DEEP LEARNING APPROACH TO MELANOMA SKIN CANCER DETECTION

Mrs.BABITHA S, M.E., Assistant Professor,
Department of Computer Science and Engineering
Mr.PRAVIN S, B.E, Student of Computer Science Engineering
St. Joseph College of Engineering, Sriperumbudur, Chennai.

ABSTRACT

The largest organ in the human body is the skin, and cancer of the skin is one of the most serious types of cancer. Genetic diseases can result in aberrant cell development in the human body due to a variety of pathogenic abnormalities. These alterations in human skin cells are extremely risky. Due to the high mortality rate of skin cancer and the slow spread of disease to other regions of the body, early detection is crucial. It is exceedingly difficult to diagnose skin cancer through eye inspection and manual examination of skin lesions.

Numerous early detection methods for skin cancer have been suggested in light of these worries. A range of deep learning, machine learning, and computer vision approaches were fused for the rapid advancement of computer-aided diagnosis systems. Our research allows us to an extensive literature review of the methods, technologies, and approaches applied for the examination of various types of skin cancer to date. The survey includes preprocessing of image, segmentation, feature extraction of the image, selection, and classification approaches for skin cancer identification.

The results of these research are good but still, some challenges occurs in the analysis of skin cancer types because of the complex and rare features. Hence, the prior objective is to examine the existing methodologies utilized in the discovery of skin lesions by finding the obstacles that helps the researchers to contribute in future research.

KEYWORDS: deep learning, machine learning, melanoma, skin lesions, Convulational neural network(CNN)

1. INTRODUCTION

Automated skin lesion classification in dermoscopy images is an essential way to improve the diagnostic performance and reduce melanoma deaths. Automatic localization of skin lesions within dermoscopy images is a crucial step toward developing a decision support system for skin cancer detection. However, segmentation of the lesion image can be challenging, as these images possess various artifacts distorting the uniformity of the lesion area. Recently, deep convolution learning-based techniques have drawn great attention for pixel-wise image segmentation.

These deep networks produce coarse segmentation, and convolutional filters and pooling layers result in segmentation of a skin lesion at a lower resolution than the original skin image. To overcome these drawbacks, the proposed system uses a super-pixel based fine-tuning strategy to effectively utilize the characteristics of the skin image pixels to accurately extract the border of the lesion. The proposed approach not only learns a global map for skin lesions, but also acquires the local contextual information, such as lesion boundary. It can, therefore, accurately segment lesions within a given skin image, even in the presence of fuzzy boundaries and complex textures.

.

1. LITERATURE SURVEY

[1] **Title: Skin Cancer Detection using Deep Learning**

Authors : R. Senthil Kumar, Amarjeet Singh, Sparsha Srinath,Nimal Kurien Thomas,Vishal Arasu,

Publication: 2022 International Conference on Electronics and Renewable Systems (ICEARS)

Identifying melanoma at the early stages of diagnosis is imperative as early detection can exponentially increase oneu2019s chances of cure.The paper first proposes a literature survey of multiple methods used for performing skin cancer classification. Our methodology consists of using Convolutional Neural Network (CNN) to identify and diagnose the skin cancer using the IS IC dataset containing 2637 images. The proposed model gives an accuracy of 88% for classifying the training dataset as either benign or malignant.",

[2] **Title : Skin cancer classification using Convolutional neural networks**

Authors: R Raja Subramanian,Dintakurthi Achuth,P Shiridi Kumar,Kovvuru Naveen kumar Reddy,Srikar Amara,Adusumalli Suchan Chowdary,

Publication: 2021 11th International Conference on Cloud Computing Data Science & Engineering (Confluence)

There is a necessary need for early detection of skin cancer and can prevent further spread in some cases of skin cancers, such as melanoma and focal cell carcinoma. Anyhow there are several factors that have bad impacts on the detection accuracy.

In Recent times, the use of image processing and machine vision in the field of healthcare and medical applications is increasing at a greater phase. In this paper, we are using the Convolution neural networks to detect and classify the class of cancer based on historical data of clinical images using CNN.Some of our objectives through this research are ,to build a CNN model to detect skin cancer with an accuracy of >80% ,to keep the false negativity rate in the prediction to below 10%, to reach the precision of above 80% and do visualization on our Data. Simulation results show that the proposed method has superiority towards the other compared methods.",

[3] Title: The melanoma skin cancer detection and classification using support vector machine

Authors: Hiam Alquran,Isam Abu Qasmieh,Ali Mohammad Alqudah,Sajidah Alhammouri,Esraa Alawneh,Ammar Abughazaleh,Firas Hasayen,

Publication 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)

Melanoma skin cancer detection at an early stage is crucial for an efficient treatment. Recently, it is well known that, the most dangerous form of skin cancer among the other types of skin cancer is melanoma because it's much more likely to spread to other parts of the body if not diagnosed and treated early. The non-invasive medical computer vision or medical image processing plays increasingly significant role in clinical diagnosis of different diseases. Such techniques provide an automatic image analysis tool for an accurate and fast evaluation of the lesion.

The steps involved in this study are collecting dermoscopy image database, preprocessing, segmentation using thresholding, statistical feature extraction using Gray Level Co-occurrence Matrix (GLCM), Asymmetry, Border, Color, Diameter, (ABCD) etc., feature selection using Principal component analysis (PCA), calculating total Dermoscopy Score and then classification using Support Vector Machine (SVM). The results show that the achieved classification accuracy is 92.1%.",

[4] Title : Analysis and Classification of Skin Cancer Based on Deep Learning Approach
Authors: Youssef Filali,Hasnae El Khoukhi,My Abdelouahed Sabri,Abdellah Aarab,
Publication: 2022 International Conference on Intelligent Systems and Computer Vision (ISCV)

Skin cancer has become more dangerous in recent years due to its rapid and widespread spread around the world. This evidence has sparked peopleu2019s interest and efforts to develop automatic diagnostic computer systems that can help in early diagnosis. Several approaches based on machine learning and deep learning have been developed to assist in the diagnosis of skin lesions.

Our objective in this paper is to conduct a comparative study between different deep learning approaches for skin cancer classification and analysis. Four deep learning-based architectures were studied; ResNet, VGG16, GoogleNet and AlexNet to classify skin cancer into melanoma or non-melanoma. As a finding from our comparative study, the ResNet architecture provided better classification accuracy with a very promising result especially for the False-positives error rate.",

[5] Title : An efficient technique for skin cancer classification using deep learning
Authors: Rehan Ashraf,Iqra Kiran,Toqeer Mahmood,Ateeq Ur Rehman Butt,Nafeesa Razzaq,Zobia Farooq,
Publication: 2020 IEEE 23rd International Multitopic Conference (INMIC)

The development and spreading of abnormal cells in the skin of the human body assumed to be as skin cancer. There are different skin cancer types,. however, melanoma is the most critical one. The prediction of cancer at an early stage could help in better and improved treatment. In this research, a deep convolutional neural network-based technique to extract spatial information is developed as a method for skin cancer classification. For experimentation, the dataset includes real data that is collected from DHQ hospital Faisalabad, Pakistan.

The classification results of the our proposed method are compared with state-of-the-art approaches while utilizing a reduced number of factors/feature vectors. The experiments exhibit that the classification accuracy is about 93.29% using our proposed method. The outcome of the experimental investigation shows that our method has higher accuracy in comparison with state-of-the-art techniques in literature

3. PROPOSED SYSTEM

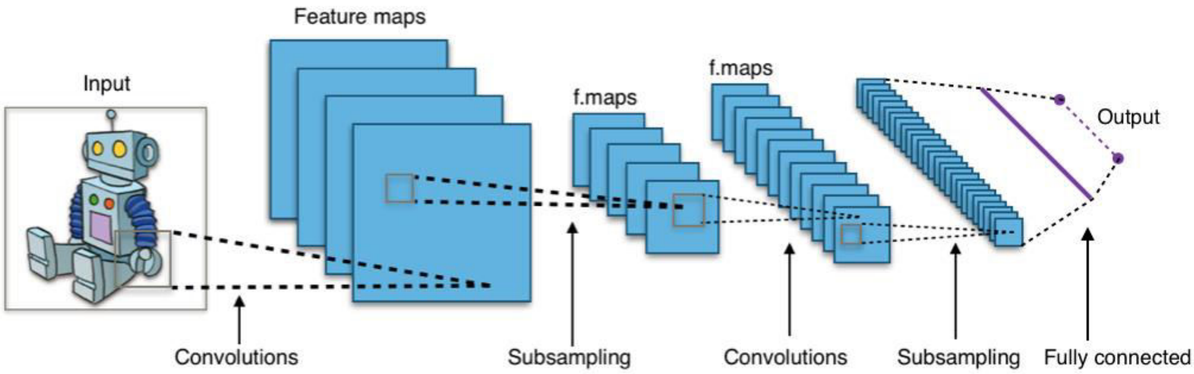
The CNN algorithm has been employed to carry out the prediction. The proposed approach detects the various types of skin cancer Disease such as melanoma,nevus,vascular lesion,etc. using CNN algorithm. It reduces the time required to predict the output and can be used for real time predictions. Feature extraction using principal component analysis and feature selection techniques were also employed. Data pre-processing refers to all transformations on the raw data before it is fed to the machine learning or deep learning algorithm. Real- world data is often incomplete, inconsistent, and lacking in certain behaviors or trends, and is likely to contain many errors.

4.SYSTEM DESIGN FOR CONVULATIONAL NEURAL NETWORK

In deep learning, a convolutional neural network (CNN, or Convolution Neural Network) is a class of deep neural networks, most commonly applied to analysing visual imagery. They have applications in image and video recognition, recommender systems, image classification, Image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

Fig.4.1. CNN architecture

A



convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically this includes a layer that does multiplication or other dot product, and its activation function is commonly ReLU. This is followed by other convolution layers such as pooling layers, fully connected layers and normalization layers.

IMPLEMENTATION

Importing Library Files:

```
from flask import Flask, render_template, url_for, redirect, request
from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user, current_user
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from werkzeug.utils import secure_filename
import subprocess
import uuid
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import load_model
from tkinter import *
import tkinter.messagebox
import PIL.Image
import PIL.ImageTk
from tkinter import filedialog
import h5py
import json
import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, array_to_img,
load_img
from keras.models import Sequential, load_model, Model
app = Flask(_name_)
db = SQLAlchemy(app)
```

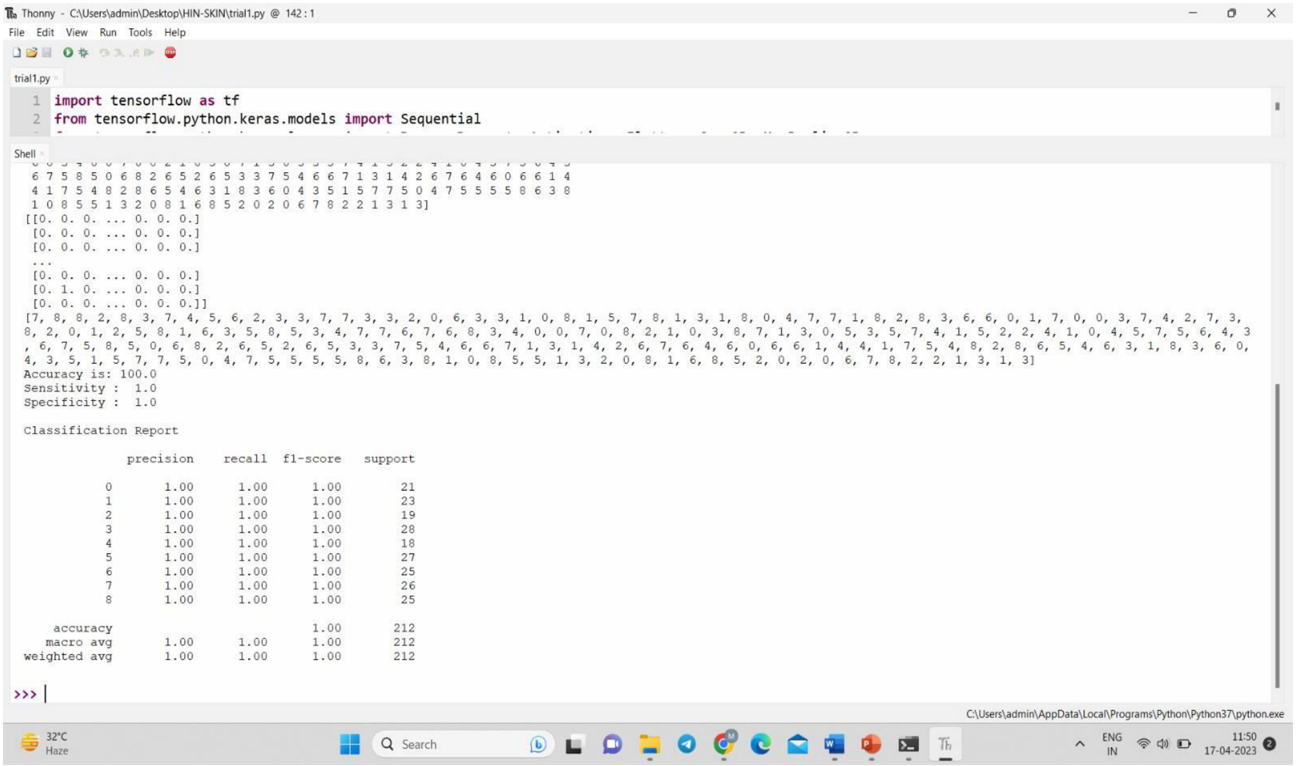
```
bcrypt = Bcrypt(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['SECRET_KEY'] = '12345678'

app.config['UPLOAD_IMAGE'] = r"C:/Users/admin/Desktop/HIN-SKIN/static/Images"
app.config['UPLOAD_PATH'] =  r"C:/Users/admin/Desktop/HIN-SKIN/static/"

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

import numpy as np
import matplotlib.pyplot as plt
import cv2



import tensorflow as tf
from tensorflow.keras.models import load_model
from tkinter import *
import tkinter.messagebox
import PIL.Image
import PIL.ImageTk
from tkinter import filedialog
import h5py
import json
import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, array_to_img,
load_img
from keras.models import Sequential, load_model, Model

CATEGORIES = ["actinic keratosis","basal cell
carcinoma","dermatofibroma","melanoma","nevus","pigmented benign keratosis","seborrheic
keratosis","squamous cell carcinoma","vascular lesion"]

model = tf.keras.models.load_model("CNN.model")

def detect_ridges(gray, sigma=1.0):
    H_elems = hessian_matrix(gray, sigma=sigma, order='rc')
    maxima_ridges, minima_ridges = hessian_matrix_eigvals(H_elems)
    return maxima_ridges, minima_ridges

def prepare(file):
    IMG_SIZE = 150
    img_array = cv2.imread(file, cv2.IMREAD_GRAYSCALE)
    clahe = cv2.createCLAHE(clipLimit=100.0, tileGridSize=(8,8))
```

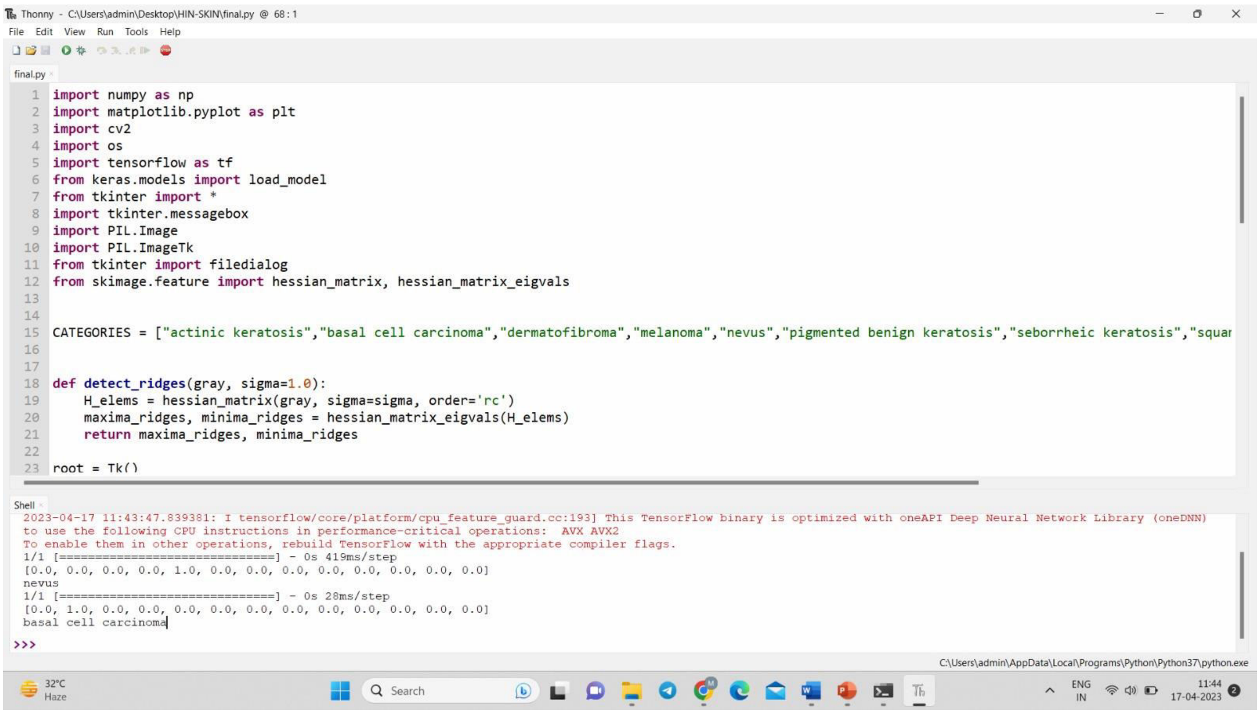
```
    img_array = clahe.apply(img_array)
    median  = cv2.medianBlur(img_array.astype('uint8'), 5)
    median = 255-median
    ret,thresh = cv2.threshold(median.astype('uint8'),165,255,cv2.THRESH_BINARY_INV)
    new_array = cv2.resize(thresh, (IMG_SIZE, IMG_SIZE))
    return new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 1)
def detect(filename):
    print("the file")
    print(filename)
    prediction = model.predict(prepare(filename))
    prediction = list(prediction[0])
    print(prediction)
    l=CATEGORIES[prediction.index(max(prediction))]
    print(CATEGORIES[prediction.index(max(prediction))])
    image_path=filename
    rty="The Result is " + str(CATEGORIES[prediction.index(max(prediction))])
    return rty

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))


class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)


class RegisterForm(FlaskForm):
    username = StringField(validators=[
                       InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
                         InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})

    submit = SubmitField('Register')

    def validate_username(self, username):
        existing_user_username = User.query.filter_by(
            username=username.data).first()
        if existing_user_username:
            raise ValidationError(
                'That username already exists. Please choose a different one.')
class LoginForm(FlaskForm):
```

```
    username = StringField(validators=[
                           InputRequired(), Length(min=4, max=20)], render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
                             InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})

    submit = SubmitField('Login')


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user)
                return render_template('dashboard.html')
    return render_template('login.html', form=form)
@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():
      print("hello")
      #subprocess.call('python final.py', shell=True)
      return "done"

@app.route('/predict',methods=['GET', 'POST'])
@login_required
def upload_image():

    if request.method == 'POST':
        image = request.files['file']
        if image.filename == '':
            print("File name is Invalid")
            return redirect(request.url)
        image = request.files['file']
        file = os.path.join(app.config["UPLOAD_PATH"],"uploadedImages")
        renamed_file = str(uuid.uuid1()) + ".jpg"
        image.save(os.path.join(file,renamed_file))
        file_path=os.path.join(app.config["UPLOAD_PATH"],"uploadedImages\\"+renamed_file)
```

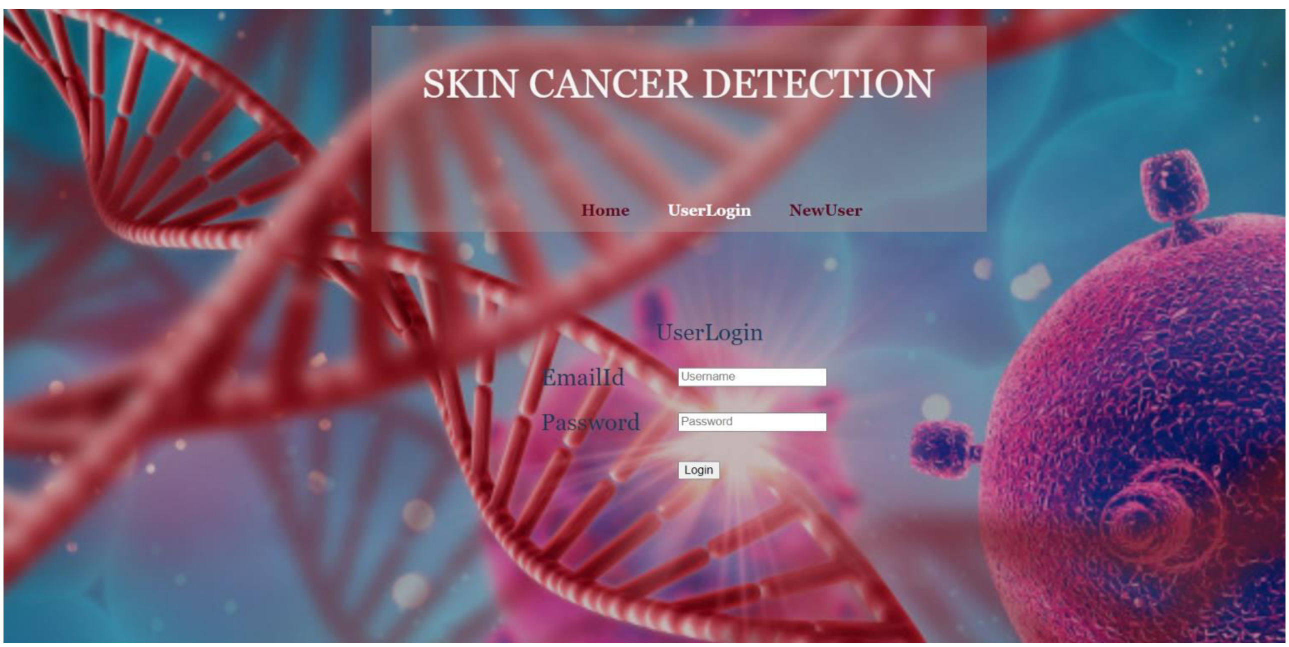
```
        print(file_path)
        amount = detect(file_path)
        print (amount)
        return render_template('dashboard.html',filename=renamed_file,amount=amount)
    return render_template('dashboard.html')

@app.route('/display')
def image_display(filename):
    return redirect(url_for('static',filename='/Images/'+filename),code=301)

@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('login'))


@ app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()

    if form.validate_on_submit():
        hashed_password = bcrypt.generate_password_hash(form.password.data)
        new_user = User(username=form.username.data, password=hashed_password)
        db.session.add(new_user)
        db.session.commit()
        return redirect(url_for('login'))

    return render_template('register.html', form=form)


if _name_ == "_main_":
    app.run(debug=True)
```

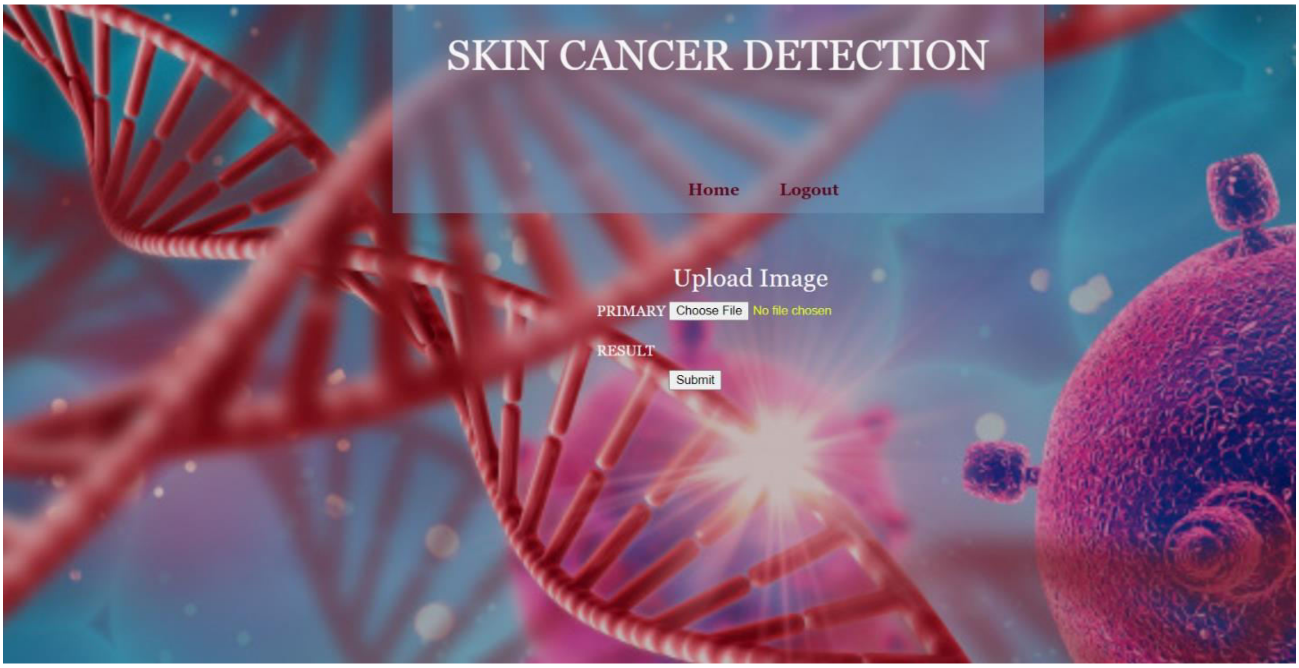
5.RESULT AND DISCUSSION

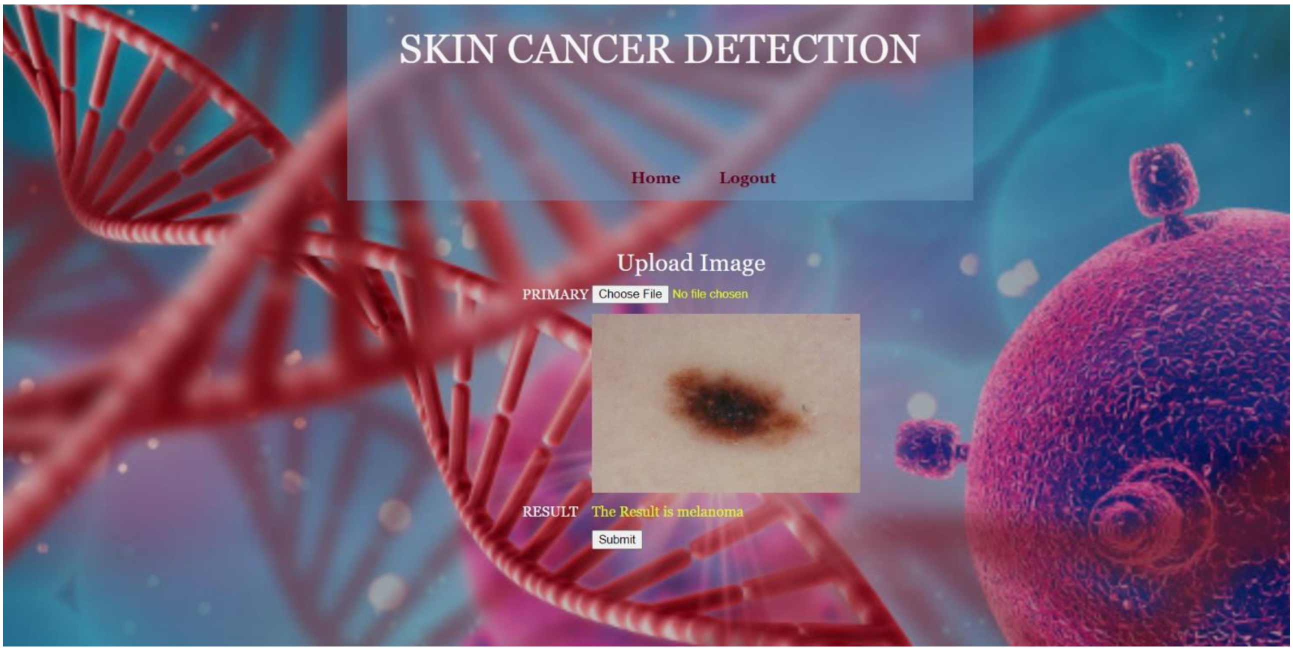

Thonny - C:\Users\admin\Desktop\HIN-SKIN\trial1.py @ 142 : 1
File Edit View Run Tools Help
trial1.py
import tensorflow as tf
from tensorflow.python.keras.models import Sequential
Shell
Accuracy is: 100.0
Sensitivity : 1.0
Specificity : 1.0
Classification Report
precision recall f1-score support
0 1.00 1.00 1.00 21
1 1.00 1.00 1.00 23
2 1.00 1.00 1.00 19
3 1.00 1.00 1.00 28
4 1.00 1.00 1.00 18
5 1.00 1.00 1.00 27
6 1.00 1.00 1.00 25
7 1.00 1.00 1.00 26
8 1.00 1.00 1.00 25
accuracy 1.00 212
macro avg 1.00 1.00 1.00 212
weighted avg 1.00 1.00 1.00 212
C:\Users\admin\AppData\Local\Programs\Python\Python37\python.exe
32°C
Haze
Search
ENG
IN
11:50
17-04-2023


Thonny - C:\Users\admin\Desktop\HIN-SKIN\final.py @ 68 : 1
File Edit View Run Tools Help
final.py
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
import tensorflow as tf
from keras.models import load_model
from tkinter import *
import tkinter.messagebox
import PIL.Image
import PIL.ImageTk
from tkinter import filedialog
from skimage.feature import hessian_matrix, hessian_matrix_eigvals
CATEGORIES = ["actinic keratosis","basal cell carcinoma","dermatofibroma","melanoma","nevus","pigmented benign keratosis","seborrheic keratosis","squar
def detect_ridges(gray, sigma=1.0):
H_elems = hessian_matrix(gray, sigma=sigma, order='rc')
maxima_ridges, minima_ridges = hessian_matrix_eigvals(H_elems)
return maxima_ridges, minima_ridges
root = Tk()
Shell
2023-04-17 11:43:47.839381: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 [==============================] - 0s 419ms/step
[0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
nevus
1/1 [==============================] - 0s 28ms/step
[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
basal cell carcinoma
C:\Users\admin\AppData\Local\Programs\Python\Python37\python.exe
32°C
Haze
Search
ENG
IN
11:44
17-04-2023


SKIN CANCER DETECTION
Home
UserLogin
NewUser
OUR JOB IS TO PROTECT YOUR SKIN
SKIN IS THE MOST VITAL ORGAN WE HAVE TO CARE FOR


SKIN CANCER DETECTION
Home
UserLogin
NewUser
UserLogin
EmailId
Username
Password
Password
Login


SKIN CANCER DETECTION
Home
Logout
Upload Image
PRIMARY
Choose File
No file chosen
RESULT
Submit


SKIN CANCER DETECTION
Home
Logout
Upload Image
PRIMARY
Choose File
No file chosen
RESULT
The Result is melanoma
Submit

5.CONCLUSION

In this project, we have proposed a reliable and robust method for skin cancer detection in highly cluttered images using CNN. The cluttered images are obtained using dermoscopy images . The image sequences also provide the candidate cancer region proposals done by multilevel graph cut. We have introduce a verification step in which the proposed region is classified into ,Melanoma,Normal classes, Thus, determining whether the proposed region is truly affected by skin cancer or not. We applied CNN features to machine learning algorithm to achieve better performance.

6.REFERENCE

1.Ashraf R., Afzal S., Rehman A.U., Gul S., Baber J., Bakhtyar M., MehmoodI., Song O.Y., Maqsood M. Region-of-Interest Based Transfer Learning Assisted Framework for Skin Cancer Detection. *IEEEAccess.* 2020;**8**:147858–147871.doi: 10.1109/ACCESS.2020.3014701. [CrossRef] [Google Scholar]

2.Byrd A.L., Belkaid Y., Segre J.A. The Human Skin Microbiome. *Nat. Rev. Microbiol.* 2018;**16**:143–155. doi: 10.1038/nrmicro.2017.157. [PubMed][CrossRef] [Google Scholar]

3.Elgamal M. Automatic Skin Cancer Images Classification. *IJACSA.* 2013;**4** doi: 10.14569/IJACSA.2013.040342. [CrossRef] [Google Scholar]

4.Key Statistics for Melanoma Skin Cancer. [(accessed on 8 February 2021)];*Am. Cancer Soc.* Available online: https://www.cancer.org/content/dam/CRC/PDF/Public/8823.00.pdf

5.Khan M.Q., Hussain A., Rehman S.U., Khan U., Maqsood M., Mehmood K., Khan M.A. Classification of Melanoma and Nevus in Digital Images forDiagnosis of Skin Cancer. *IEEE Access.* 2019;**7**:90132–90144.doi: 10.1109/ACCESS.2019.2926837. [CrossRef] [Google Scholar]

6.Premier Surgical Staff What Is the Difference between Melanoma Andnon-Melanoma Skin Cancer? [(accessed on 6 February2021)];*PSS.* Available online: https://www.premiersurgical.com/01/whats-the-difference- between-melanoma-and-non-melanoma-skin-cancer/

7.Rashid H., Tanveer M.A., Aqeel Khan H. Skin Lesion Classification Using GAN Based Data Augmentation; Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); Berlin, Germany. 23–27 July 2019; pp. 916–919. [PubMed] [CrossRef] [Google Scholar]
]

8.Farag A., Lu L., Roth H.R., Liu J., Turkbey E., Summers R.M. A Bottom-Up Approach for Pancreas Segmentation Using Cascaded Superpixels and (Deep) Image Patch Labeling. *IEEE Trans. ImageProcess.* 2017;**26**:386–399. doi: 10.1109/TIP.2016.2624198. [PubMed][CrossRef] [Google Scholar]

9.Schlosser R.W. The Role of Systematic Reviews in Evidence-BasedPractice, Research and Development. [(accessed on 2 February 2021)];*Focus.* 2006 **15**:1–4. Available online: https://ktdrr.org/ktlibrary/articles_pubs/ncddrwork/focus/focus15 [Google Scholar]

10.Mallett R., Hagen-Zanker J., Slater R., Duvendack M. The Benefits andChallenges of Using Systematic Reviews in International Development Research. *J. Dev. Eff.* 2012;**4**:445–455.doi: 10.1080/19439342.2012.711342. [CrossRef] [Google Scholar]

11.Milton M.A.A. Automated Skin Lesion Classification Using Ensembleof Deep Neural Networks in ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge. [(accessed on 22 January2021)];*arXiv.* 2019 Available online: http://arxiv.org/abs/1901.108021901.10802 [Google Scholar]

Mrs. BABITHA S, M.E. is a Department of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. She has completed her M.E, in Rajalakshmi Engineering College in Computer Science and Engineering in 2013 from Chennai, Tamilnadu. She has done her B.E, CSE in G.K.M. College of Engineering and Technology from Chennai in the year 2011. Mrs. BABITHA S, has 1years of teaching experience and has 1 publications in International Conference.

Mr. PRAVIN S, BE., Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. I have completed my internship in web development at Drawlead. I had attended a course in Data Analytics. I have completed the python full stack course. I recently got placed in Drawlead.