

1 INTRODUCTION

1.1 Overview

In this proposed system we are able to identify the rate of breast cancer. By prediction of Breast cancer in healthcare we are able to take suitable measures accordingly to control the cancer tissue. Due to the large size of each image in the training dataset, we propose a technique which consists of two consecutive convolutional neural networks. By using deep neural networks we are predicting the cancer is benign or malignant. Users have feasibility to upload scanned images on a web page to know about the status.

1.2 Purpose

Breast cancer is one of the main causes of cancer death worldwide. Early diagnostics significantly increases the chances of correct treatment and survival, but this process is tedious and often leads to a disagreement between pathologists. Computer-aided diagnosis systems showed potential for improving diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible.

2 LITERATURE SURVEY

2.1 Existing problem

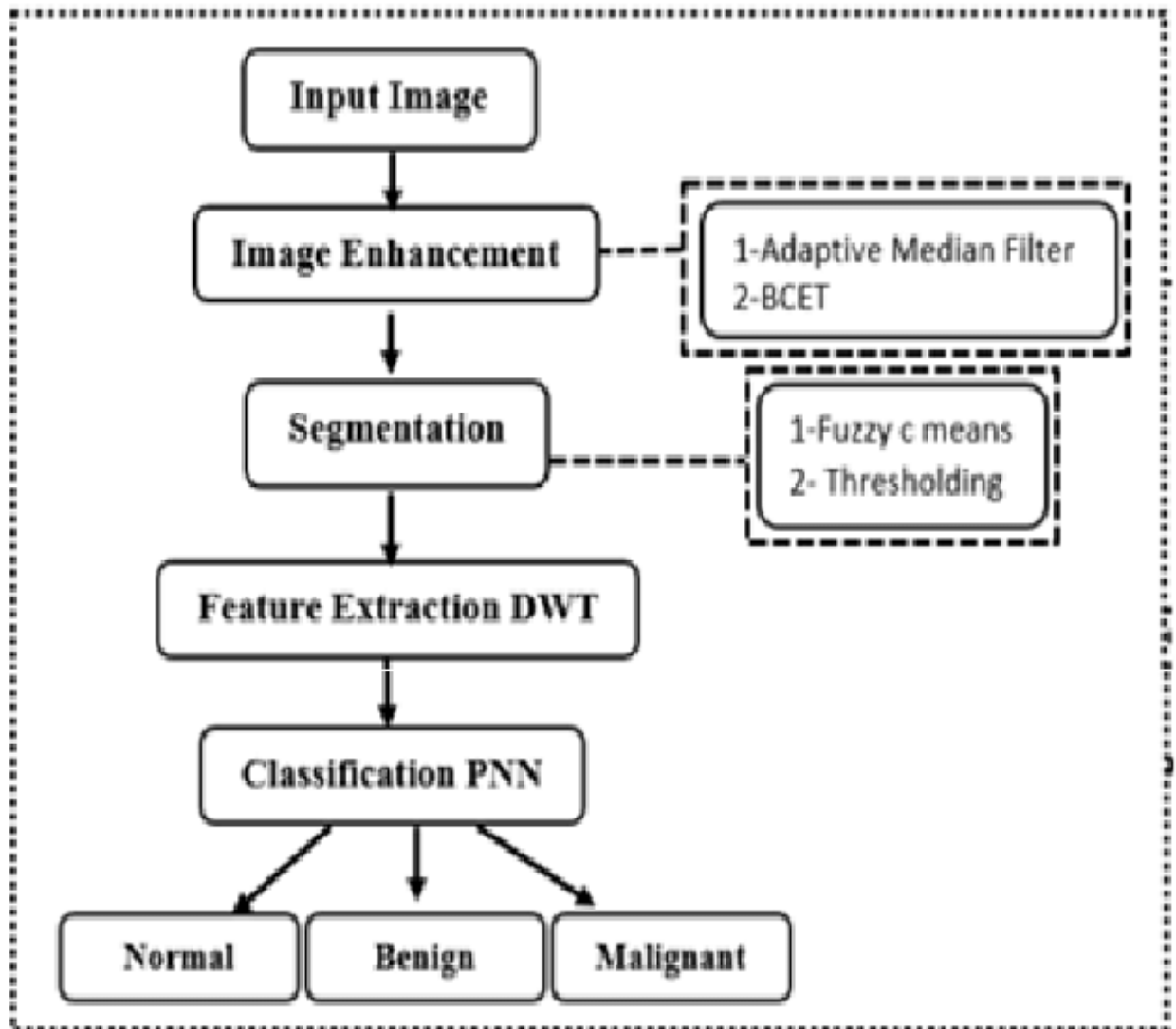
Identifying the disease in your body is important, the more early you realize the symptoms the more better but what if it's Cancer, A Disease with less exposed symptoms but the more delay you do in getting treated the more trouble you face , So Identifying Such Deadly Cancer takes priority, The Traditional methods of Diagnosis is a long process eating time which could have been used to treat the patient so efficient methods of identifying the Cancer are Important.

2.2 Proposed solution

We propose a technique which consists of two consecutive convolutional neural networks. By using deep neural networks we are predicting the cancer is benign or malignant. Users have feasibility to upload scanned images on a web page to know about the status.

3 THEORITICAL ANALYSIS

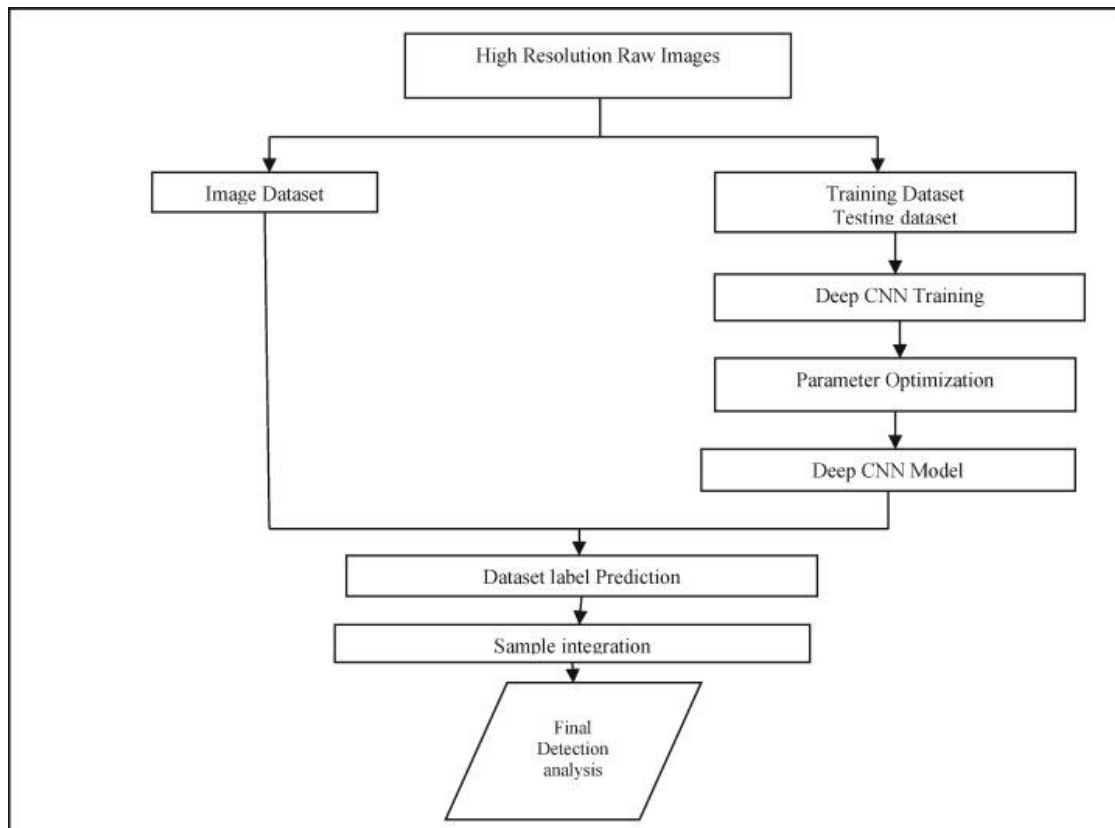
3.1 Block diagram



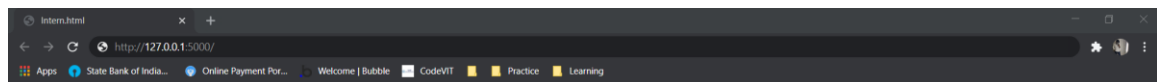
3.2 Hardware / Software designing

We used Tensor-Flow for developing our model. Training was done on Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz. The batch size had 21 epochs and then with each a rate of 31.

4 FLOWCHART



RESULT

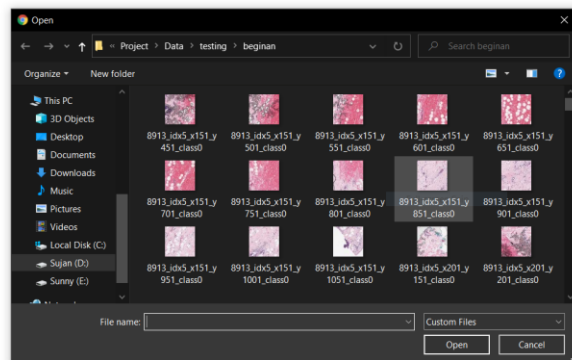


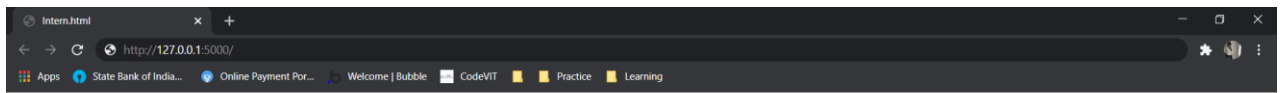
Check if you have Breat Cancer

Powered By : Deep Learning Techniques for Breast Cancer Risk Prediction using Python

Choose Image... Choose File No file chosen

Predict



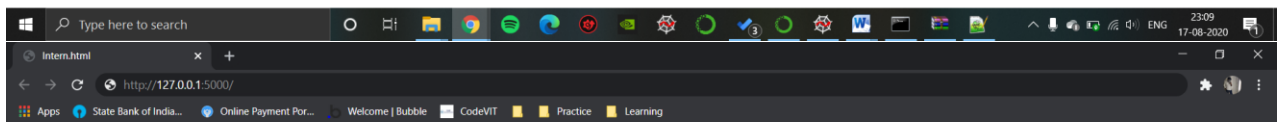


Check if you have Breat Cancer

Powered By : Deep Learning Techniques for Breast Cancer Risk Prediction using Python

Choose Image... Choose File 8913_idx5_x..._class0.png
Breat Cancer +ve

Full-screen Snap



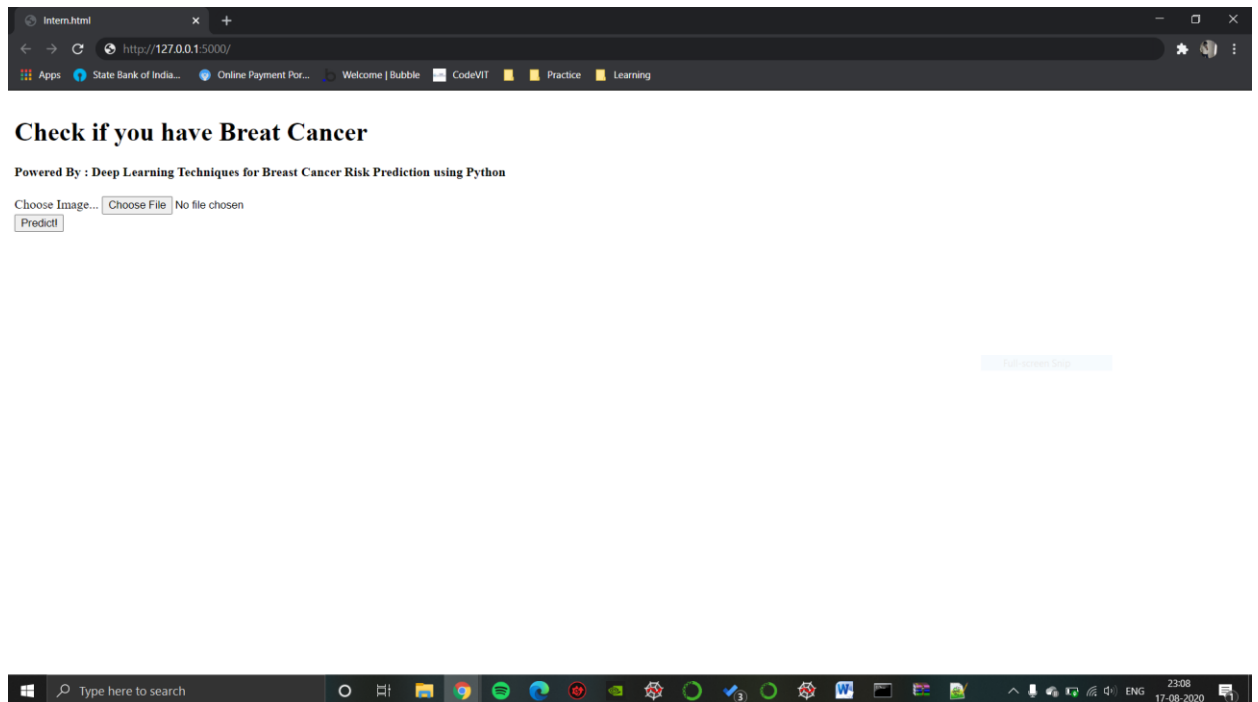
Check if you have Breat Cancer

Powered By : Deep Learning Techniques for Breast Cancer Risk Prediction using Python

Choose Image... Choose File 8913_idx5_x..._class0.png
Breat Cancer -ve

Full-screen Snap





7 ADVANTAGES & DISADVANTAGES

The Traditional methods of Diagnosis is a long process eating time which could have been used to treat the patient so efficient methods of identifying the Cancer are Important. Computer-aided diagnosis systems showed potential for improving diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible.

8 APPLICATIONS

Medical Field, Treating Cancer as Soon as possible contributes to the Life of the patient in many unimaginable ways. So, we should do our best in making the process as short and efficient as possible

APPENDIX

A. Source code

```
from keras.models import Sequential
```

```
from keras.layers import Dense
```

```
from keras.layers import Conv2D
```

```

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.models import load_model

import numpy as np

import cv2

from skimage.transform import resize

from keras.preprocessing.image import ImageDataGenerator

model=Sequential()

model.add(Conv2D(32,3,3,input_shape=(64,64,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.add(Dense(output_dim=128,activation='relu',init='random_uniform'))

model.add(Dense(output_dim=2,activation='sigmoid',init='random_uniform'))

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory(r'D:\Internship\Project\Data\training',target_size=(64,64),batch_size=32,class_mode='categorical')

x_test=train_datagen.flow_from_directory(r'D:\Internship\Project\Data\testing',target_size=(64,64),batch_size=32,class_mode='categorical')

print(x_train.class_indices)

model.fit_generator(x_train,samples_per_epoch=1000,epochs=25,validation_data=x_test,nb_val_samples=250)

model.save('sujan.h5')

```

```
model =load_model('sujan.h5')

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

def detect(frame):

    try:

        img= resize(frame,(64,64))

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

        if(np.max(img)>1):

            img =img/255.0

        prediction =model.predict(img)

        print (prediction)

        prediction_class = model.predict_classes(img)

        print(prediction_class)

    except AttributeError:

        print("shape not found")

frame= cv2.imread("8913_idx5_x251_y1251_class1.jpg")

data= detect(frame)
```