SKIN CANCER DETECTION
A PROJECT REPORT

Submitted by

ADIGOPULA.BALAJI(37130006)

Under the guidance of

Mrs .T.VIJAYASHREE..,M.E.,(Ph.D)

Assistant Professor

Department of ECE

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

CHENNAI – 600119

APRIL-2021

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report DETECTING SKIN CANCER is the bonafide work of **BALAJI. ADIGOPULA (37130006)** who carried out the project work under my supervision during the academic year 2**020-2021.**

DR. T.RAVI..M.E.,Ph.D

PROF&HEAD

DEPARTMENT OF ECE

Mrs. T.VIJAYASHREE..M.E.,(Ph.D)

SUPERVISOR

DEPARTMENT OF ECE

INTERNAL EXAMINER

Name:_____

Designation: _____

EXTERNAL EXAMINER

Name: _____

Designation:_____

Project Viva - voce conducted on _____

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
GLCM	Gray Level Co-occurrence Matrix

FFNN	**Feed Forward Neural Network**

ACKNOWLEDGEMENT

First and foremost, I would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it. We wish to express our deep sense of gratitude from the bottom of our heart to our guide **Mrs. T.VIJAYASHREE.,M.E.,(Ph.D), Assistant Professor**, **Electronics and communication Engineering,** for his motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

I would like to extend my heartfelt gratitude to **Dr. T.RAVI.,M.E.,Ph.D, Professor& Head, Department of Electronics and communication Engineering** for her valuable suggestions and support in successfully completing the project.

I wish to thank my Project Coordinator and Panel members for keeping our project in the right track. I would like to thank all the teaching, technical and non-technical staff of Department of Computer Science and Engineering for their courteous assistance.

I thank the management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, I would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

BALAJI.ADIGOPULA(37130006)

ABSTRACT

Now a day's skin cancer is a major problem human beings are facing, To recognize skin cancer new methodology for the diagnosing skin cancer by images of dermatologic spots using image processing presented. Currently in skin cancer one the most frequent diseases humans. This methodology based Fourier spectral analysis using filters such classic, inverse and to k-law nonlinear. The sample images are obtained by a specialist as an replacement spectral to technique is developed and quantitative measurement in the complex pattern found cancerous skin spots. Finally in which spectral index calculated get a variety spectral indices defined carcinoma . Our results show confidence of level in 95.4%. carcinoma mainly occurs thanks to exposure of sunlight. Ozone is depletion and maintained chemical exposures in other factors involved precipitating carcinoma . Mutations of p53 gene involved UV- induced as carcinogenesis. P53 gene acts vital development in SCC.

Skin Cancer alarming is disease for mankind , the need early diagnosis the skin cancer are increased due to the rapid climb rate of Melanoma skin cancer , its high treatment Costs , and death rate. The cancer cells are detected manually and it takes time to cure in most of the cases. This project proposed a man-made carcinoma detection system using image Processing and machine learning method. The features of the affected skin cells are extracted after the segmetation of the pictures using feature extraction technique. A deep learning based method Convolutional neural network classifier is employed for the stratification of the extracted features..

Skin Cancer is an alerting issue and it must be detected as early as possible. The diagnostic is a manual process that is time consuming as well as expensive. But, today's world science has become advanced by using machine learning make easy detecting cancerous cells to the machine learning specially convolution neural network is employed to detect cancerous cell more in quickly , and to efficiently.

CHAPTER - 1

INTRODUCTION

Cancer forms when healthy cells in change in and grow out control, forming an the called a the tumor. A tumor can cancerous r benign. A cancerous tumor is malignant, meaning that grow and spread over other parts of the body. As there bengun as a tumour means that tumor can be grow but won't spread.

Doctors diagnose carcinoma additional than 3 million Americans annually , making in foremost common sort of cancer. If carcinoma is found early, it can usually be treated with topical medications, procedures wiped out offic a dermatologist, or outpatient surgery. A dermatologist may doctor who focuses diseases and conditions of the skin. As an result, carcinoma is liable for but 1% all cancer deaths.

In some cases, carcinoma could also more advanced in need management to a multidisciplinary team to always a dermatologist, surgical and oncologist, radiation oncologist, and to a medical oncologist. These are in doctors meet there patient, and together they're going recommend the simplest path forward treat cancer. In such \\ instances, the surgical oncologist will recommenda surgery be performed operating room because the procedure treat the cancer too extensive for an office setting.

CHAPTER – 2

2.1 LITERATURE REVIEW

S.NO	PAPER TITLE	AUTHOR NAME	PUBLICATION DETAILS	ISSUES ADRESSED
1	Face Recognition	**Ahmad Tolba Ali El-Baz Ahmed A El-Harby**	January 2005	FACE RECOGNITION
2	Skin Disease Recognition Method Based on Image Color and Texture Features	**John Mitchell**	Received10 Apr 2018 Article ID 8145713	Disease recognition
3	Methodology for diagnosing of skin cancer in images of dermatologic sports by spectral analysis	**Josué Álvarez-Borrego**	DOI: 10.1364/BOE.6.003876	Diagnosing skin cancer
4	A REVIEW ON SKIN CANCER	**S. Ramya Silpa V. Chidvila**	DOI: 10.7897/2230-8407.04814	Review of skin cancer
5	Public Opinion Polls	**Rachel Macreadie**	July 2011 DOI: 10.13140/2.1.2546.4646 Affiliation: Parliament of Victoria	POLLS
6	Opinion research	**Paul J. Lavrakas**	2008 Encyclopedia in of Survey such Research Methods	public opinion

2.2 EXISTING SYSTEM

- This project may be a method for the detection of Melanoma carcinoma using the Image as processing tools.
- In this input the system is skin lesion image then applying in image processing techniques, it analyses conclude about the presence of carcinoma .
- The Lesion is Image to analysis tools checks as varied Melanoma in parameters, Color, Area perimeter, diameter to texture, size to shape analysis for image segmentation and the feature stages.
  The extracted feature parameters that are wont to classify image as Non Melanoma and also Melanoma cancer lesion.

2.3 PROPOSED SYSTEM

- This project may be a method for the detection of Melanoma carcinoma using Image processing tools.
- In this input the system is that skin lesion image then applying image processing techniques, it analyses conclude about the presence carcinoma .
- In Lession to Image analysis tools checks in the varied Melanoma parameters, Color, Area perimeter, diameter etc texture, size and shape analysis for image segmentation and the feature stages.
- The extracted to feature parameters wont of classify the image as Non Melanoma and Melanoma cancer lesion. Through poll we are getting to collect patient after treatment.

CHAPTER-3

METHODOLOGY

3.1 DATA COLLECTION :

- Dataset used for this are extracted from kaggle towards skin cancer Detection .
- It consists of 10000 images of skin cancer.
- The training data consists of 8000 images and testing data consists of 2000 images.

Fig 3.1 IMAGES OF SKIN CANCER DATASET

3.2 IMAGE PREPROCESSING :

Image preprocessing is done by using OPEN CV and NUMPY.

3.2.1 OpenCV :

- OpenCV-Python library of Python bindings in designed unravel computer vision problems.
- OpenCV-Python makes use Num py, by which may highly optimized library numerical operations a MATLAB-style syntax.
- All tin Open CV array are structures converted a and from Num py arrays.
- This also makes it easier to integrate other a libraries is that use Num py SciPy and Matplotlib.
- OpenCV to be capable image analysis and processing.

3.2.2 NumPy :

Import- numpy :as np

- NumPy, that stands Numerical Python, be a library consisting of multi_dimensional as array objects and set a routines for processing those arrays.
- Using as Num Py, mathematical and logical on operations are arrays in often performed.
- The array object in NumPy is named ndarray, it provides tons of supporting functions that make working with nedarray very easy.
- NumPy is an open-source numerical Python library. Num Py a extension o Numeric and Num array.
- Num py contains random number generators. NumPy may wrapper around library implemented in C.
- Pandas is objects reply heavily NumPy objects. Essentially, Pandas extends Numpy.

3.3 IMAGE SEGMENTATION & FEATURE EXTRACTION :

Image segmentation is a process of dividing image into regions or categories. In the dermoscopic images two types of fabric things first normal skin and second is lesion area so here we have done segmentation with Otsu thresholding technique. Using Texture-Based segmentation extracting the features from the image. GLCM (Gray Level Co-occurrence Matrix) is the statistical method examining the spatial relationship between the pixel. This technique works by creating the co-occurrence matrix were to calculate the frequency of occurrence of a pixel with the grey-level value is adjacent to a pixel with grey-level value j in any given direction and selected separating distance The GLCM matrix gives four statistics Correlation, Contrast, Energy, Homogeneity. There some problem in segmentation of dermoscopic images due to the contrast of images like under segmentation and over-segmentation so we are concentrating on segmentation based on texture features.

3.4 IMAGE CLASSIFICATION :

Deep learning is one of the best techniques for image classification. Based on the texture features we are training the dataset for classification. Here first we are giving Extracted feature to the Neural network for checking performance of image classification then we are using CNN (Convolutional Neural Network) it is one of the deep learning techniques for classification, Dermoscopic images classification is done in 7 classes .Melanocytic nevi','Melanoma','Benign keratosis','Basal cell carcinoma', 'Actinic keratoses', 'Vascular lesions', ' Dermatofibroma ' it is done by using automated extracted features by CNN images. In this step, we are passing Preprocess Images to the CNN classification.

CHAPTER - 4

SYSTEM REQUIREMENTS & LIBRARIES

4.1 SOFTWARE REQUIREMENTS :

Operating System : Windows, Mac OS, Linux

Application : Spyder with Python 2.0 or above

4.2 LIBRARIES USED :

In Skin Cancer Detection we use some libraries in python. The list of libraries are :-

- ✧ TensorFlow
- ✧ Pandas
- ✧ NumPy
- ✧ OpenCV
- ✧ Keras

4.2.1 TensorFlow :

- TensorFlow is a free and open-source software library for machine learning.
- It are often across the range of tasks but features a particular specialise to training on inference of deep neural in networks.
- Tenso r flow may a symbolic math library supported data flow differentiable with programming.
- GPU/CPU computing where an equivalent code are often executed to do both architectures .
- High on scalability computation across in machines and large data sets import tensorflow

```
import tensorflow as tf
```

4.2.2 Pandas :

- Pandas is a popular Python a library on data analysis.

- It directly in related Machine Learning. As all we have the dataset to be prepared before the training.
- In this case, Pandas are handy it as an developed \\to data extraction and preparation.
- It provides the implementation is high-level find data structures a wide variety tools for data analysis.
- It provides many methods for as groping, combining and filtering data.

```
import pandas as pd
```

4.2.3 NumPy :

- NumPy means for Numerical Python, in a library of multidimensional array an objects to collection of routines for processing those arrays.
- Using Num Py, mathematical and logical operations we in arrays can be performed.
- Num py to open-source numerical Python library.

```
import numpy as np
```

4.2.4 OpenCV :

- OpenCV-Python to have library Python bindings designed and s computer vision problems.
- Open CV is capable in image analysis and processing.

4.2.5 Keras :

- Keras is a popular Machine Learning library for Python to do.
- It high-level neural networks (API) capable in the running top on Tensor Flow, CNTK, a Theano.
- Keras makes really to ML beginners an build a design a Neural Network.
- One in best thing about Keras that allows easy to fast prototyping.

```
from tensorflow import keras
```

CHAPTER - 5

SYSTEM ARCHITECTURE

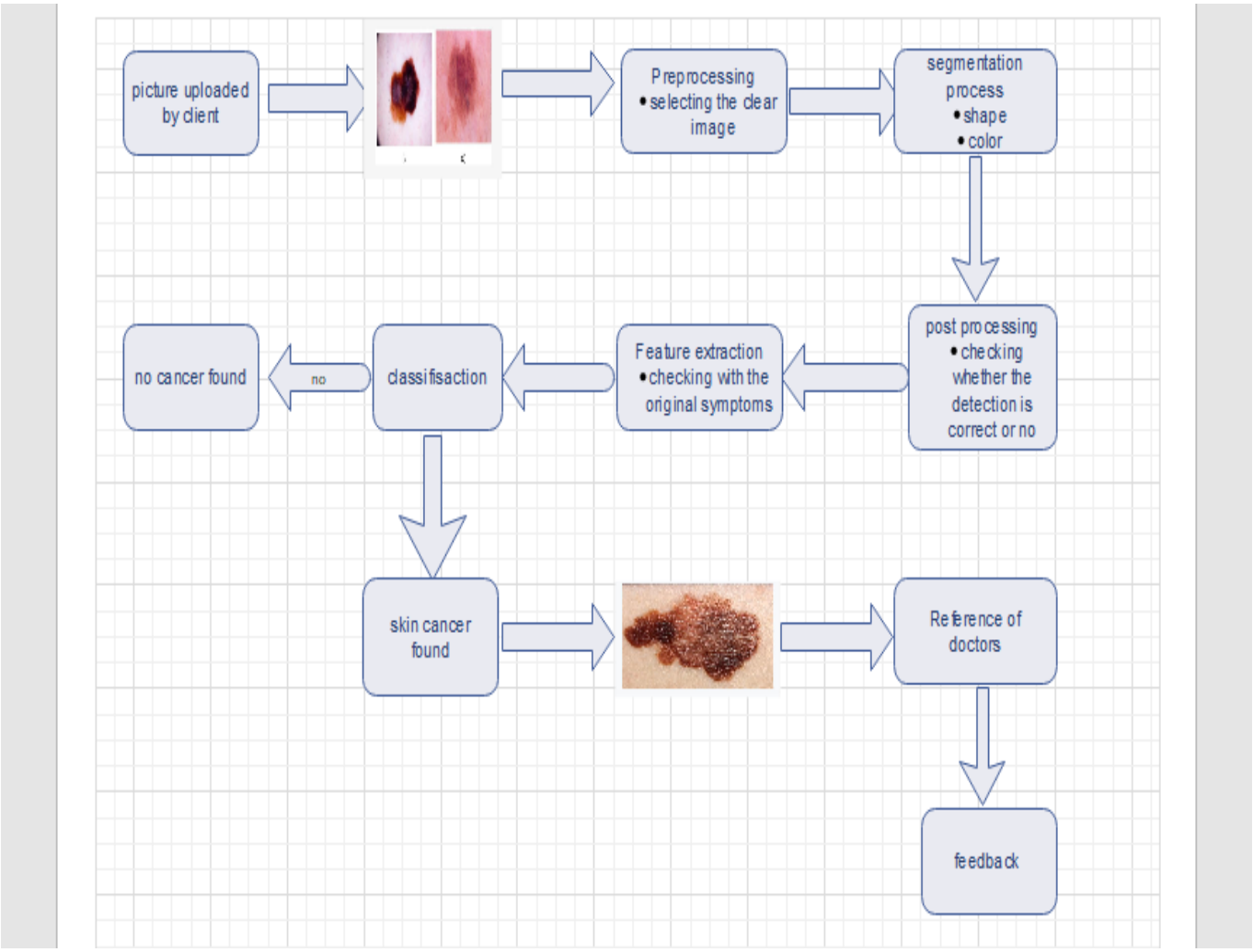


Fig 5 system architecture diagram

In this figure of system architecture diagram we have clearly explained the steps for detecting 7 types of skin cancer. First step comes here is taking picture from the client or customer for detecting. After this next step is preprocessing which is used to convert the picture to gray scale and reshaping is also done and the next step is segmentation process in which the shape and color of the symptom or the patch will be identified. Next step is post processing in which the detections done in the before steps are correct or not, after his feature extraction is done in which the symptoms given in the picture by client is compared with the original cancer symptoms. Next step here comes is classification in which the website gives whether it is cancer or not.

CHAPTER - 6

MODULES

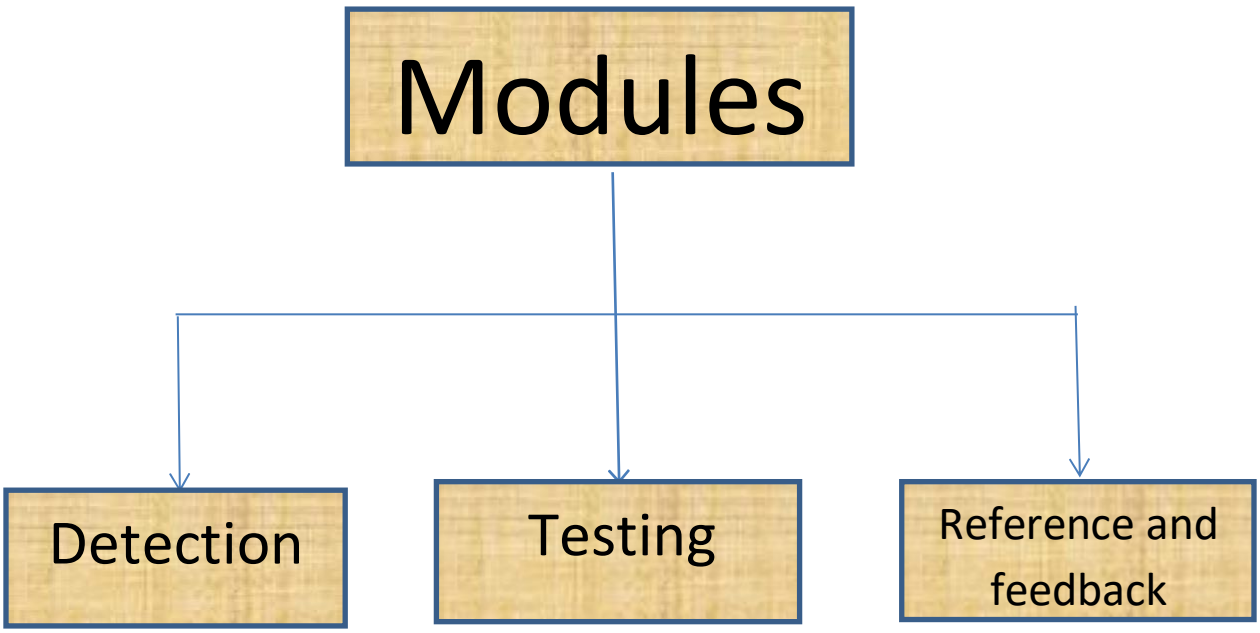


Fig 6 Modules

We have 3 modules in Skin Cancer Detection . They are :-

- ✧ Detection
- ✧ Testing
- ✧ Reference & Feedback

6.1 DETECTION :

Detection module used them detect the image of skin cancer. In this we detect images from skin cancer by using " FEED FORWARD NEURAL NETWORK ALGORITHM " .

- ➢ A feed forward neural network have bimologically inspired by classification which algorithm. It consist of number of simple to neuron-like as processing in units, organized layers. Every unit in a layer connected with in the units in the previous layer. This is they are called feedforward neural networks.

- The feed forward neural network is the in first and simplest type of artificial neural network devised. In the network, the information in one direction—forward—from a input nodes, through the hidden to nodes and to the output nodes. There non cycles in loops inthe network.
- Two basic feed-forward neural networks (FFNNs) created using TensorFlow in deep learning library in Python.
- Steps required build an simple feed-forward neural network to Tenso r Flow by explaining each step details. For before actual building an neural network, some preliminary steps recommended to discussed.

The summarized steps are as follows:

1. Reading the training data (inputs and outputs)
2. Building to connect an neural networks layers
3. Building a loss function to assess the prediction error
4. Create the training loop for training network and updating parameters
5. Applying some testing data to assess the network prediction accuracy

This module briefly introduces the core concepts employed in modern convolutional neural networks, with an emphasis on methods that have been proven to be effective for tasks such as object detection and semantic segmentation. Basic network architectures, common components and helpful tools for constructing and training networks are described.

6.2 TESTING :

Testing module is used to test and predict the image of skin cancer. For testing we used " Evaluation function from keras " .

- Evaluation a is process during development to the model check whether this model fit for given problem and corresponding data.
- Keras provides a function, evaluate which does evaluation of the model.

- There are three main arguments,

1. Test data
2. Test data label

3.verbose - true r false

Keras separate an portion of your training data to validation of dataset and evaluate that performance of your model on validation dataset to each epoch. You can do this by setting the validation_split argument on the fit() function to a percentage of the size of your training dataset .

6.3 REFERNCE & FEEDBACK :

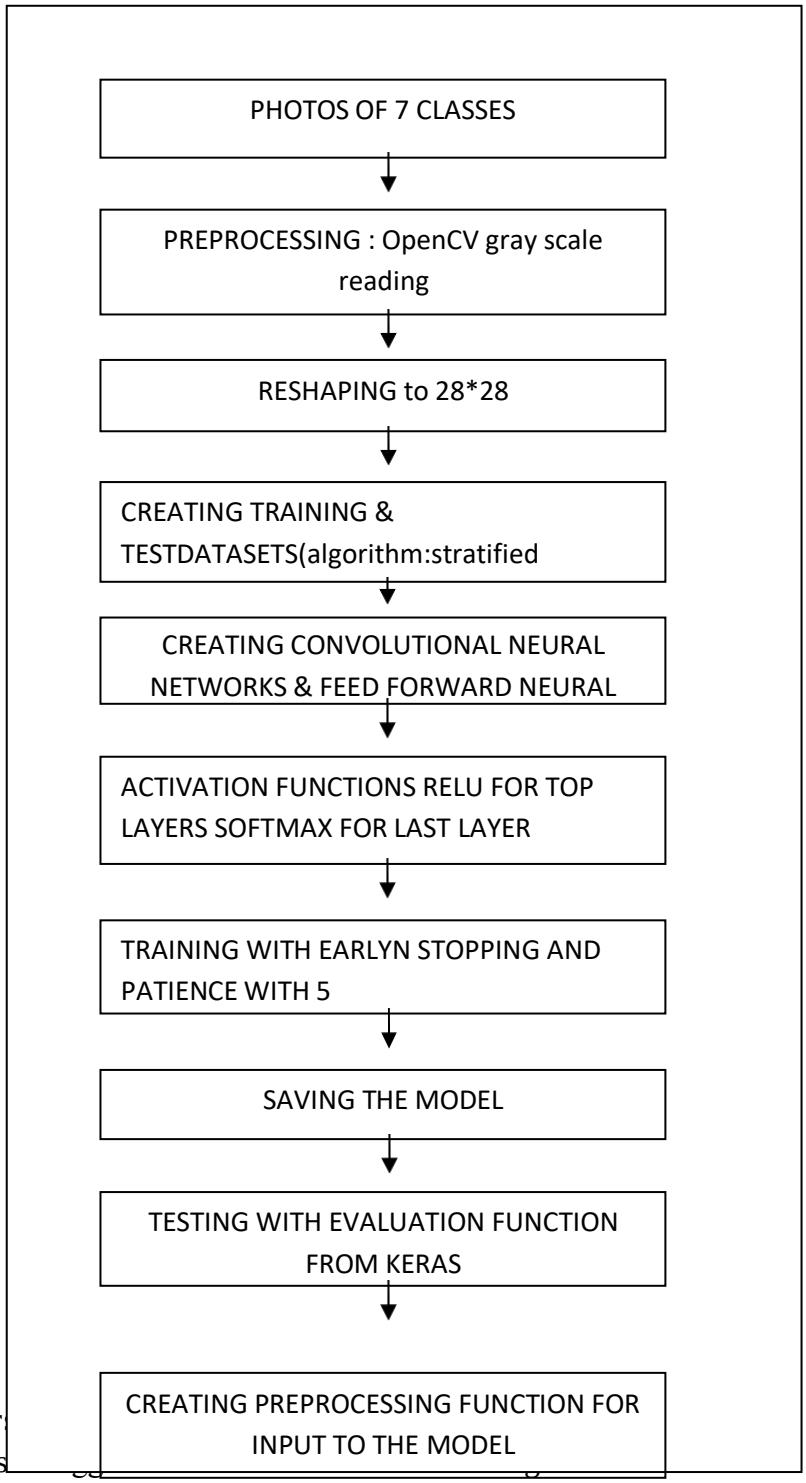
We have used an website named as 'AidaForm' for creating feedback form.

- This helps to create an form and generates an hyperlink so that we can send it to our customers via mail or we can directly paste in our website.
- AidaForm let do that is a blank template in we the add form elements with as simple on drag-and-drop in motions. Plan with in out what to evaluate and fill custom.
- Review response in the summaries as if statistics with average to ratings in frequently yes chosen to options gain why insights and improve.
- Export responses to data sheets of Excel for deeper evaluation and there feedback data efficiently.

CHAPTER-7

WORKING

Here comes the flowchart of this project skin cancer detection:



In any project fir from the famous website known as have taken 8000 images training and the a remaining images for testing .so this paroject mainly have 7 clases of classification for skin cancer:

7.1 PHOTOS OF 7 CLASSES:

As our project classifies 7 types of skin cancers first we have to collect the sample images.so this step helps in collecting the images.

7.2 PREPROCESSING :

After taking the image from the customer , the image may or mayn't have clarity so to avoid this problem and to make the program to classify the image we are using opencv gray scale to make it clear.

7.3 RESHAPING :

Now comes the other task , that every image given by the customer will not be in the same size so to avoid this we have done reshapping all the images to 28*28 size. So this makes the program to run successfully.

7.4 CREATING TRAINING & TESTDATASETS:

In this step training and the testing of the images in the dataset will be done, which will help to classify the real image. This is done using stratified shuffle split algorithm.

Stratified ShuffleSplit cross-validator:

Provides train or test indices to separate data in train/test sets.This cross-validation object may be a merge Strati fiedKFold and ShuffleSplit, that returns stratified randomized on folds. The folds made by preserving share of samples for each class.

7.5 CREATING CONVOLUTIONAL NEURAL NETWORKS & FEED FORWARD NEURAL:

In this step creation of Convolution neural network(CNN) and Feed forward neural is done for detection of cancer in the pictures provided by the customer.

7.6 ACTIVATION FUNCTIONS :

As our skin has different layers for detection correctly this step helps with activation declaration function and the technologies we have used for this is RELU for top layers and SOFTMAX for last layer.

7.7 TRAINING WITH EARLYN STOPPING AND PATIENCE WITH 5:

The training is done by EARLYN stopping and patience has been done by 5.

7.8 SAVING THE MODEL:

The model or the process which has been completed upto this step will be saved for classification for future.

7.9 TESTING WITH EVALUATION FUNCTION:

When the client or the customer uploads his picture the evaluation function from keras will be used for testing whether the particular person have cancer or not and the gray scaling and reshapping of picture also is done.

7.10 CREATING PREPROCESSING FUNCTION FOR INPUT TO THE MODEL:

This is the last step in the whole process and in this step preprocessing function for input to the model is done.And the final output will be displayed on the webpage.So this will be the working process of this project.

CHAPTER-8

RESULT

Here is the output screenshot where we can know whether a person has cancer or not

This picture is for detecting Melanoma cancer which is one of the type of skin cancer.

Fig 8.1 detection of melanoma

This picture is for detecting Melanoma cancer which is one of the type of skin cancer.

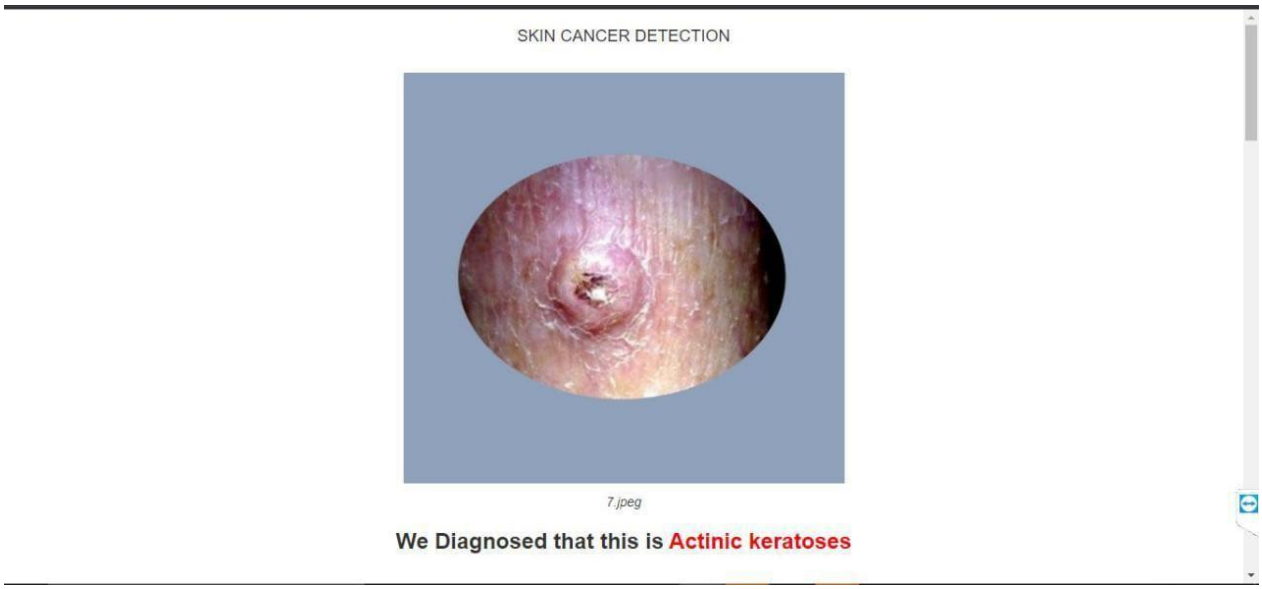


Fig 8.2 detection of actinic keratoses

This is the picture where we can find the reference of doctors

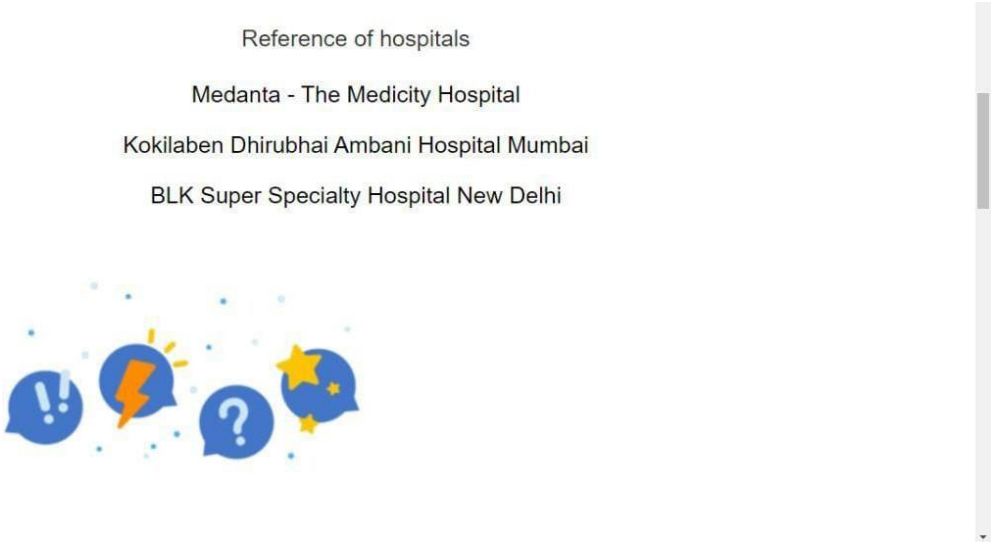


Fig 8.3 reference of hospitals

These are the pictures where you can find the feedback form

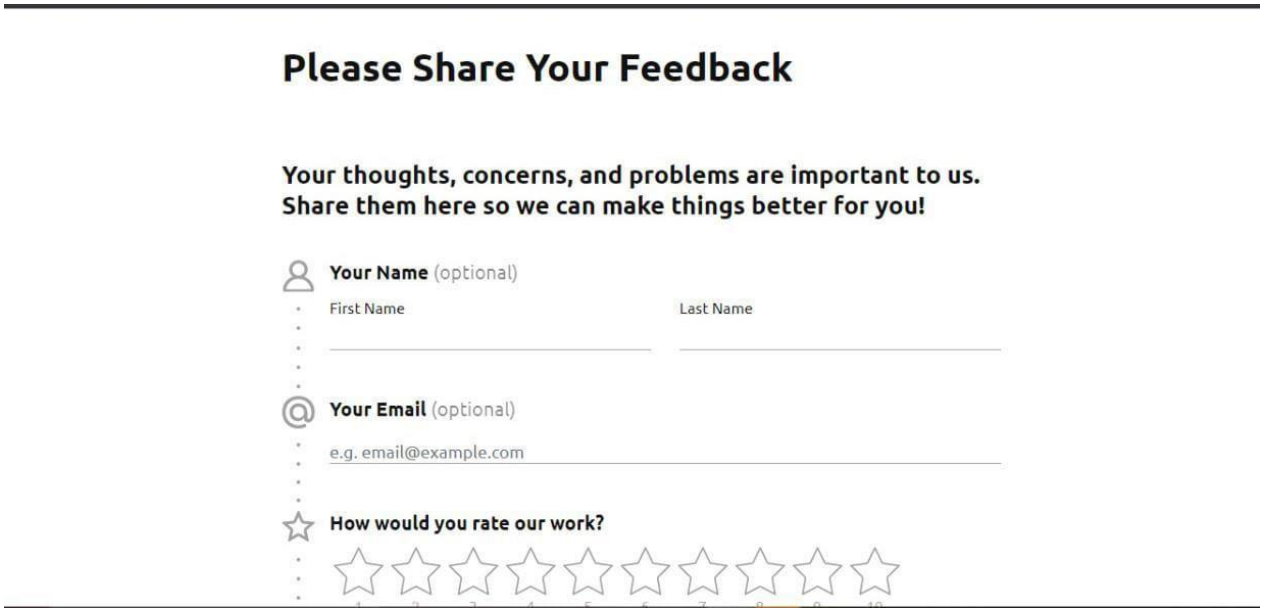


Fig 8.4 feedback form

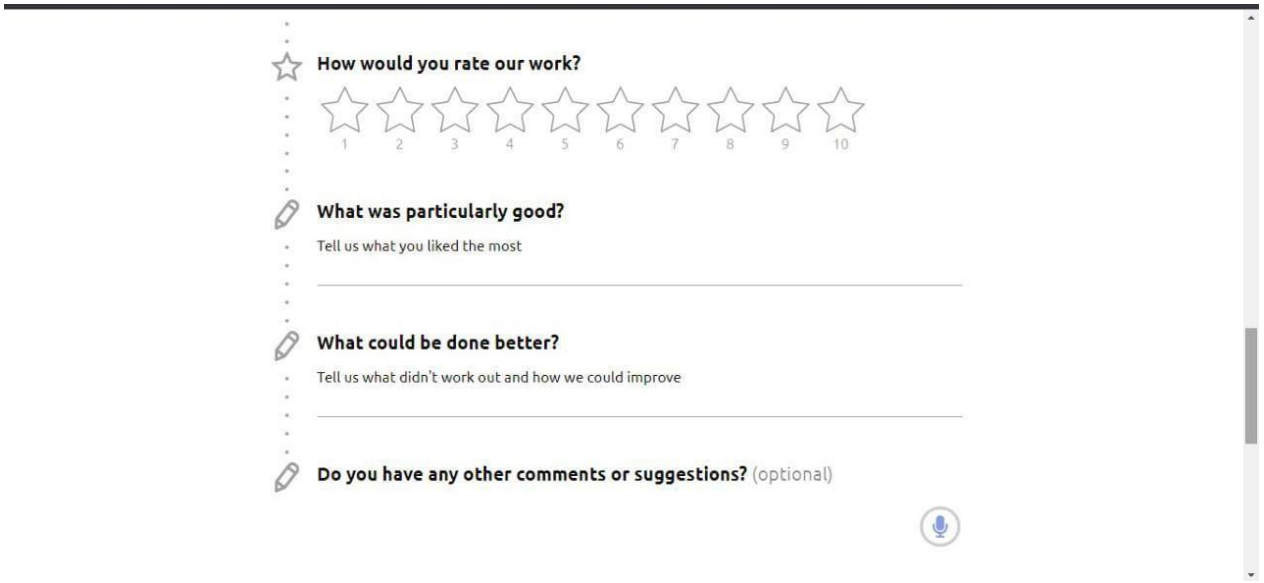


Fig:8.5 feedback form

CHAPTER-9

CONCLUSION

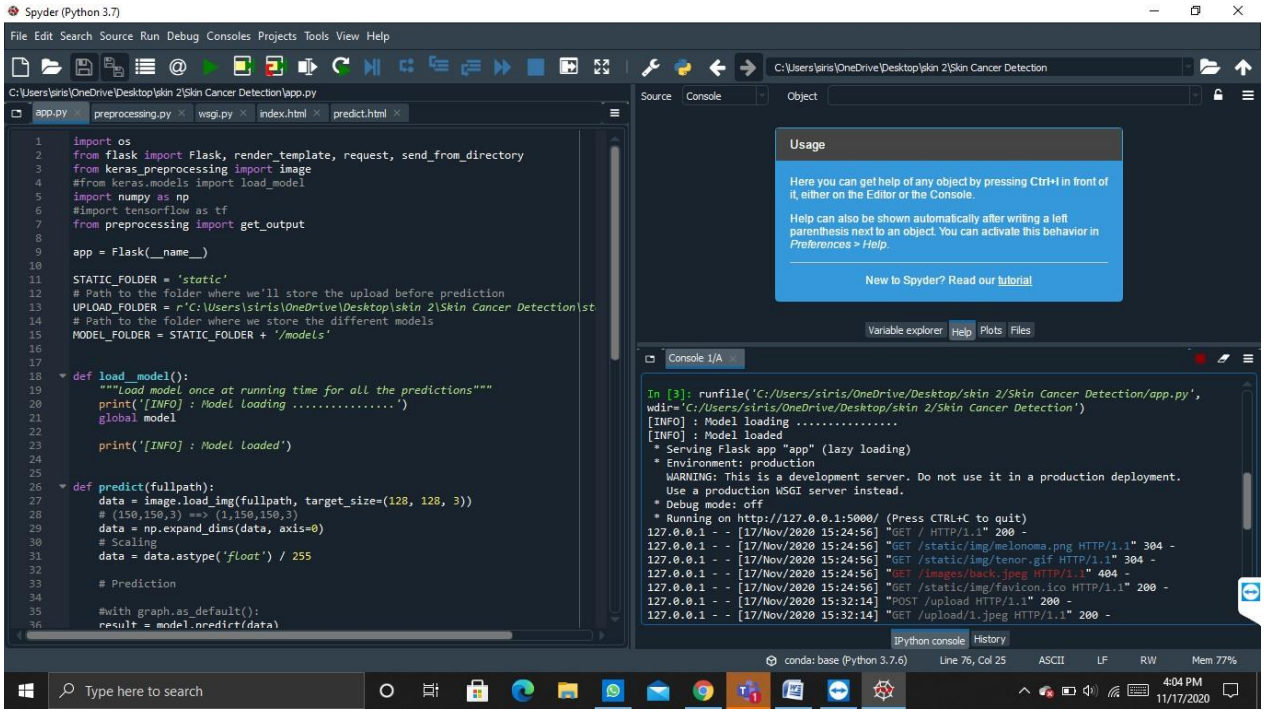
In the proposed system, Image Pre-Processing, Image Segmentation and Image Classification steps are performed for categorizing skin lesion images into melanoma or benign. Data augmentation technique is used in Convolutional Neural Network for increasing the number of images which leads to better performance of proposed method. Experimental results show an accuracy of CNN algorithm developed with data augmentation is higher than the CNN algorithm created without data augmentation. The proposed method detects melanoma faster than the biopsy method. The proposed method can be extended to identify different types of skin related diseases.In this project we also designed for the reference of doctors and a feedback form which is used to know the experience of the patients.

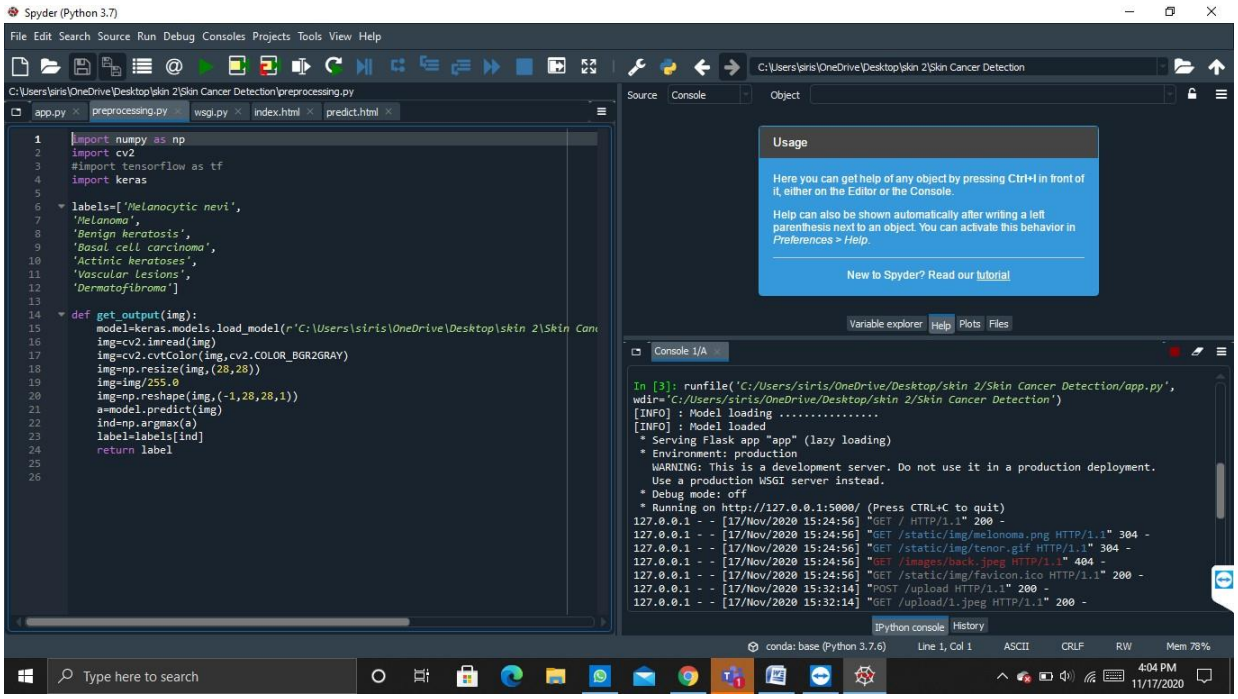
REFERENCES

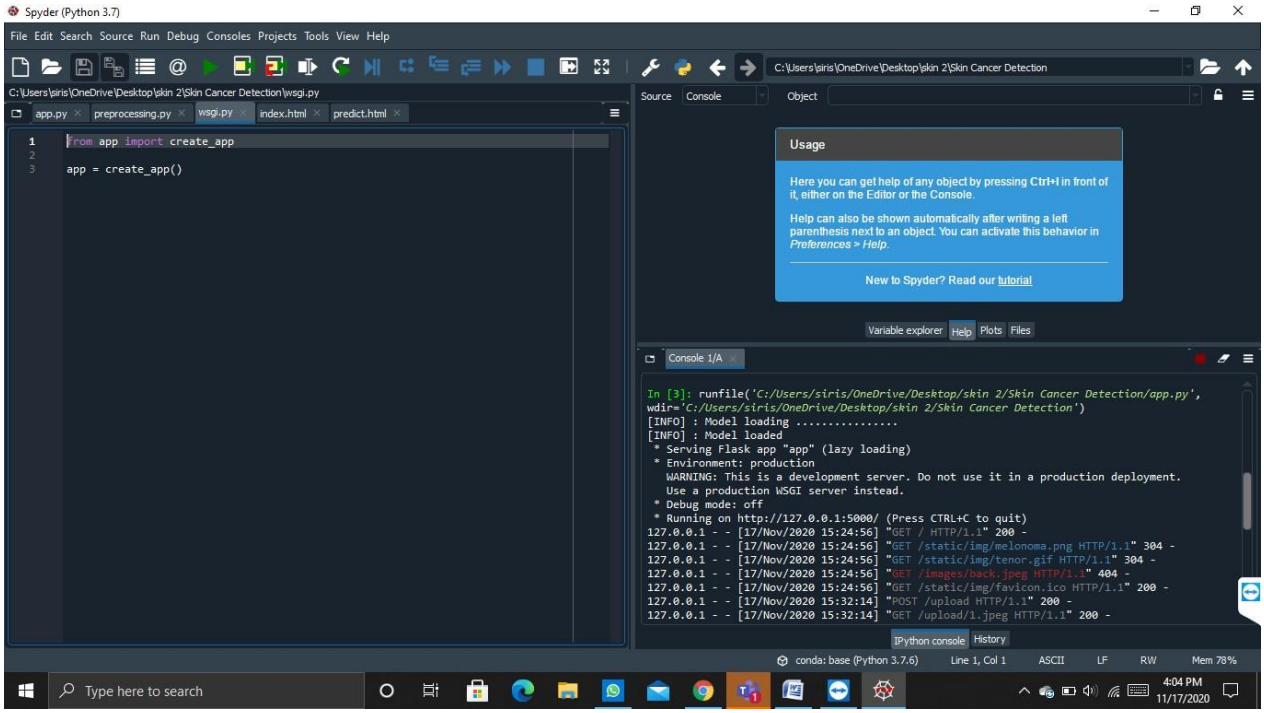
1. https://uwaterloo.ca/vision-image-processing-lab/research-demos/skin-cancer-detection
2. Skin cancer statistics, https://www.wcrf.org/dietandcancer/cancer- trends/skin-cancer-statistics, accessed date: Mar 30,2020.
3. Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. 2012. ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems (2012).
4. Xino Yao. 1999. Evolving artificial neural networks. Proc. IEEE 87, 9 (1999), 1423—1447
5. mirreza Mahbod, Gerald Schaefer, Chunliang Wang, Rupert Ecker, Isabella Ellinger, "Skin Lesion Classification Using Hybrid Deep Neural Networks", IEEE, International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1229-1233,2019.
6. Balazs Harangi, Agnes Baran , Andras Hajdu, "Classification of Skin Lesions Using An Ensemble of Deep Neural Networks", IEEE, 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.2575-2578, 2018.
7. Signs and Symptoms of Skin Cancer, https://www.cancer.org/cancer/melanoma-skin-cancer/detection- diagnosis-staging/signs-and-symptoms.html, accessed date: Mar 30,2020.
8. Tests for Melanoma Skin Cancer, https://www.cancer.org/cancer/melanoma- skin-cancer/detection- diagnosis-staging/how-diagnosed.html, accessed date: Mar 30, 2020.

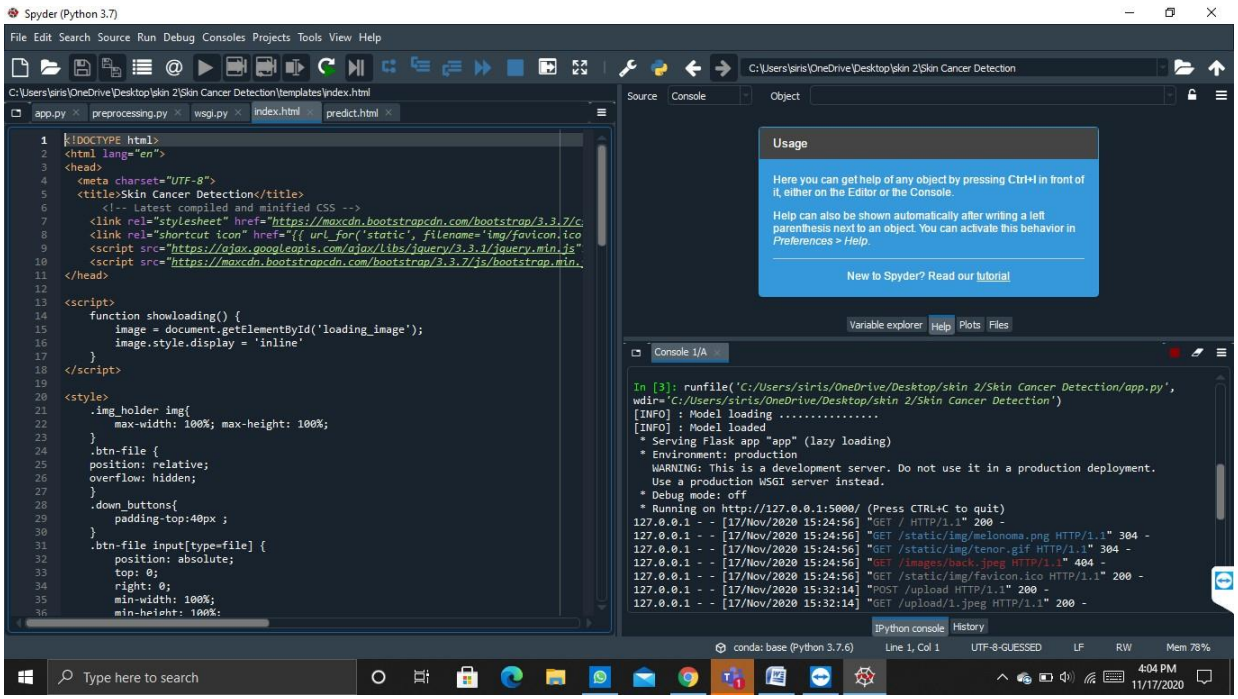
APPENDIX A

SAMPLE SCREEN


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\app.py
app.py preprocessing.py wsgi.py index.html predict.html
import os
from flask import Flask, render_template, request, send_from_directory
from keras_preprocessing import image
#from keras.models import load_model
import numpy as np
#import tensorflow as tf
from preprocessing import get_output
app = Flask(__name__)
STATIC_FOLDER = 'static'
# Path to the folder where we'll store the upload before prediction
UPLOAD_FOLDER = r'C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\st
# Path to the folder where we store the different models
MODEL_FOLDER = STATIC_FOLDER + '/models'
def load__model():
"""Load model once at running time for all the predictions"""
print('[INFO] : Model loading .....')
global model
print('[INFO] : Model loaded')
def predict(fullpath):
data = image.load_img(fullpath, target_size=(128, 128, 3))
# (150,150,3) ==> (1,150,150,3)
data = np.expand_dims(data, axis=0)
# Scaling
data = data.astype('float') / 255
# Prediction
#with graph.as_default():
result = model.predict(data)
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files
Console 1/A
In [3]: runfile('C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection/app.py', wdir='C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection')
[INFO] : Model loading .....
[INFO] : Model loaded
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/melonoma.png HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/tenor.gif HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /images/back.jpeg HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "GET /upload/1.jpeg HTTP/1.1" 200 -
IPython console History
conda: base (Python 3.7.6) Line 76, Col 25 ASCII LF RW Mem 77%
Type here to search
4:04 PM 11/17/2020


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\preprocessing.py
app.py preprocessing.py wsgi.py index.html predict.html
import numpy as np
import cv2
#import tensorflow as tf
import keras
labels=['Melanocytic nevi',
'Melanoma',
'Benign keratosis',
'Basal cell carcinoma',
'Actinic keratoses',
'Vascular lesions',
'Dermatofibroma']
def get_output(img):
model=keras.models.load_model(r'C:\Users\siris\OneDrive\Desktop\skin 2\Skin Can
img=cv2.imread(img)
img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
img=np.resize(img,(28,28))
img=img/255.0
img=np.reshape(img,(-1,28,28,1))
a=model.predict(img)
ind=np.argmax(a)
label=labels[ind]
return label
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files
Console 1/A
In [3]: runfile('C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection/app.py', wdir='C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection')
[INFO] : Model loading .....
[INFO] : Model loaded
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/melonoma.png HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/tenor.gif HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /images/back.jpeg HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "GET /upload/1.jpeg HTTP/1.1" 200 -
IPython console History
conda: base (Python 3.7.6) Line 1, Col 1 ASCII CRLF RW Mem 78%
Type here to search
4:04 PM 11/17/2020


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\wsgi.py
app.py preprocessing.py wsgi.py index.html predict.html
from app import create_app
app = create_app()
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files
Console 1/A
In [3]: runfile('C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection/app.py', wdir='C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection')
[INFO] : Model loading .....
[INFO] : Model loaded
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/melonoma.png HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/tenor.gif HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /images/back.jpeg HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "GET /upload/1.jpeg HTTP/1.1" 200 -
IPython console History
conda: base (Python 3.7.6) Line 1, Col 1 ASCII LF RW Mem 78%
Type here to search
4:04 PM 11/17/2020


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\templates\index.html
app.py preprocessing.py wsgi.py index.html predict.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Skin Cancer Detection</title>
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/c
<link rel="shortcut icon" href="{{ url_for('static', filename='img/favicon.ico
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.
</head>
<script>
function showloading() {
image = document.getElementById('loading_image');
image.style.display = 'inline'
}
</script>
<style>
.img_holder img{
max-width: 100%; max-height: 100%;
}
.btn-file {
position: relative;
overflow: hidden;
}
.down_buttons{
padding-top:40px ;
}
.btn-file input[type=file] {
position: absolute;
top: 0;
right: 0;
min-width: 100%;
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer Help Plots Files
Console 1/A
In [3]: runfile('C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection/app.py', wdir='C:/Users/siris/OneDrive/Desktop/skin 2/Skin Cancer Detection')
[INFO] : Model loading .....
[INFO] : Model loaded
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/melonoma.png HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/tenor.gif HTTP/1.1" 304 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /images/back.jpeg HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2020 15:24:56] "GET /static/img/favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2020 15:32:14] "GET /upload/1.jpeg HTTP/1.1" 200 -
IPython console History
conda: base (Python 3.7.6) Line 1, Col 1 UTF-8-GUESSED LF RW Mem 78%
Type here to search
4:04 PM 11/17/2020

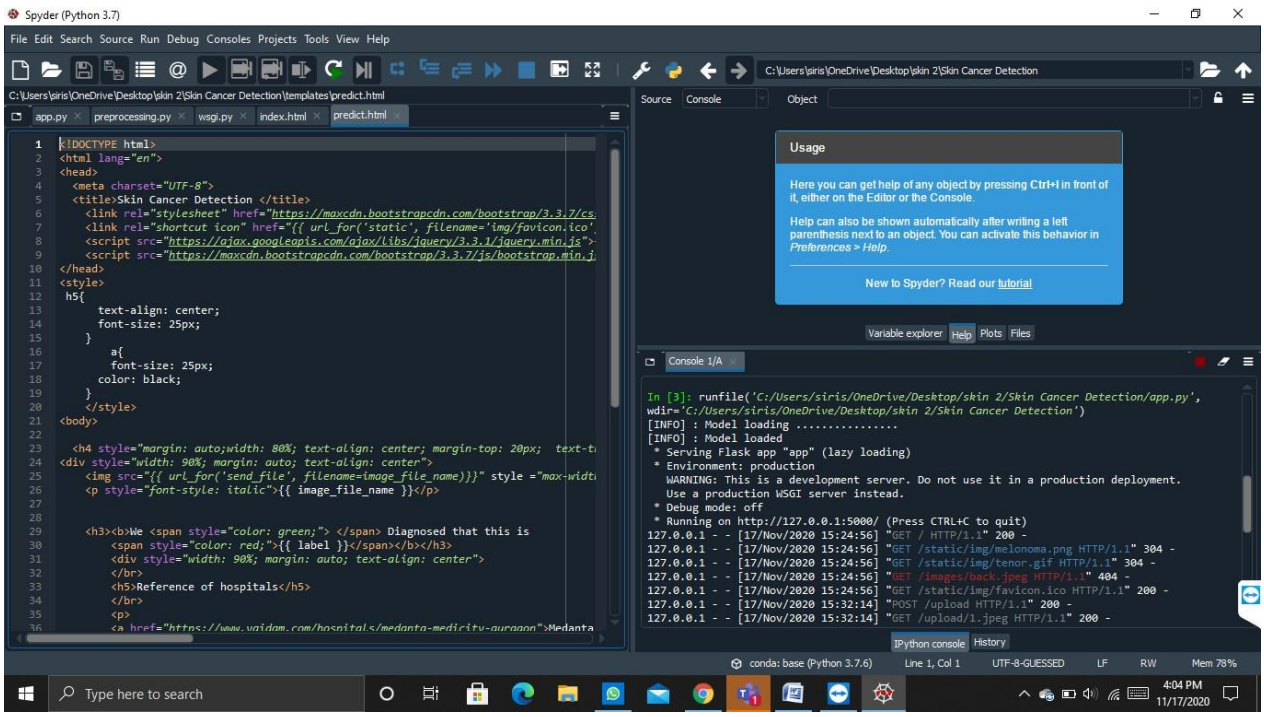


Fig :Sample Screens for various models

APPENDIX B

SAMPLE CODE

Code for detection:

```
import os


from flask import Flask, render_template, request, send_from_directory
from keras_preprocessing import image
#from keras.models import load_model
import num py as np
#import tensor flow as tf
from preproc essing import get_output
app = Flask(_name_)
```

```
STATIC_FOLDER = 'static'

# Path to the folder where we'll store the upload before prediction

UPLOAD_FOLDER = r'C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\static\uploads'

# Path to the folder where we store the different models

MODEL_FOLDER = STATIC_FOLDER + '/models'

def load__model():

    """Load model once at running time for all the predictions"""

    print('[INFO] : Model loading ..... ')

    global model

    print('[INFO] : Model loaded')

def predict(fullpath):

    data = image.load_img(fullpath, target_size=(128, 128, 3))

    # (150,150,3) ==> (1,150,150,3)

    data = np.expand_dims(data, axis=0)

    # Scaling

    data = data.astype('float') / 255

    # Prediction

    #with graph.as_default():

    result = model.predict(data)

    return result

# Home Page

@app.route('/')

def index():

    return render_template('index.html')

# Process file and predict his label

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

def upload_file():
```

```
    if request.method == 'GET':
        return render_template('index.html')
    else:
        file = request.files['image']
        fullname = os.path.join(UPLOAD_FOLDER, file.filename)
        file.save(fullname)
        label= get_output(fullname)
        return render_template('predict.html', image_file_name=file.filename, label=label)
@app.route('/upload/<filename>')
def send_file(filename):
    return send_from_directory(UPLOAD_FOLDER, filename)
def create_app():
    load_model()
    return app
if__name__== '_main_':
    app = create_app()
    app.run(debug=False)
```

Code for preprocessing:

```
import numpy as np
import cv2
#import tensorflow as tf
import keras
labels=['Melanocytic nevi',
'Melanoma',
'Benign keratosis',
'Basal cell carcinoma',
```

```
'Actinic keratoses',

'Vascular lesions',

'Dermatofibroma']

def get_output(img):

    model=keras.models.load_model(r'C:\Users\siris\OneDrive\Desktop\skin 2\Skin Cancer Detection\static\models\cancer2.h5')

    img=cv2.imread(img)

    img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    img=np.resize(img,(28,28))

    img=img/255.0

    img=np.reshape(img,(-1,28,28,1))

    a=model.predict(img)

    ind=np.argmax(a)

    label=labels[ind]

    return label
```

Code for app:

```
from app import create_app

app = create_app()
```

Code for first webpage:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Skin Cancer Detection</title>

     <!-- Latest compiled and minified CSS -->

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
    <link rel="shortcut icon" href="{{ url_for('static', filename='img/favicon.ico') }}" type="image/x-icon">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<script>
    function showloading() {
        image = document.getElementById('loading_image');
        image.style.display = 'inline'
    }
</script>
<style>
    .img_holder img{
        max-width: 100%; max-height: 100%;
    }
    .btn-file {
    position: relative;
    overflow: hidden;
    }
    .down_buttons{
        padding-top:40px ;
    }
    .btn-file input[type=file] {
        position: absolute;
        top: 0;
        right: 0;
        min-width: 100%;
        min-height: 100%;
```

```
    font-size: 100px;
    text-align: right;
    filter: alpha(opacity=0);
    opacity: 0;
    outline: none;
    background: white;
    cursor: inherit;
    display: block;
}
.main{
    background-color: blanchedalmond) ;
    background-repeat: no-repeat;
    background-size: 100% 100%;
}
h1
{
  text-align: center;
  margin: 0;
  font-size: 25px;
  color:#14274E;
}
body{
  background-color: #ebe5e5;
  background-image:url("images/back.jpeg");
  background-size: 1500px 1500px;
  background-repeat: no-repeat;
}
```

```
p{
  text-align: center;
  font-size: 20px;
}
hr{
  border-style: none;
  border-top-style: dotted;
  border-color: #A6A6A4;
  border-width:5px;
  width: 100%;
}
h5{
  text-align: center;
  font-size: 25px;
}
a{
  color: black;
}
h2{
  text-align:center;
}
.contact-us{
  text-align: center;
}
.middle-container{
 text-align: center;
}
```

```
</style>

<body class='main'>

  <body>

    <h1>SKIN CANCER DETECTION</h1>

    <br />

   <div class="middle-container">

   </div>

   <br />

    <p>

      SKIN CANCER  develop anywhere on  body. They most  develop  areas that have  exposure in a  sun, such as your back, legs, arms and face.

      <br />

      Skin cancer also occur areas in that don't receive sun to exposure, such as a soles of your feet, palms  hands and fingernail beds.

      <br />

      These hidden skin cancer are more common in people with darker skin.

      <br />

      Make sure u r save from melonama by trying our detection center

      </p>

      <br />

      <hr />

    </body>

<div class="container" style="margin: auto;width:40%; text-align: center; margin-top: 70px;  text-transform: uppercase">

 <h3 style="margin: auto;width: 80%; text-align: center; margin-top: 40px;  text-transform: uppercase">Skin Cancer Detection</h3>

  <div style="text-align: center; margin-top: 10px" >

    <img style="max-width:  300px"  src="{{  url_for('static',  filename='img/melonoma.png') }}"/>
```

```
  </div>

  <form action="/upload" method="post" class="down_buttons" enctype="multipart/form-data" style="margin-top: 50px; width: 60%; text-align: center; margin: auto;" onsubmit="showloading()">

     <span class="text-left btn btn-default btn-file">

       Upload Image <input type="file" name="image">

    </span>

     <span class="text-right">

       <input type="submit" value="Predict" class="btn btn-primary">

     </span>

  </form>

    <div style="text-align: center">

       <img id="loading_image" style="display: none" src="{{ url_for('static', filename='img/tenor.gif')}}">

    </div>

          <br />

          <hr />

    <div class="contact-us">

       <h2>Please ask us if you have any queries</h2>

       <h2>Don't fight SKIN CANCER alone.</h2>

       </br>

       <a class="btn" href="mailto:chowdarya997@gmail.com">CONTACT US</a>

     </div>

     </br>

    <div class="bottom-container">

     <p class="copyright">© 2021 Avinash ,Balaji <br /><br />@Sathyabama institute of  science and technology.</p>

    </div>

</div>
```

```
</body>

</html>
```

Code for second webpage:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Skin Cancer Detection </title>

   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

   <link rel="shortcut icon" href="{{ url_for('static', filename='img/favicon.ico') }}" type="image/x-icon">

   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</head>

<style>

 h5{

     text-align: center;

     font-size: 25px;

   }

       a{

       font-size: 25px;

     color: black;

   }

   </style>

<body>
```

```
 <h4 style="margin: auto;width: 80%; text-align: center; margin-top: 20px;  text-transform: uppercase">Skin Cancer Detection </h4>

<div style="width: 90%; margin: auto; text-align: center">

  <img src="{{ url_for('send_file', filename=image_file_name)}}" style ="max-width: 500px; max-height: 450px; margin-top: 30px; margin-bottom: 10px">

  <p style="font-style: italic">{{ image_file_name }}</p>

  <h3><b>We <span style="color: green;"> </span> Diagnosed that this is

    <span style="color: red;">{{ label }}</span></b></h3>

    <div style="width: 90%; margin: auto; text-align: center">

    </br>

    <h5>Reference of hospitals</h5>

    </br>

    <p>

    <a href="https://www.vaidam.com/hospitals/medanta-medicity-gurgaon">Medanta - The Medicity Hospital</a>

    <br/>

    </br>

    <a href="https://www.vaidam.com/hospitals/kokilaben-dhirubhai-ambani-hospital-mumbai">Kokilaben Dhirubhai Ambani Hospital Mumbai</a>

    <br/>

    </br>

    <a href="https://www.vaidam.com/hospitals/blk-hospital-new-delhi">BLK Super Specialty Hospital New Delhi</a>

    <br/>

    </p>

    <br/>

  <div data-aidaform-widget="form-2019-12" data-url="https://sirishma.aidaform.com/free-feedback-form" data-width="100%" data-height="500px" data-do-resize></div>

  <script>(function(){var r,d=document,gt=d.getElementById,cr=d.createElement,tg=d.getElementsByTagName,id="aidaform-
```

```
embed";if(!gt.call(d,id)){r=cr.call(d,"script");r.id=id;r.src="https://embed.aidaform.com/embed.j
s";(d.head || tg.call(d,"head")[0]).appendChild(r);}})()</script>

    <br/>

     <div class="contact-us">

       <h2>Please ask us if you have any queries</h2>

       <h2>Don't fight Melonoma alone.</h2>

       <a class="btn" href="mailto:chowdarya997@gmail.com">CONTACT US</a>

     </div>

   <div class="bottom-container">

     <p class="copyright">©2021 Avinash ,Balaji <br /><br />@Sathyabama institute of science
and technology.</p>

   </div>

</div>

</br>

    <a href="/">Back to Home</a>

</div>

</body>

</html>
```