. That would be inefficient.
2. **Kind of Ensemble Learning**: Since the number of images are not sufficient in HAM10000 for training purposes, the model is retrained on two different datasets.
3. **Pytorch:** This would be used to support the development, training, and testing of deep learning models.
4. **Anaconda:** This is a development environment, which contains all modules, packages and libraries which will be used in the development of the system.
5. **Google Colab**: This is an online environment that provides its storage, RAM, processing, GPU, CPU, TPU for enhanced effective training and testing sessions.
6. **Django**: Django is a model-template-views-based web framework that is free and open-source that uses Python. It is a very advanced Python web framework that enables rapid development of secure and maintainable websites.
7. **HTML, CSS, JavaScript**: These were used for building the frontend of the application
8. **Sklearn**: The most effective and reliable Python machine learning library is named Skearn (Skit-Learn). Through a consistent Python interface, it offers a variety of effective methods for dimensionality reduction, clustering, and classification. It provides many utilities that are useful in the development of deep learning models.
9. **Git and Github**: Git is a version control tool widely used to develop projects. It was used to track progress on every stage throughout the project development. And Github was used to push the code in a repository.
10. **Datasets**: ISIC 2019 and HAM10000 datasets are used.

**4.4 TESTING**

After the implementation phase, the whole system was tested manually by giving different inputs.

***Chapter 5***

# **DETAILED DESIGN AND ARCHITECTURE**

The architecture for the AI DermAssistant system was designed using Django framework architecture. It is a model-template-views (MTV) architecturally sound web framework built on Python. An MVC architecture may be observed in the fundamentals of the Django framework. It consists of a regular-expression-based URL dispatcher, an object-relational mapper (ORM) that mediates between data models (specified as Python classes) and a relational database, and a system for handling HTTP requests ("Controller"). This way it organized the structure of the system in presentation, application, and data layers. Moreover for the backend, there is implementation of the logic of the model in the Pytorch. Finally, each subsystems’ architecture is demonstrated using the functional description with the help of data flow and structural decomposition diagrams.

## **5.1 ARCHITECTURE DESIGN APPROACH**

The 3-layered architecture design approach was used to compose 3 layers of logical computing. A three-layered architecture is a client-server architecture in which the user interface, functional process logic, and data access are developed and maintained as independent modules on separate platforms.

As illustrated in Figure 1, the presentation layer contains interfaces of web application, the application layer contains business logic and finally, the data layer contains the database of the system. This approach was selected because all the components on the presentation layer communicate with the application layer using views through the Django framework.

# 

# *Figure 1 Architecture Diagram*

## **5.2 ARCHITECTURE DESIGN**

Basically the system is based on two architectures, one is its frontend for User Interface and other is its backend for deep learning based approach where neural network is used.

***Frontend***

The front end is divided in two portions. One is user based and the other is admin based.

***Admin***: Admin dashboard is provided by Django. The admin can update, delete the user in the system. Apart from that, the admin can also use the system the same way the user can use it.

***User***: The user of the Automated skin cancer classification system can register himself, log in to the system, access the dashboard, upload images to get the classification results of the trained model and see the uploaded images to the web application.

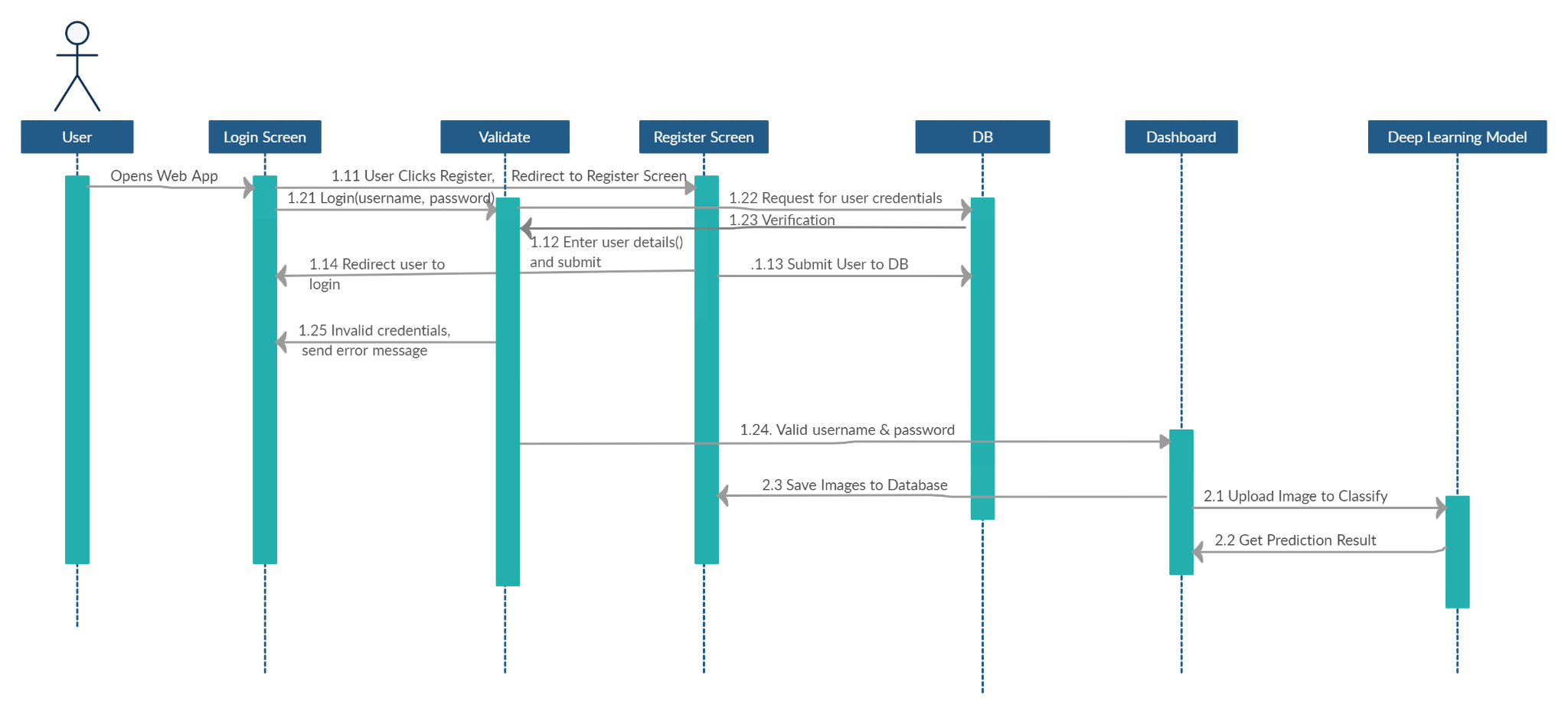
Both these systems interact with the backend on the django framework for authentication, authorization.

When a user signs up to the system, his information goes into the database via the backend. Once a user successfully signs up then he gets the success message and gets redirected to the sign in page from where the user gets redirected to the dashboard, once he signs in successfully. Dashboard allows users to go to the classification test page, or users can see the images uploaded, and can explore the statistics of skin cancer available on the dashboard. Users can also find the sources where they can study more about skin cancer.

## 

## **5.3 SEQUENCE DIAGRAM**

The design layout of our web application, we have explained further using:

**Sequence diagram**: It demonstrates the way object objects in the real follow the flow on the web application by providing interaction using shapes and users.

# *Figure 2 Sequence Diagram*

Figure 2 Description:

It shows that whenever the user opens the web application, he gets redirected to the login screen. Because if a user is not logged in, then he cannot access the web application.

After successful login, the application directs the user to the dashboard, from where he can access different screens and the screen to where the model is deployed.

## **5.4 ACTIVITY DIAGRAM**

The design layout of our web application, we have explained further using:

* **Activity diagram of backend:** It visually presents a series of actions in the project/web application, It will have an initial state and a final state.

## 

# *Figure 3 Database Diagram*

Figure 3 demonstrates that:

The dataset of images would be split into a train set and testset. For the backend processing, convolutional neural network layers are used for feature extraction. There are different model architectures used for the training, i,e. ResNet, Vgg, DenseNet.

* **Activity diagram of frontend**

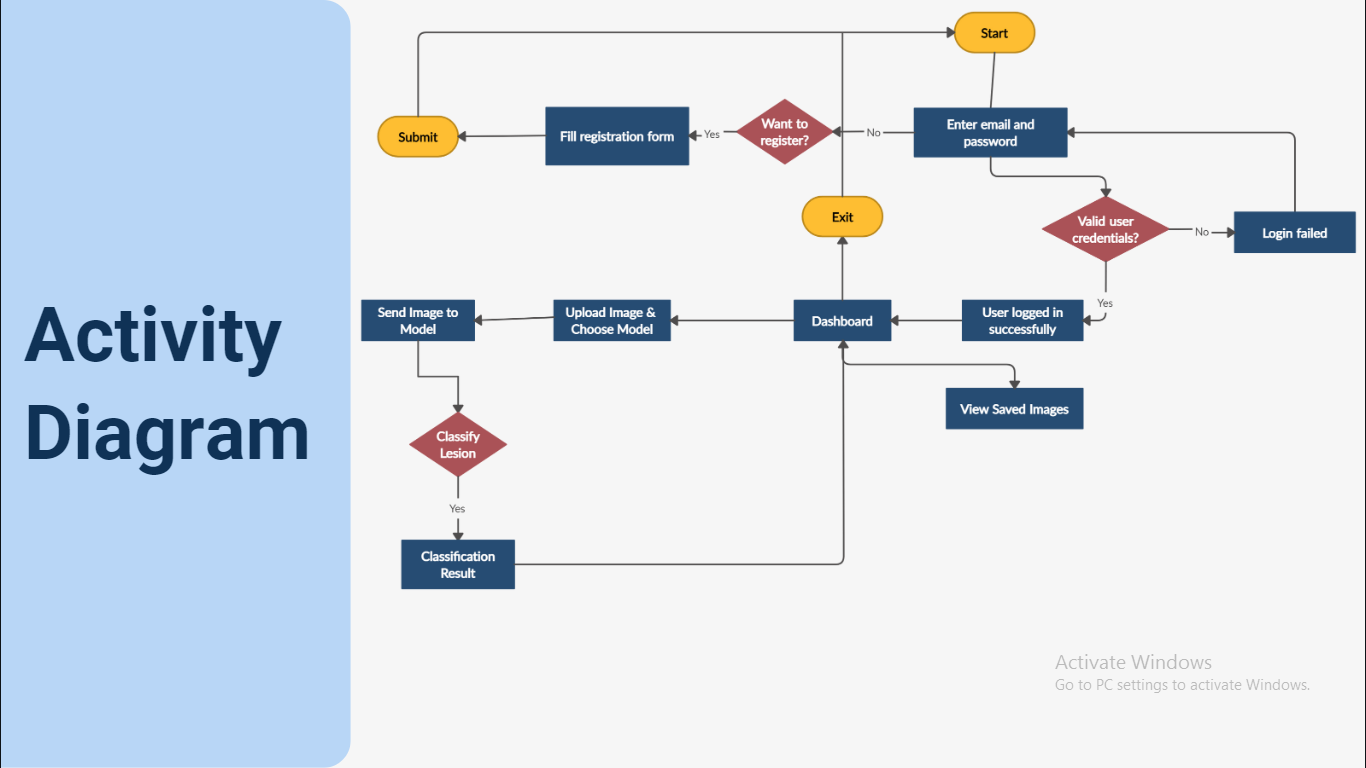
*******Figure 4 Activity diagram of frontend*

Figure 4 demonstrates that the flow of the web application

## **5.5 FUNCTIONAL HIERARCHY**

## *Figure 5 Functional Hierarchy*

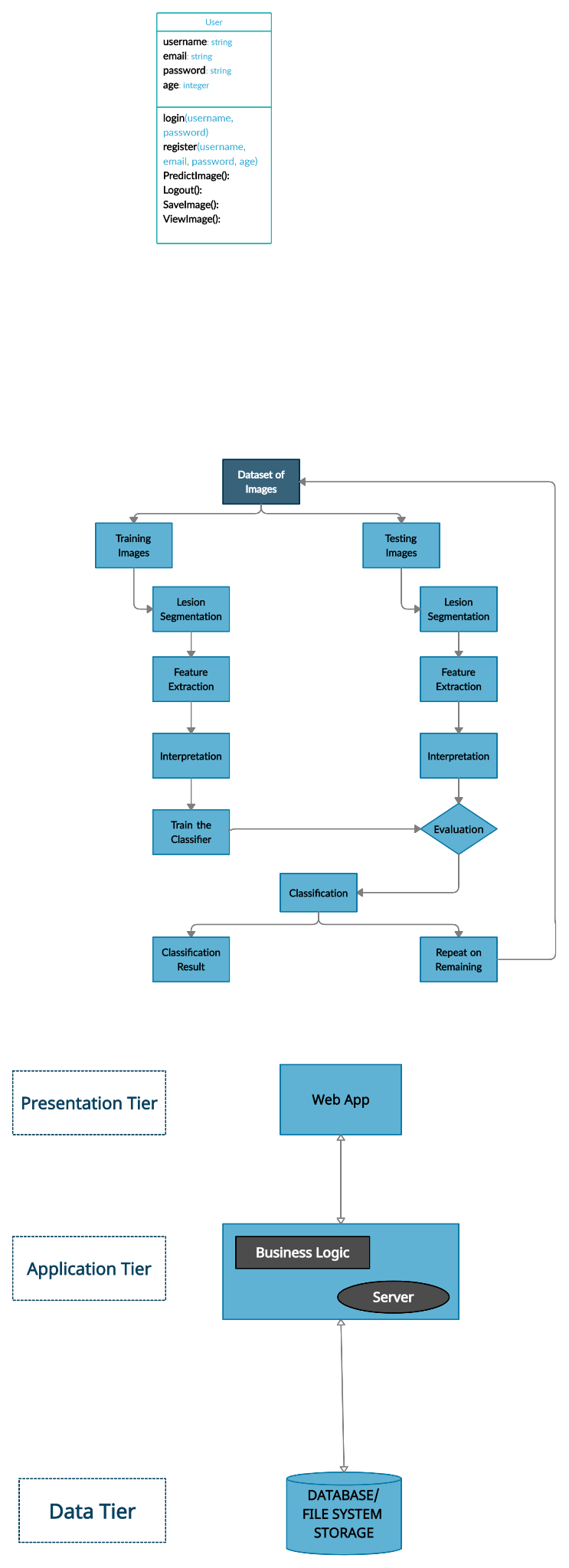
## **5.6 CLASS DIAGRAM**

To present the structure of the application, Class diagrams are used to explain conceptual modeling and detailed modeling.

# *Table 3 Class Diagram*

| **Class Name** | **Attributes of the class** | **Methods of Class and Description** |
| --- | --- | --- |
| User | username, email, password, age | * Login(): using email and password, user can login * Register(): providing different information, user can register * PredictImage(): if user is logged in, he can provide image to predict type of skin lesion * Logout(): user can logout from dashboard * ViewImage(): user can load the saved images |

**Relationship between classes:**

There is only one class, that is of user. 

# 

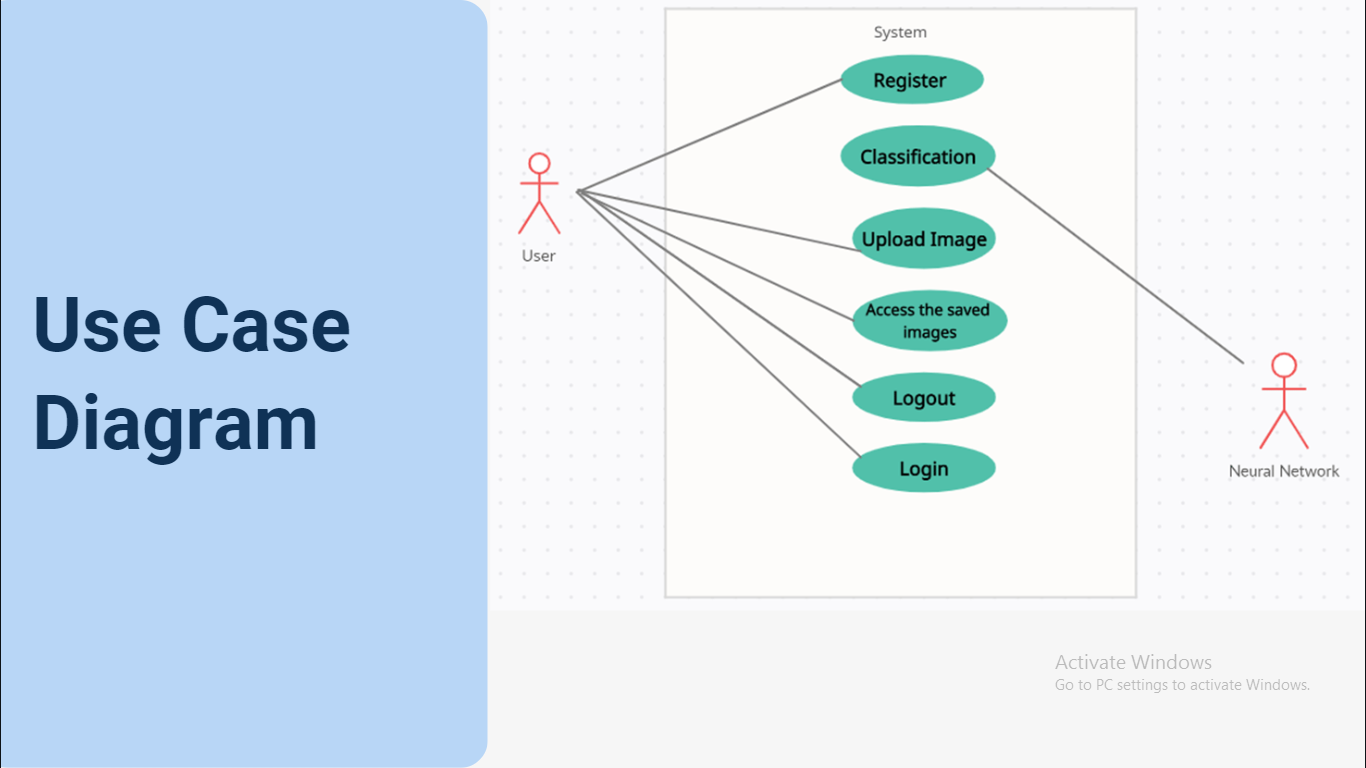
# 

# 

# 

# *Figure 6 Class Diagram*

## **5.7 USE CASE DIAGRAM**

******

# *Figure 7 Use Case Diagram*

Figure 7 demonstrates that:

* In this diagram, there are two actors. One is the user of the application and another is the neural network model architecture.
* The neural network works for classification of the lesion images only.
* The user can access the registration page, login page, dashboard, access the gallery, and logout.

***Chapter 6***

# **IMPLEMENTATION AND TESTING**

All the methods, tools, trends, the kinds of testing methodologies that are used or implemented to develop the Minimum Viable Product (MVP) are discussed in this chapter. Not only that, but this chapter also covers the core functionalities in narrative format.

## **6.1 IMPLEMENTATION**

To implement the system, different libraries were used for frontend web application, and backend model training and creation.

### **6.1.1 Frontend (Web Application)**

* **Django:** Django is a high-level model-template-views-based web framework that is free and open-source that uses Python. It is very advanced in a sense that rapid development of secure and maintainable websites is achieved using Django. This framework allowed me to create templates for the user interface and provided me with views for handling the backend of those templates.
* **MYSQL:** This Database was used to store, organize and manage the data throughout the system.
* **Material Kit:** This library was helpful in providing built-in components for web application interface designing and easy implementation of reactive and attractive layouts.

### **6.1.2 Backend**

* **Pytorch:** Pytorch is used for backend development, it is open source. Pytorch is used to support the development, training, and testing of deep learning models.
* **OpenCV, PIL and other libraries:** These libraries were used for the image transformation, resizing of the images, reading images so that images could be provided to the model for training purpose and testing as well.

The core functionalities of this project include:

* **Registration:** The most important functionality of the system is the registration of the user to this web based system.
* **Authentication:** User once registers himself, can login to the system. Based on the user credentials, it can be confirmed that the user is authenticated.
* **Authorization:** While interacting with the systems, users can only perform actions that they are authorized to perform according to their roles. For example, only the admin can access the admin dashboard using his credentials.
* **View Dashboard:** User can view the dashboard on his successful signing in. From the dashboard, users can see the graphs of the facts and figures of the skin cancer statistics. Users can explore different pages using the navigation bar at the header of the dashboard.
* **Upload Image for classification:** Logged in users can upload images and get classification results. The uploaded image would be saved to the server.
* **View Images:** The images that user has uploaded for classification are saved to the server and that can be viewed using the image gallery.

To implement these core functionalities, optimized code was written to achieve code reusability, understandability and easy management. A brief walkthrough of the code is included from the backend logic implementation.

## **6.2 TESTING**

For the purpose of testing the system, the system has to pass through some test cases so that it could be assumed that the system ensures the expected functionalities and works as required.

# *Table 4 Test Case 1*

| **Test ID** | 01 |
| --- | --- |
| **Test Title** | Register User |
| **Module Name** | Signup User |
| **Test Priority** | High |
| **Description** | Verify the format of data entered in the Registration |

| **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| Check the Signup user module by providing data in a correct format | 1. Provide Username  2. Provide Email  3. Provide Password  4. Provide Password Check | Name = test  Email=girl.code@gmail.com  Password = siba123!  Password check = siba123! | The user should be registered and message should be displayed to the user for successful registration | As Expected | Pass |
| Check the register module by providing data the in incorrect format | 1. Provide Username  2. Provide Email  3. Provide Password  4. Provide Password Check | Name = test  Password = siba123!  Password check = siba123! | The user should not be registered and message should be displayed to the user  e.g. Please fill out this field | As Expected | Fail |

This test Case 1 is created to test the registration of the donor. When the user opens the web application, he gets directed to the login page. if the user does not have an account created, he will have to register himself by clicking on the sign up button. From the click, he goes to the register page, there he provides all the required fields and signs himself up.

# *Table 5 Test Case 2*

| **Test ID** | 02 |
| --- | --- |
| **Test Title** | Login user |
| **Module Name** | Login |
| **Test Priority** | High |
| **Pre-condition** | The user must have registered |
| **Description** | Verify user email and password is in the correct format and the user is registered or not. |

| **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| Check the user login with the correct validation | 1. Enter username  2. Enter Password  3. Click the login button | username = test  Password = siba123! | If the credentials are authentic then the user should be able to log in. | As Expected | Pass |
| Check the user login with Incorrect validation | 1. Enter username  2. Enter Password  3. Click the login button | username = test  Password = kiba123 (This is not the actual password) | The user should not be able to log in. because the password is incorrect. | As Expected | Fail |

The above test case is developed to log in to the user (user and admin). The pre-condition for this test case is that the user must be registered to our system. The user must enter the username and password in the correct format. That email and password would be checked against the registration data. If it is valid then it will get access to the system.

# *Table 6 Test Case 3*

| **Test ID** | 03 |
| --- | --- |
| **Test Title** | Upload Image |
| **Module Name** | Upload Image |
| **Test Priority** | High |
| **Description** | In the upload image, All fields are required. If the user left any field empty, it will generate a MultiValueDictKeyError. |

| **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| Check the user enters all the data  fields. | 1. Browse Image  2. Select Model from DropDown  3. click submit button | Model = any from the dropdown items | The user successfully uploaded an image and got the classification results | As Expected | Pass |
| Check the user enters all the data  elds. | 1. Browse Image  2. Select Model from DropDown  3. click submit button | No file is chosen. | The user would get an error with MultiValueDictKeyError | As Expected | Fail |
| Check the user enters all the data  elds. | 1. Browse Image  2. Select Model from DropDown  3. click submit button | Image file is not chosen | The user would get an error with UnidentifiedImageError | As Expected | Fail |

The above test case is developed to upload images and get predictions from the system. The post condition for that is images will be saved in the file handling system of django and the user can not leave any field empty. All fields are required.

***Chapter 7***

# **RESULTS AND DISCUSSION**

The results of the developed system are presented in this chapter. Besides, screen snapshots of the frontend and backend research findings are also attached below to demonstrate the novelty, interface and working of the proposed system.

## **7.1 BACKEND MODEL**

#### Measuring the performance of our model

#### Confusion Matrix

True Positive (TP): A label was predicted positive and it came out to be true in real

True Negative (TN): A label was predicted negative and it came out to be true in real

False Positive (FP): A label was predicted positive and it came out to be untrue in real

False Negative(FN): A label was predicted negative and it came out to be untrue in real

#### Precision = TP/(TP+FP)

It demonstrates how many labels are predicted correctly as a percentage of the total number of predicted labels.

#### Recall = TP/(TP+FN)

It demonstrates how many labels are actually positive. If a model has a high recall, it performs good for all the positive cases. On the contrary, a model with a low recall may not be able to find all the positive cases.

#### F1 score= 2.(precision.recall/(precision+recall))

F1 score performs for balancing out the precision and recall for positive classes

#### ResNet Model Architecture

The reason why the ResNet model is used is that it performs efficiently for computer vision tasks. To escape the vanishing gradient problem faced by a very deep neural network, ResNet makes use of skip connection using which it can present the output from an earlier layer to a later layer. I have made use of ResNet-50, this is basically a sub-version of ResNet-152 which has 152 layers. ResNet-50 performs well for transfer learning as well. The way ResNet ensures efficiency and performance is by using the skip connection, they allow the shortcut path to let their gradient flow through those layers and by using an identity function to confirm that the higher layer at least performs like the lower layers in terms of good accuracy. ResNet makes use of Element-wise addition.

# *Table 7 ResNet on ISIC 2019 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 40.5 | 0.52 | 0.38 | 0.44 |
| Basal cell carcinoma | 84.2 | 0.66 | 0.85 | 0.74 |
| Benign keratosis-like lesions | 59 | 0.55 | 0.59 | 0.57 |
| Melanocytic nevi | 68.8 | 0.90 | 0.69 | 0.79 |
| Melanoma | 78 | 0.47 | 0.76 | 0.58 |
| Vascular lesions | 71.2 | 0.48 | 0.72 | 0.58 |
| Dermatofibroma | 55 | 0.59 | 0.57 | 0.58 |

# *Table 8 ResNet on ISIC 2019 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 65.6 | 0.62 | 0.70 | 0.66 |
| Basal cell carcinoma | 83.4 | 0.73 | 0.86 | 0.79 |
| Benign keratosis-like lesions | 45 | 0.57 | 0.46 | 0.50 |
| Melanocytic nevi | 49.4 | 0.97 | 0.48 | 0.64 |
| Melanoma | 11 | 0.59 | 0.13 | 0.21 |
| Vascular lesions | 96.9 | 0.36 | 0.97 | 0.53 |
| Dermatofibroma | 92.31 | 0.70 | 0.95 | 0.80 |
| Squamous cell carcinoma | 19.51 | 0.83 | 0.24 | 0.38 |
| Unknown | Nan | N/A | N/A | N/A |

# *Table 9 Retrained ResNet on ISIC 2019 and HAM10000 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 70 | N/A | N/A | N/A |
| Basal cell carcinoma | 85.7 | N/A | N/A | N/A |
| Benign keratosis-like lesions | 46.5 | N/A | N/A | N/A |
| Melanocytic nevi | 97.62 | N/A | N/A | N/A |
| Melanoma | 0 | N/A | N/A | N/A |
| Vascular lesions | 0 | N/A | N/A | N/A |
| Dermatofibroma | 75 | N/A | N/A | N/A |

# *Table 10 ResNet on HAM10000 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 36.67 | N/A | N/A | N/A |
| Basal cell carcinoma | 91.43 | N/A | N/A | N/A |
| Benign keratosis-like lesions | 31.82 | N/A | N/A | N/A |
| Melanocytic nevi | 92.53 | N/A | N/A | N/A |
| Melanoma | 69.23 | N/A | N/A | N/A |
| Vascular lesions | 52.17 | N/A | N/A | N/A |
| Dermatofibroma | 62.5 | N/A | N/A | N/A |

#### DenseNet Model Architecture

For a standard deep convolutional neural network, output from one layer is the input to the upcoming layer. However in DenseNet, there is an additional input to each layer, which is coming from the preceding layer and passing the feature map to its subsequent layers. Concatenation is used. The reason why the DenseNet model is used is that feature maps are passed from all the preceding layers to each layer, so the overall network can be compact having fewer channels. Thus DenseNet allows efficiency in computation and memory.

# *Table 11 DenseNet on ISIC 2019 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 38.8 | 0.55 | 0.42 | 0.48 |
| Basal cell carcinoma | 88.1 | 0.70 | 0.89 | 0.78 |
| Benign keratosis-like lesions | 66.5 | 0.62 | 0.64 | 0.63 |
| Melanocytic nevi | 68.7 | 0.93 | 0.68 | 0.79 |
| Melanoma | 70.7 | 0.76 | 0.68 | 0.72 |
| Vascular lesions | 81.4 | 0.48 | 0.81 | 0.60 |
| Dermatofibroma | 65 | 0.76 | 0.65 | 0.70 |

# *Table 12 DenseNet on ISIC 2019 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 69.6 | 0.85 | 1.00 | 0.92 |
| Basal cell carcinoma | 83.2 | 0.88 | 0.97 | 0.92 |
| Benign keratosis-like lesions | 45.2 | 0.84 | 0.86 | 0.85 |
| Melanocytic nevi | 47.8 | 0.98 | 0.75 | 0.85 |
| Melanoma | 9 | 0.90 | 0.88 | 0.89 |
| Vascular lesions | 96.9 | 0.62 | 0.99 | 0.76 |
| Dermatofibroma | 92.31 | 0.76 | 1.00 | 0.87 |
| Squamous cell carcinoma | 29.2 | 0.89 | 0.80 | 0.85 |
| Unknown | Nan |  |  |  |

# *Table 13 Retrained DenseNet on ISIC 2019 and HAM10000 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 90 | N/A | N/A | N/A |
| Basal cell carcinoma | 57 | N/A | N/A | N/A |
| Benign keratosis-like lesions | 43.1 | N/A | N/A | N/A |
| Melanocytic nevi | 94.5 | N/A | N/A | N/A |
| Melanoma | 61.5 | N/A | N/A | N/A |
| Vascular lesions | 39.1 | N/A | N/A | N/A |
| Dermatofibroma | 50 | N/A | N/A | N/A |

# *Table 14 DenseNet on HAM10000 Dataset*

| Class | Accuracy(%) | F1 Score | Precision | Recall |
| --- | --- | --- | --- | --- |
| Actinic keratoses | 66.6 | N/A | N/A | N/A |
| Basal cell carcinoma | 77.1 | N/A | N/A | N/A |
| Benign keratosis-like lesions | 43.1 | N/A | N/A | N/A |
| Melanocytic nevi | 97.62 | N/A | N/A | N/A |
| Melanoma | 0 | N/A | N/A | N/A |
| Vascular lesions | 0 | N/A | N/A | N/A |
| Dermatofibroma | 87.5 | N/A | N/A | N/A |

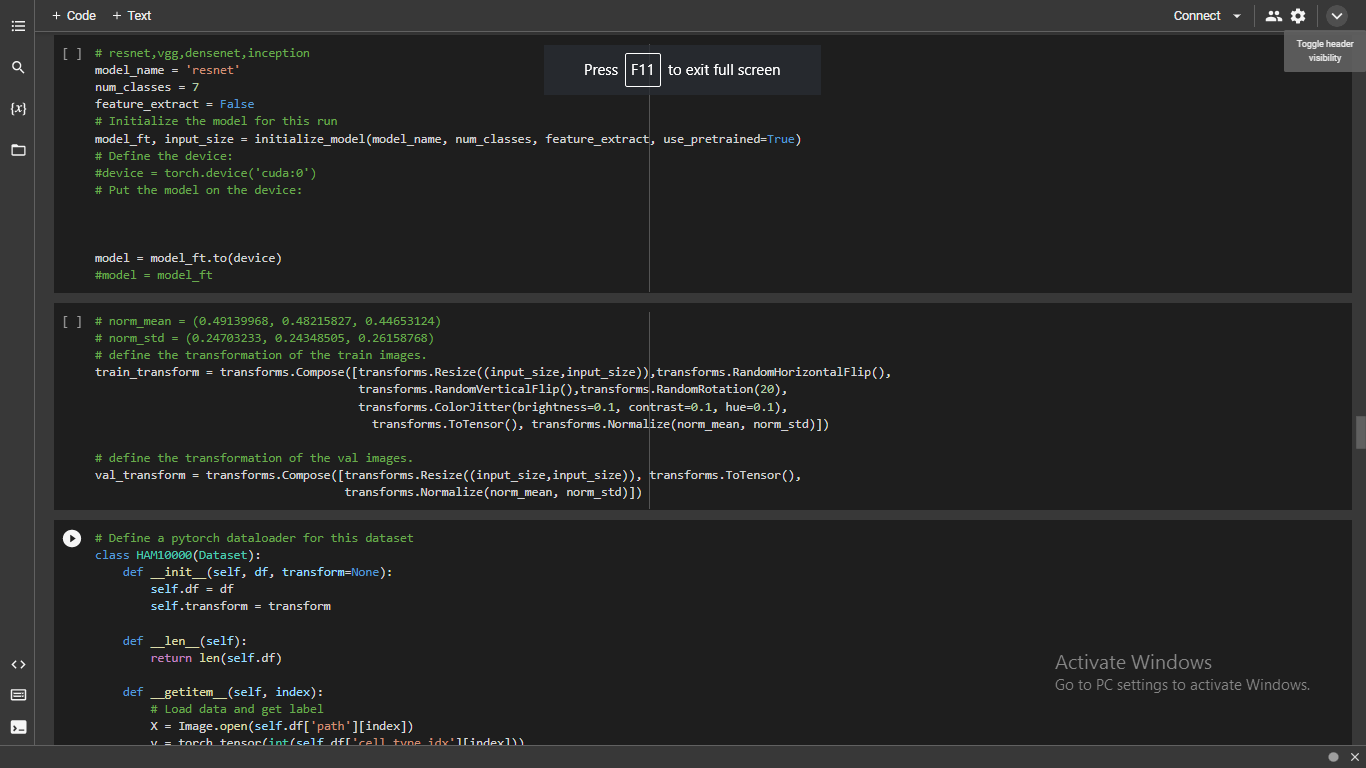
# *Table 15 Classification Results of Retrained models*

| Classifier and Training Algorithm | Dataset | Description | Skin Cancer  Classes | Results(%) |
| --- | --- | --- | --- | --- |
| ResNet50 | HAM10000 & ISIC 2019 | N/A | 7 classes | Validation Accuracy: 90.2% |
| DenseNet | HAM10000 & ISIC 2019 | N/A | 7 classes | Validation Accuracy: 91.2% |

## 

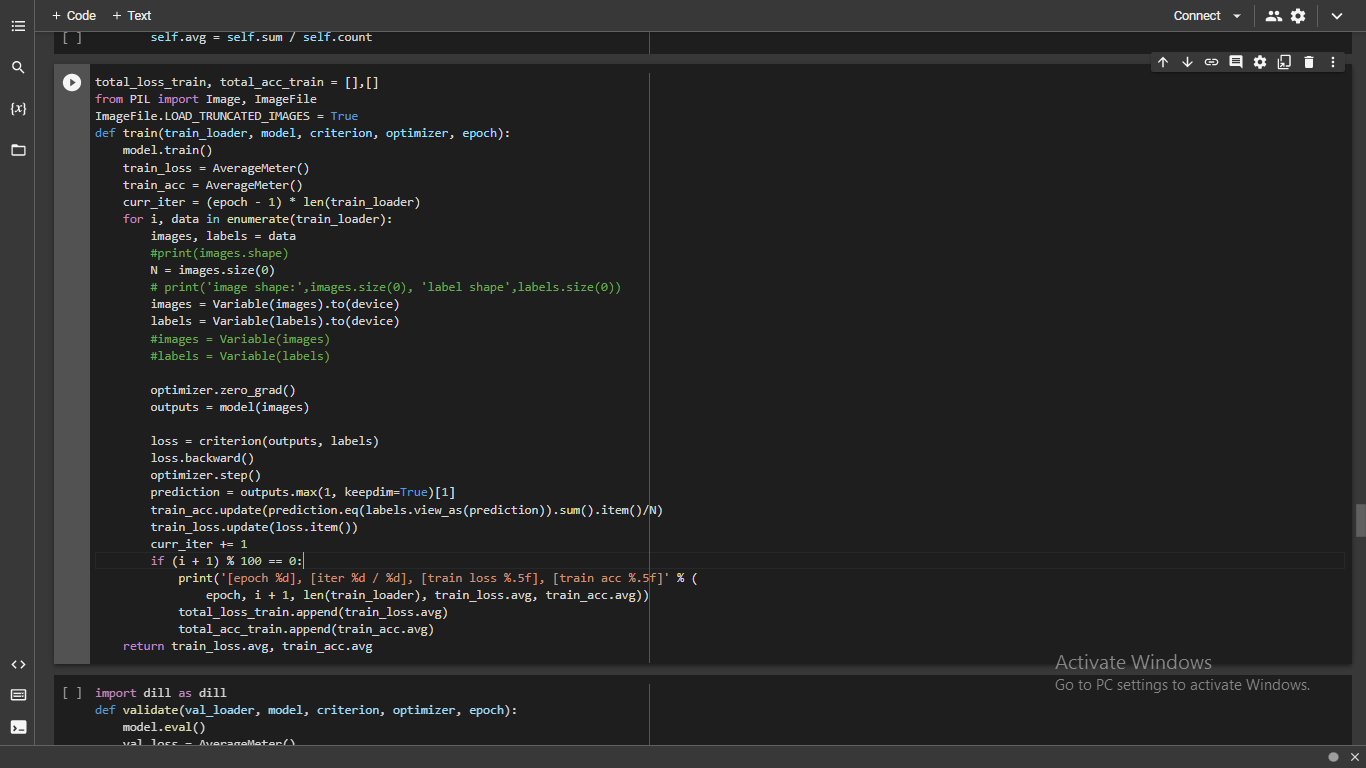
# *Figure 8 validation dataframe and train dataframe*

The Figure 8 shows the data frame of validations and train



# *Figure 9 model initialization and transforms*

The Figure 9 resnet model is initialized, and transforms are created, all the model architectures are using all the same functions except the model initialization, where models are changed as per the requirements



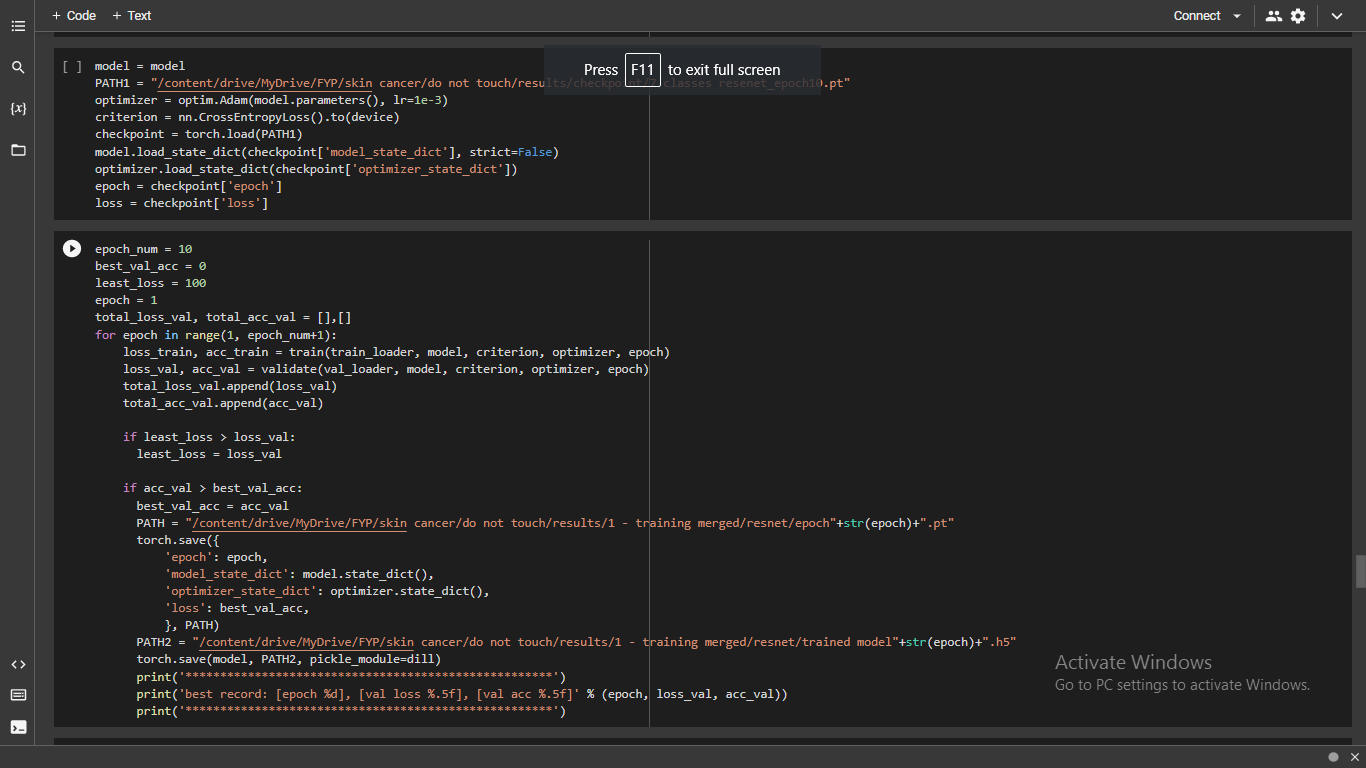
# *Figure 10 train function*

The Figure 10 shows the train function of the model



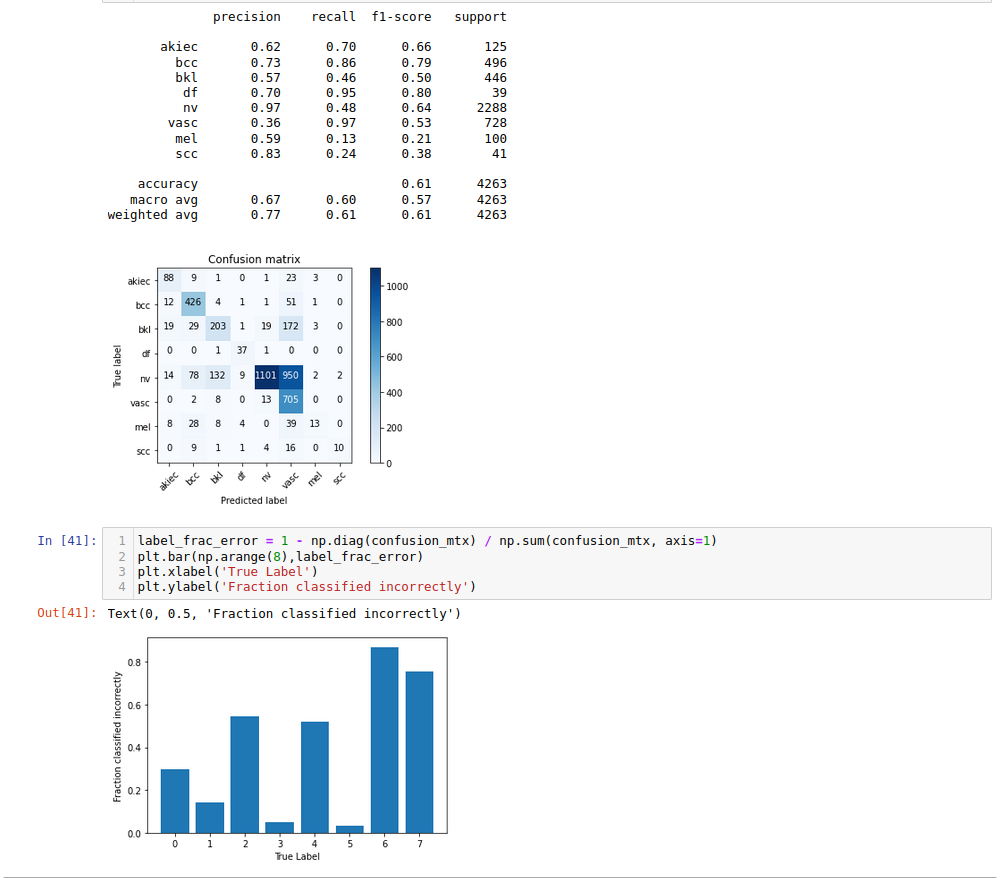
# *Figure 11 validate function*

The Figure 11 shows the validate function of the model

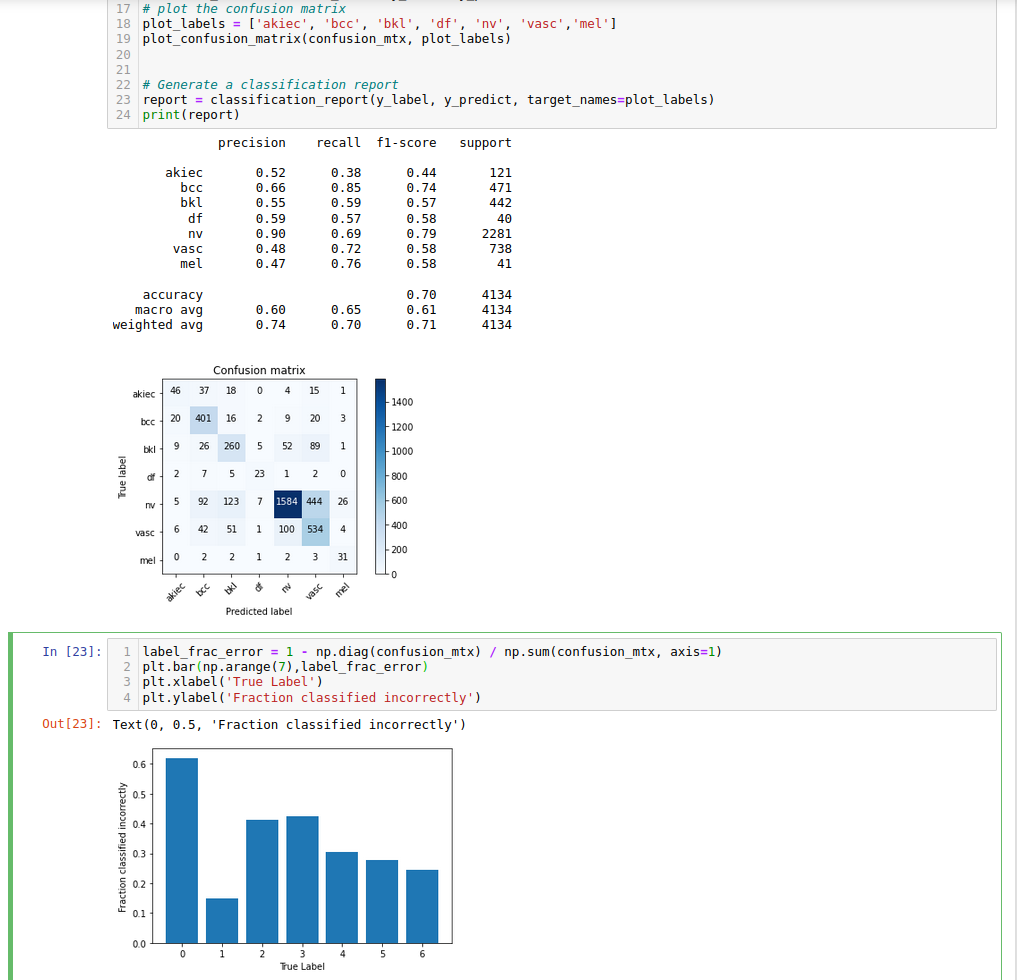


# *Figure 12 train iteration*

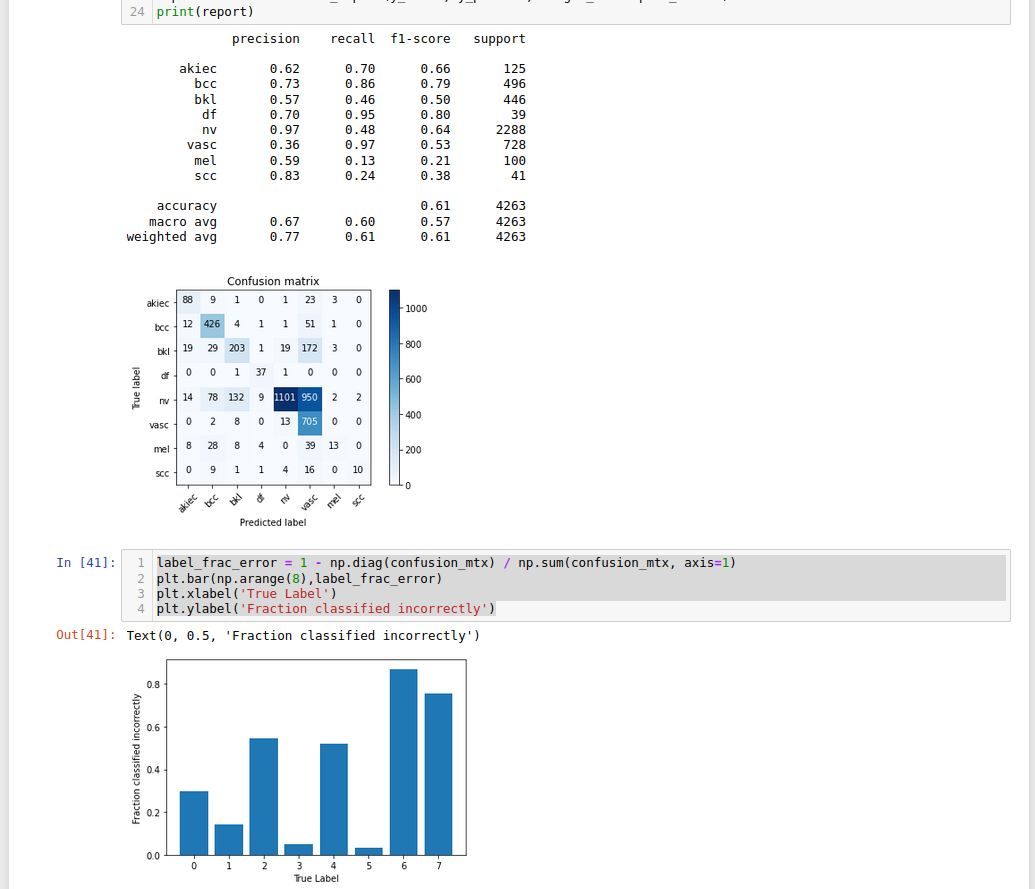
The Figure 12 shows the training iteration



# *Figure 13 Classification report on ResNet model ISIC 2019 (9 classes)*



# *Figure 14 Classification report on DenseNet model ISIC 2019 (9 classes)*



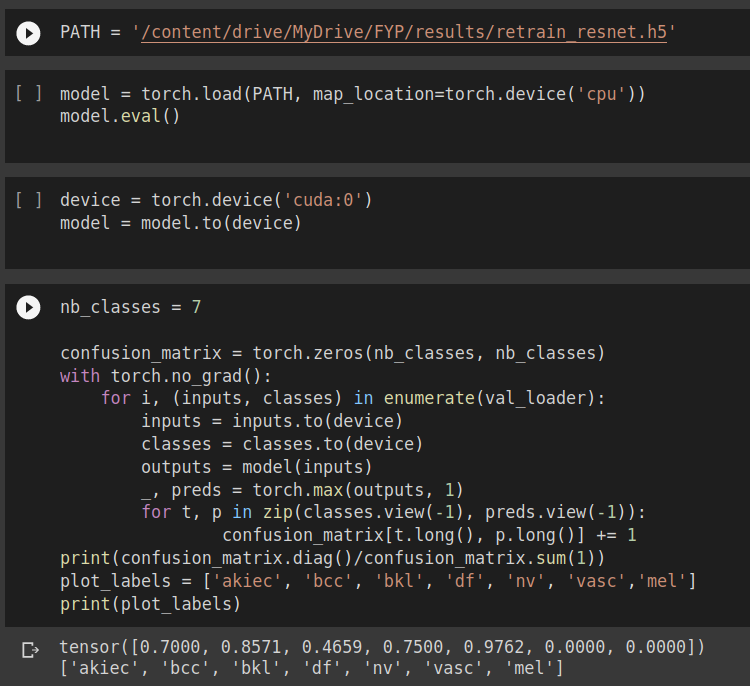
# *Figure 15 Classification report on ResNet model ISIC 2019 (7 classes)*

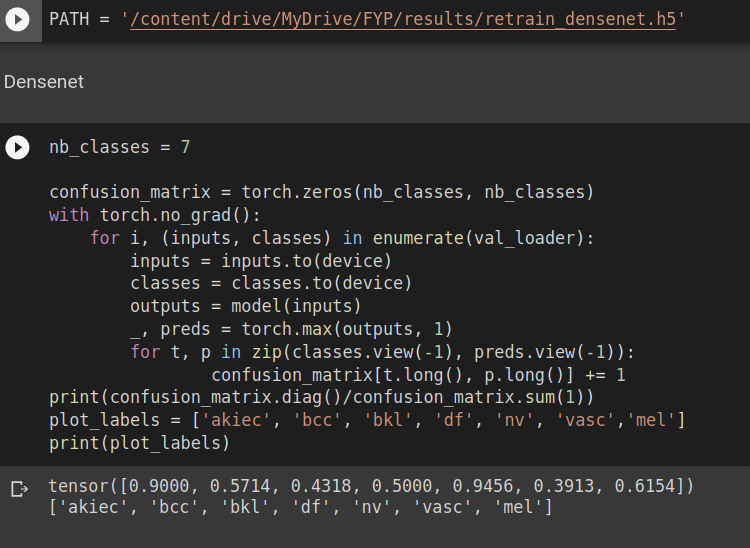
# 

# *Figure 16 Classification report on DenseNet model ISIC 2019 (7 classes)*

# 

# 

*Figure 17 Classification report on retrained ResNet model*



# *Figure 18 Classification report on retrained DenseNet model*

## 

## **7.2 AI DERMASSISTANT WEB BASED SCREENS**

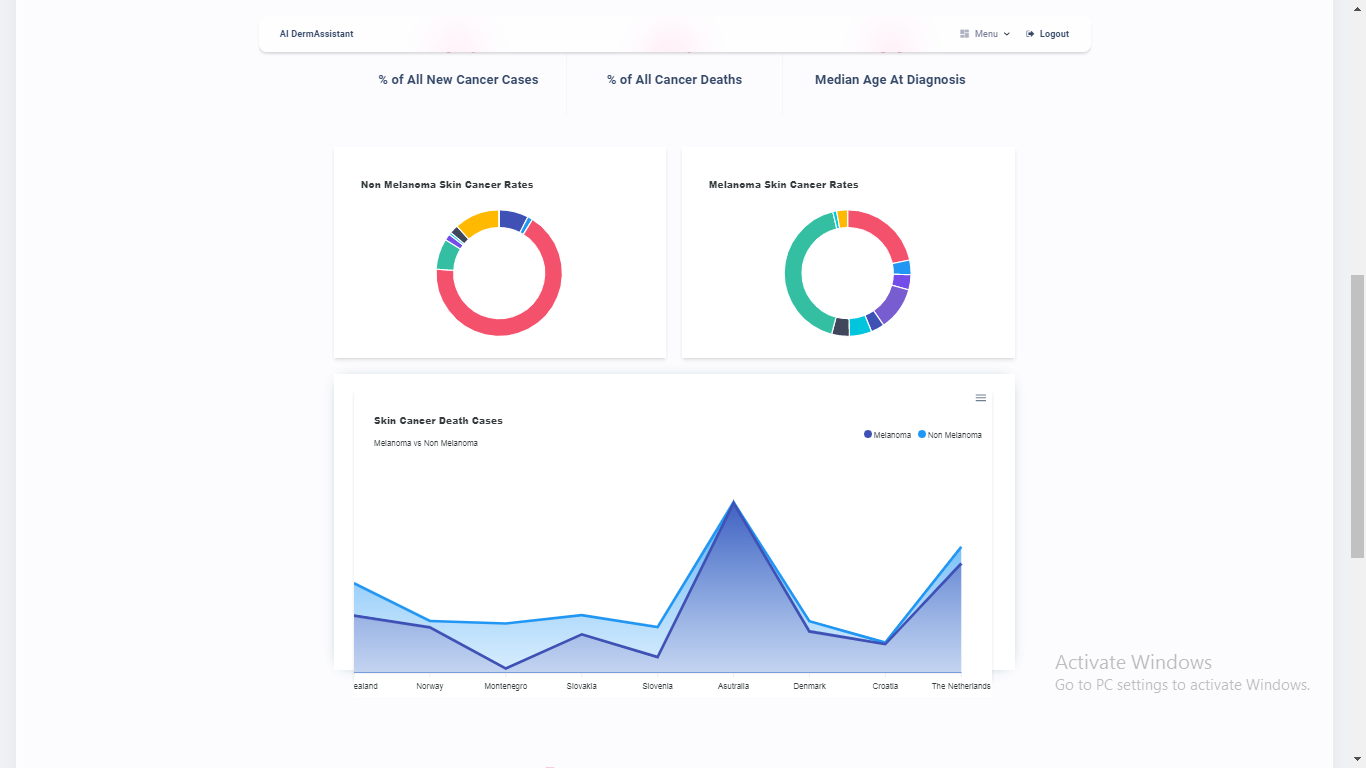
Figure 18, 19, 20, 21, 22 , 23, 24, 25, 26, 27, 28 and 29 presents the screen of the web application AI DermAssistant. Every picture has a description.

# *Figure 19 Register Screen*

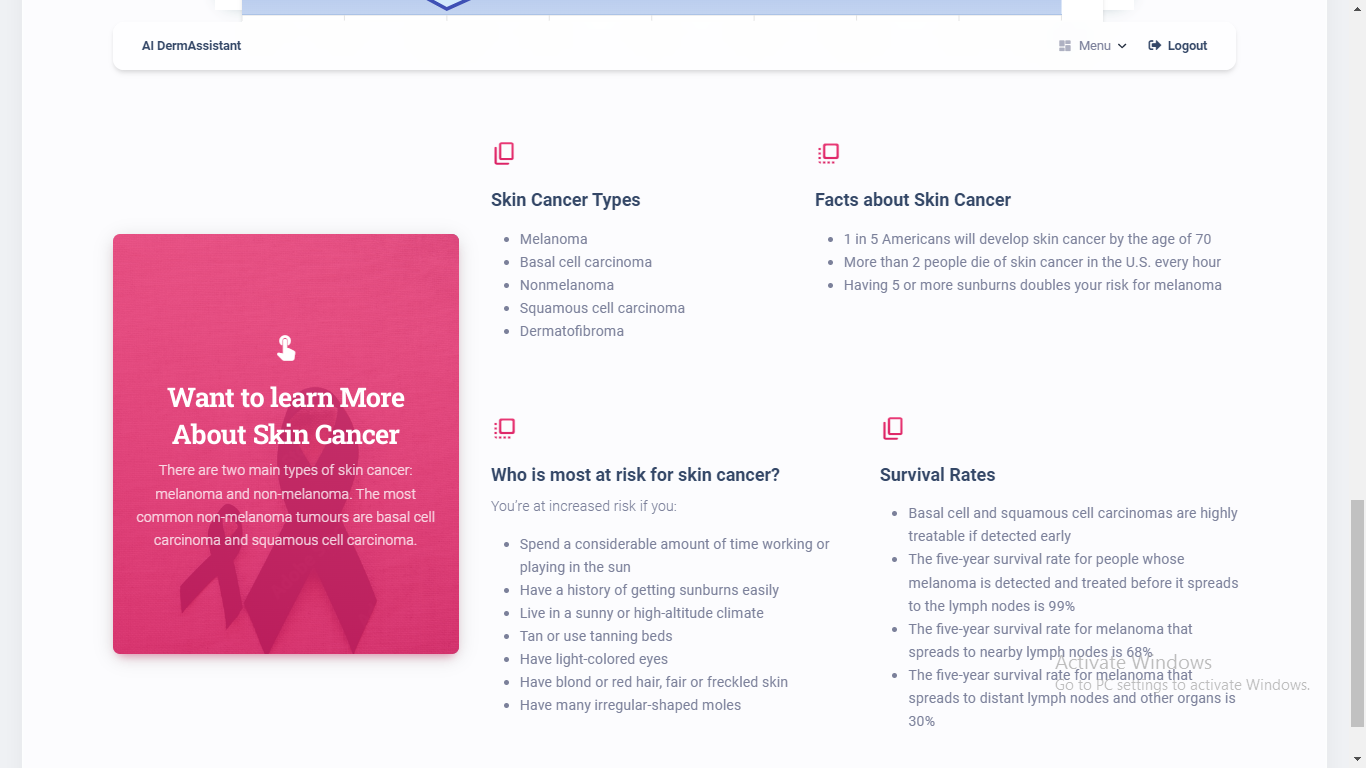


# *Figure 20 Login Screen*

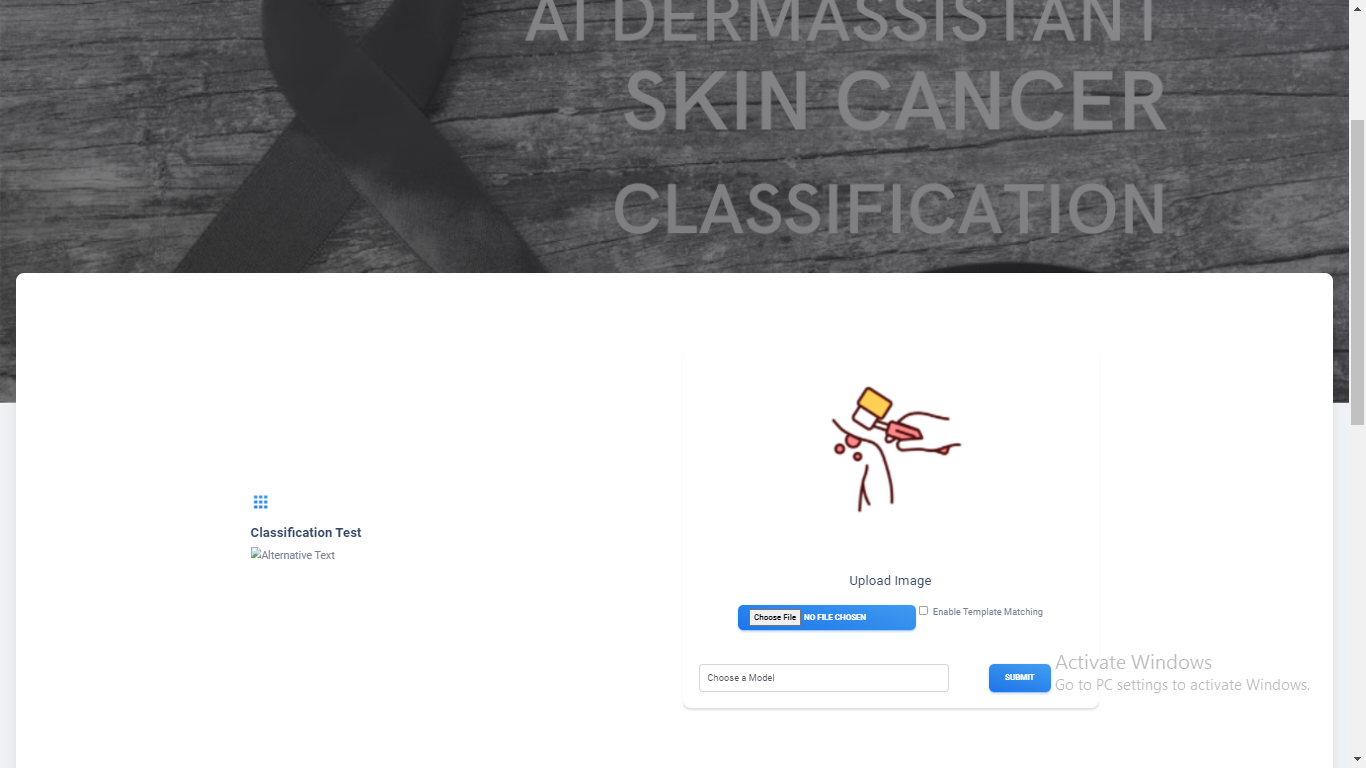
# *Figure 21 Dashboard Statistics Screen*

**

# *Figure 22 Dashboard graphs*

**

# *Figure 23 Dashboard Screen Data*

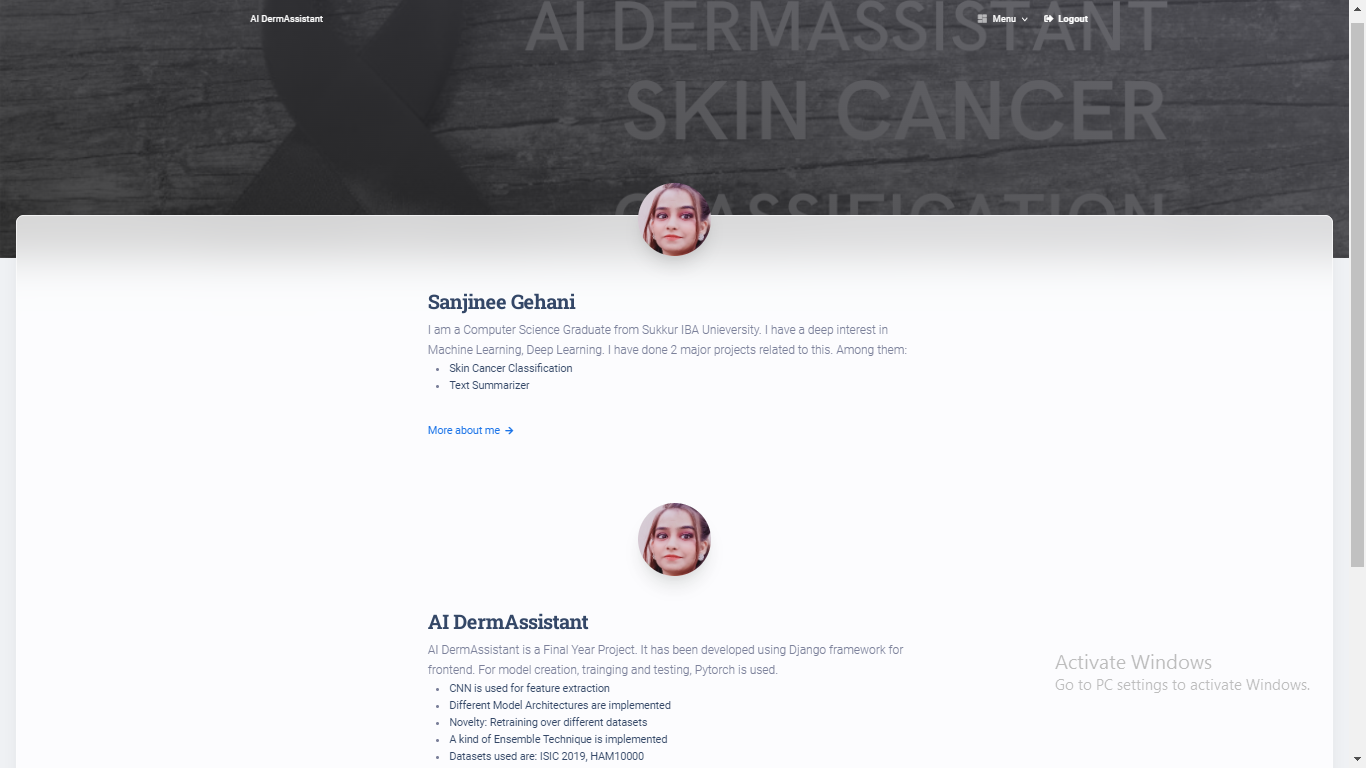


# *Figure 24 Dashboard graphs*

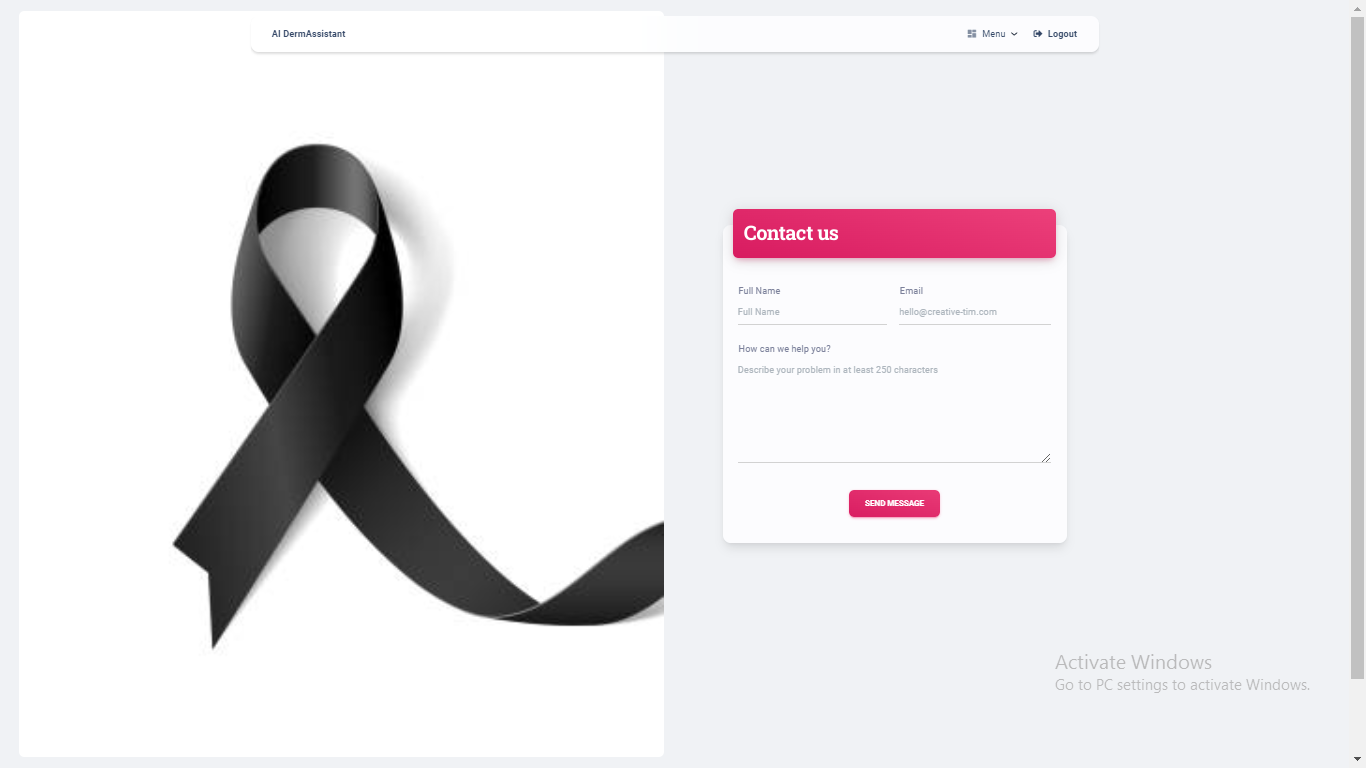
# *Figure 25 Cancer Classification Test Screen Contributors Section*



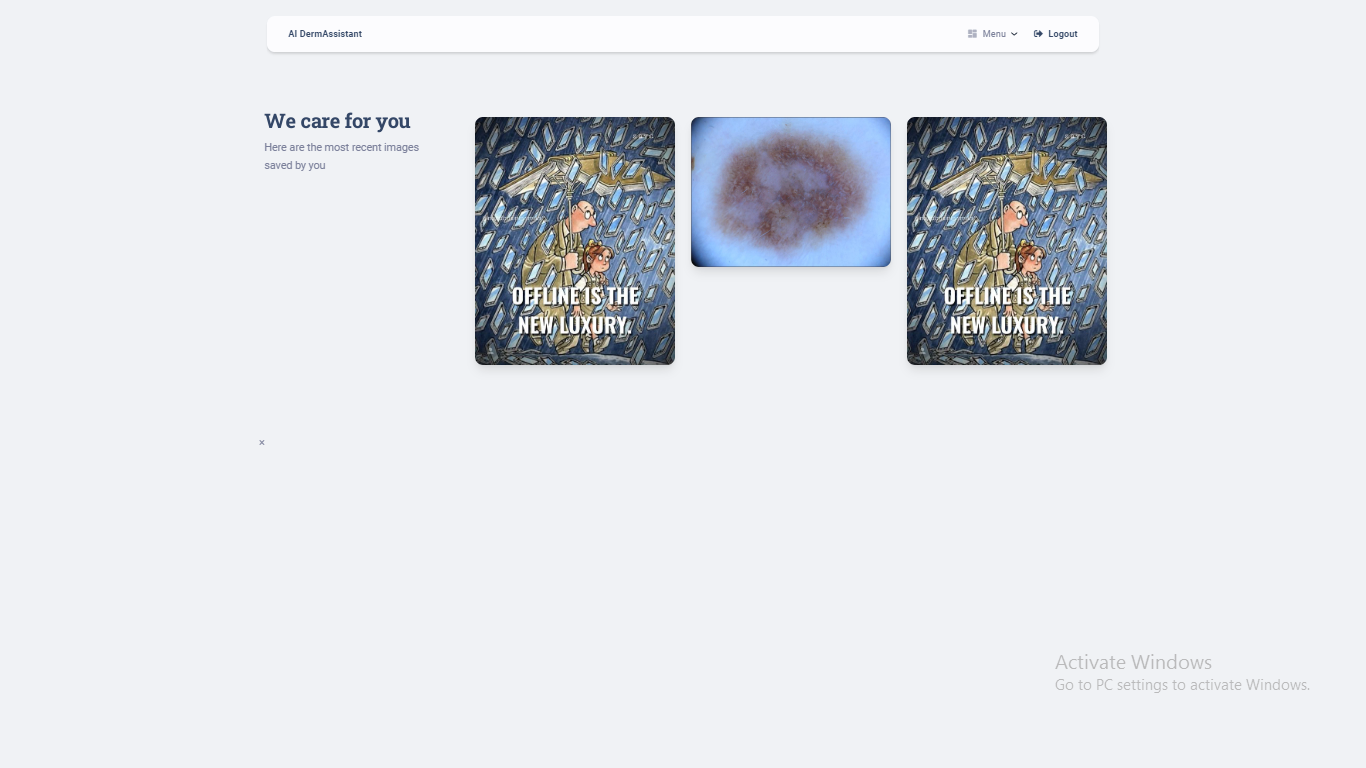
# *Figure 26 Cancer Classification Test Screen Technologies Used Section*



# *Figure 27 Author Screen*



# *Figure 28 Contact Us Screen*



# *Figure 29 Image Gallery*

***Chapter 8***

# **CONCLUSION AND FUTURE WORK**

# **8.1. CONCLUSION**

The AI DermAssistant is a web application that allows the user to upload the lesion images and get the classification results over the lesion. As demonstrated from the implementation and results, it has its foundation for providing the interface to a deep learning model. It can be assumed to be a platform where users can easily upload his lesion image, and get the classification results. User does not have to go for any installation, rather what he only needs to have is the image of the lesion in his device, and an internet connection to access this website. My focus of the work was based on the training of the classification model, where I faced the problem of having insufficient images in the HAM10000 dataset of Skin Cancer, so I performed retraining (kind of ensemble learning) in my project at backend. For that I trained my model initially on the ISIC 2019 dataset and then the same model was trained using the HAM10000 dataset. Another problem that I encountered was that I had 9 classes in the ISIC 2019 dataset, whereas the HAM10000 dataset had only 7 classes of Skin Cancer. So I had to remove the 2 classes from ISIC 2019.

# **8.2. RECOMMENDATIONS AND FUTURE DIRECTIONS**

There is nothing in the universe that is perfect, and there is always scope of improvement in everything we do. AI DermAssistant can also be improved and many new features can be introduced and implemented into the system and also the existing features can be improved.

As per now, the admin should be assigned more privileges. The images can also be captured at runtime. The images can be saved only if the user wants to save. The images need to be deleted using the admin dashboard. The classification results should be saved.

***Chapter 9***

# **REFERENCES**

1. (*Xie, F.; Fan, H.; Li, Y.; Jiang, Z.; Meng, R.; Bovik, A. Melanoma Classification on Dermoscopy Images Using a Neural Network Ensemble Model. IEEE Trans. Med. Imaging 2017*, n.d., )
2. (*Jaleel, J.A.; Salim, S.; Aswin, R. Artificial Neural Network Based Detection of Skin Cancer. Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 2012, 1, 200–205.*, n.d., )
3. *(“Choudhari, S.; Biday, S. Artificial Neural Network for SkinCancer Detection. IJETTCS 2014, 3, 147–153” )*
4. (*Aswin, R.B.; Jaleel, J.A.; Salim, S. Hybrid Genetic Algorithm: Artificial Neural Network Classifier for Skin Cancer Detection. In Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (IC*)
5. (*Ercal, F.; Chawla, A.; Stoecker, W.V.; Hsi-Chieh, L.; Moss, R.H. Neural Network Diagnosis of Malignant Melanoma from Color Images. IEEE Trans. Biomed. Eng.*)
6. (*Mahmoud, K.A.; Al-Jumaily, A.; Takruri, M. The Automatic Identification of Melanoma by Wavelet and Curvelet Analysis: Study Based on Neural Network Classification. In Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS)*)
7. (*Cueva, W.F.; Munoz, F.; Vasquez, G.; Delgado, G. Detection of Skin Cancer ”Melanoma” through Computer Vision. In Proceedings of the 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Cusco, Peru*)
8. (“DeVries, T.; Ramachandram, D. Skin Lesion Classification Using Deep Multi-Scale Convolutional Neural Networks”)
9. (“Yu, L.; Chen, H.; Dou, Q.; Qin, J.; Heng, P.-A. Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks. IEEE Trans. Med. Imaging”)
10. (*Mahbod, A.; Schaefer, G.; Wang, C.; Ecker, R.; Ellinge, I. Skin Lesion Classification Using Hybrid Deep Neural Networks. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton*)
11. (*Mendes, D.B.; da Silva, N.C. Skin Lesions Classification Using Convolutional Neural Networks in Clinical Images*)
12. (*Dorj, U.-O.; Lee, K.-K.; Choi, J.-Y.; Lee, M. The Skin Cancer Classification Using Deep Convolutional Neural Network. Multimed. Tools Appl*)