



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
Department of Computer Science and Engineering
Major Project Phase-II
Review-2

Smart Platform for Breast Cancer Classification using Deep Learning Techniques

- Guide : G.Mallikarjuna Rao,Professor

M.BHAVITA-20241A0591

M.SAHITHI-20241A0595

N.VARSHA-20241A0599

R.NIKITHA-20241A05B1



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Proposed Architecture:

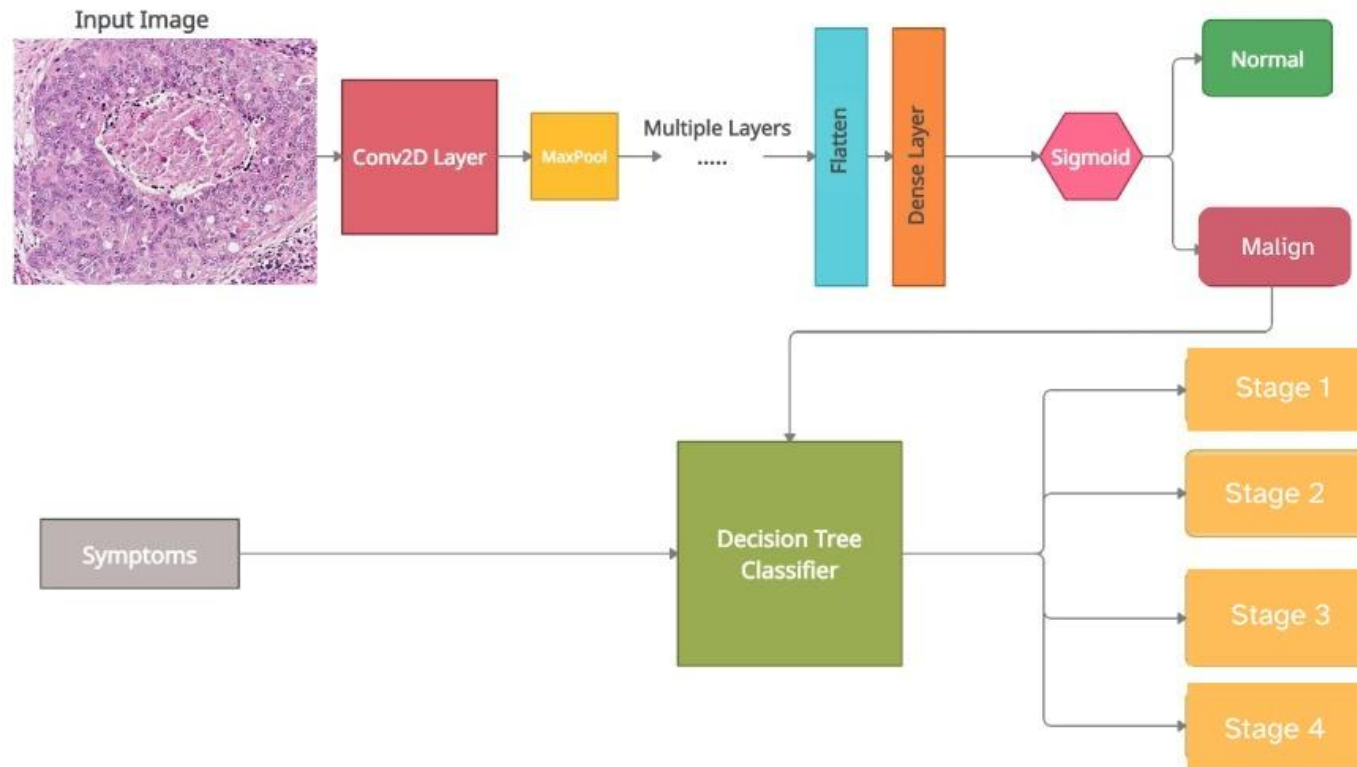
In our breast cancer classification project, we will utilize medical image dataset to identify intricate patterns. These datasets will be fed into various classifiers to discern whether a given instance is benign or malignant by using deep learning cnn models. Upon identifying patterns indicative of cancer, we will further classify the stage of breast cancer by using symptoms and classifying them with DecisionTreeClassifier on a symptoms dataset. Additionally, we will provide specific precautions and recommend medicines to mitigate the condition. To facilitate easy access and understanding, we will develop a user interface for online checking. However, it's important to note that this interface is meant for informational purposes only, and for optimal results, individuals should consult trained medical professionals.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Proposed Architecture:



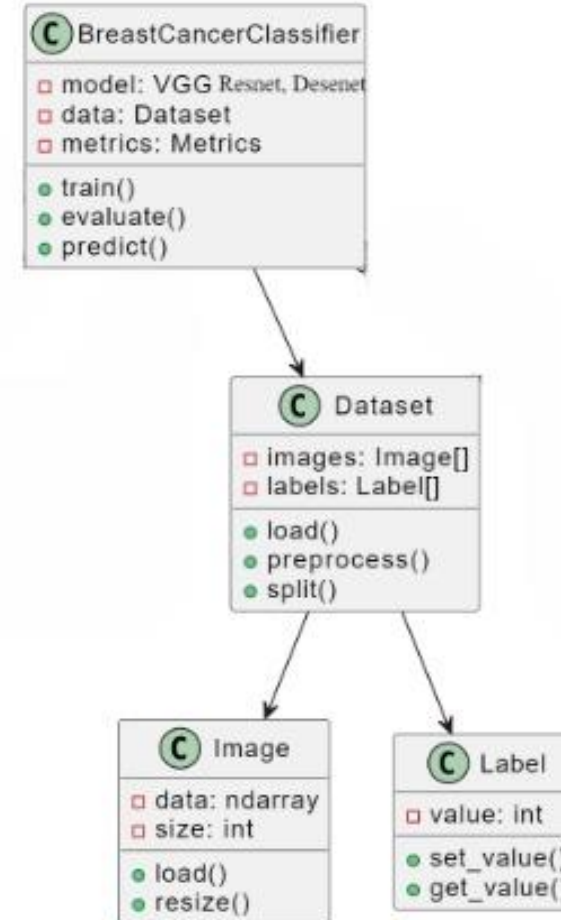


GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Module 1: Pre-Processing

We will start by gathering the necessary data for our project. After acquiring the data, we will perform several essential data preprocessing tasks including resizing, color conversion, normalization, labeling, and overall data preparation. This crucial step is vital to ensure that the data is appropriately formatted and ready for our analytical procedures.



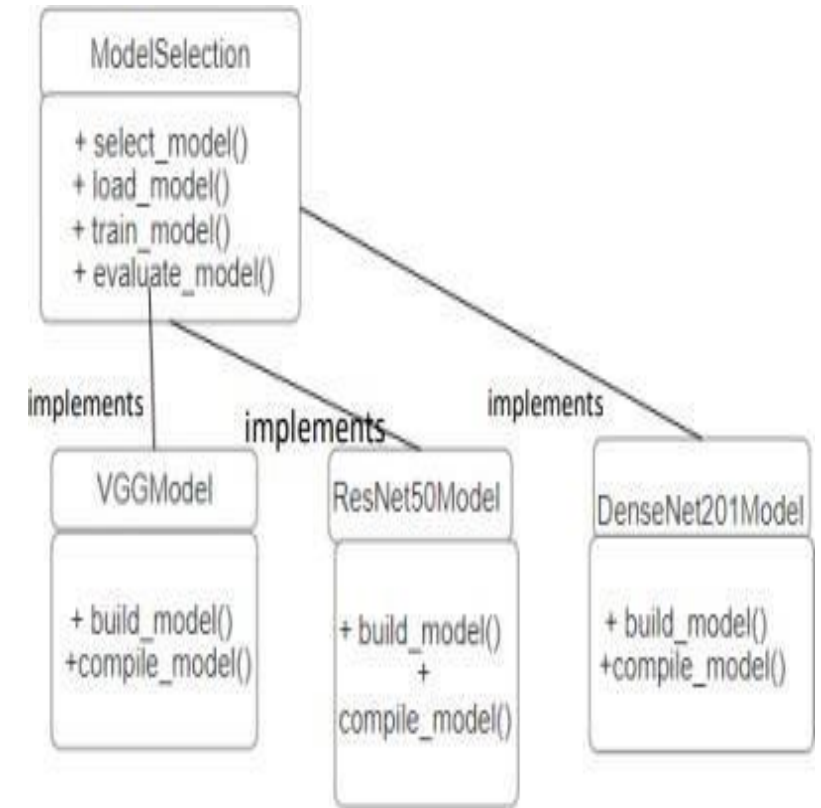


GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Module 2: Model Selection

- DenseNet201 may be a good choice if you have a large dataset and need a highly accurate model. It excels at capturing intricate image features
 - ResNet50 is a solid choice for most image classification tasks. It strikes a balance between model complexity and performance and can work well with medium-sized datasets.
 - VGG models, such as VGG16 or VGG19, are simpler and have fewer parameters compared to DenseNet201 and ResNet50. They may be suitable for smaller datasets or when computational resources are limited.
- DenseNet-201 achieves high accuracy while maintaining parameter efficiency, making it suitable for resource-constrained environments.





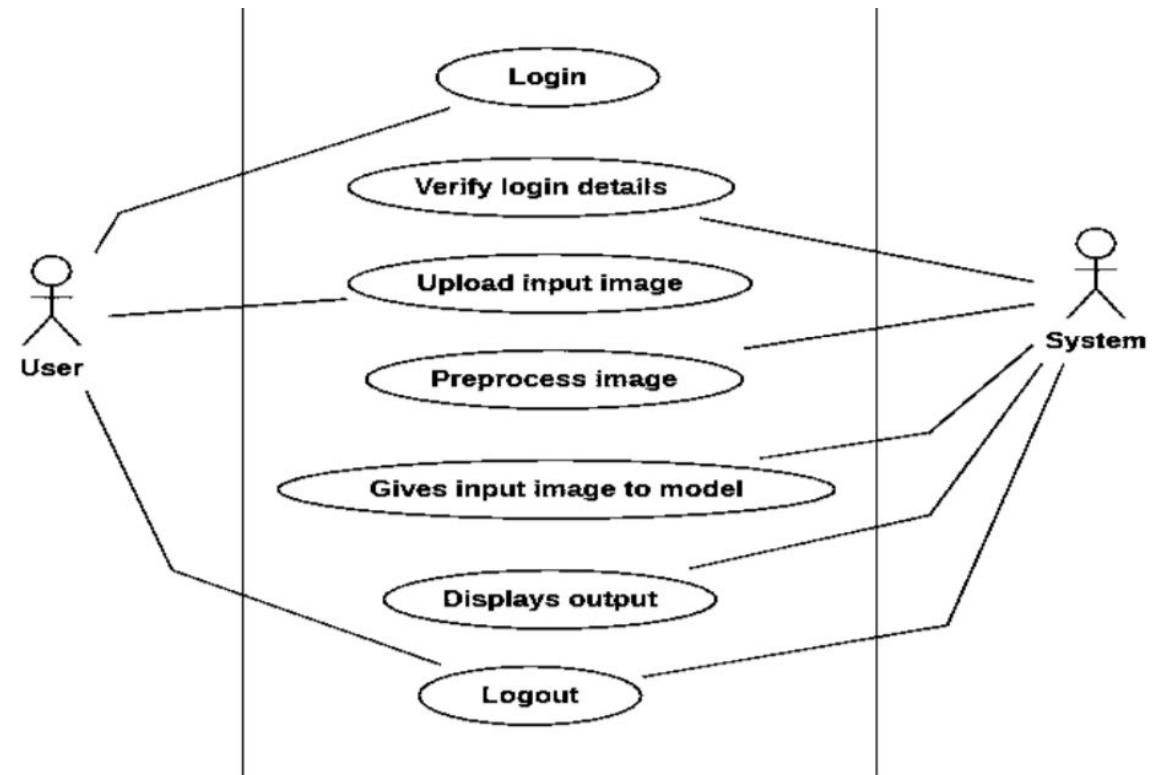
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Module 3: Testing and User Interface

The data is ready for model training and evaluation. The training process typically involves monitoring performance on the validation set to optimize the model. Finally, the model's generalization is assessed by evaluating it on the shuffled test set (X_{test} and Y_{test}) to determine how well it performs on unseen data.

Developing a platform where individuals can upload breast cancer images for classification. A website where users can upload their breast cancer image and result of it (cancerous or not and the stage) can be directly seen on screen.

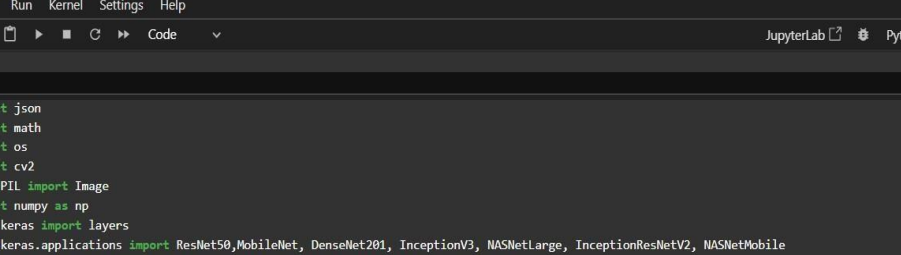




Experimentation:

importing libraries

Dividing into training and testing Data:



The screenshot shows a JupyterLab window titled "Densenet201" with a "Last Checkpoint: 3 hours ago" status. The interface includes a top menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a right-hand toolbar (JupyterLab icon, Python 3 (ipykernel) dropdown). The main area contains a code cell with the following Python code:

```
[1]: import json
import math
import os
import cv2
from PIL import Image
import numpy as np
from keras import layers
from keras.applications import ResNet50, MobileNet, DenseNet201, InceptionV3, NASNetLarge, InceptionResNetV2, NASNetMobile
from keras.callbacks import Callback, ModelCheckpoint, ReduceLROnPlateau, TensorBoard
from keras.callbacks import ReduceLROnPlateau, ModelCheckpoint
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical
from keras.models import Sequential
from keras.optimizers import Adam
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import cohen_kappa_score, accuracy_score
import scipy
from tqdm import tqdm
import tensorflow as tf
from keras import backend as K
import gc
from functools import partial
from sklearn import metrics
from collections import Counter
import json
import itertools
```

```

]: def Dataset_loader(DIR, RESIZE, sigmaX=10):
    IMG = []
    read = lambda imname: np.asarray(Image.open(imname).convert("RGB"))
    for IMAGE_NAME in tqdm(os.listdir(DIR)):
        PATH = os.path.join(DIR, IMAGE_NAME)
        _, ftype = os.path.splitext(PATH)
        if ftype == ".png":
            img = read(PATH)
            img = cv2.resize(img, (RESIZE, RESIZE))
            #img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            #img = cv2.equalizeHist(img)
            img = cv2.GaussianBlur(img, (5,5), cv2.BORDER_DEFAULT)
            arr=np.array(img)
            #arr1=np.repeat(arr[... , np.newaxis], 3, -1)
            IMG.append(arr)

    return IMG

benign_train = np.array(Dataset_loader("C:\\Users\\Varsha\\Desktop\\DataSources\\BreakHis 400X\\train\\benign", 224))
malign_train = np.array(Dataset_loader("C:\\Users\\Varsha\\Desktop\\DataSources\\BreakHis 400X\\train\\malignant", 224))
benign_test = np.array(Dataset_loader("C:\\Users\\Varsha\\Desktop\\DataSources\\BreakHis 400X\\test\\benign", 224))
malign_test = np.array(Dataset_loader("C:\\Users\\Varsha\\Desktop\\DataSources\\BreakHis 400X\\test\\malignant", 224))

```

100%			371/371	[00:09<00:00, 38.18it/s]
100%			777/777	[00:20<00:00, 37.57it/s]
100%			176/176	[00:04<00:00, 39.06it/s]
100%			369/369	[00:08<00:00, 42.95it/s]

```

]: # Breast Cancer: Malignant vs. Benign

```




GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Building densenet Model:

```
MajorProj.ipynb ☆
File Edit View Insert Runtime Tools Help Last edited on September 21

+ Code + Text

[ ] def build_model(backbone, lr=1e-4):
    model = Sequential()
    model.add(backbone)
    model.add(layers.GlobalAveragePooling2D())
    model.add(layers.Dropout(0.5))
    model.add(layers.BatchNormalization())
    model.add(layers.Dense(2, activation='softmax'))

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

    return model
K.clear_session()
gc.collect()

resnet = DenseNet201(
    weights='imagenet',
    include_top=False,
    input_shape=(224, 224, 3)
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/de
f_kernels_notop.h5
74836368/74836368 [=====] - 0s 0us/step
Model: 'sequential'

-----
Layer (type)                Output Shape                Param #
-----
densenet201 (Functional)    (None, 7, 7, 1920)         18321984

global_average_pooling2d (G (None, 1920)                0
lobalAveragePooling2D)

dropout (Dropout)           (None, 1920)                0

batch_normalization (BatchN (None, 1920)                7680
ormalization)

dense (Dense)               (None, 2)                   3842

-----
Total params: 18,333,506
Trainable params: 18,100,610
Non-trainable params: 232,896
-----
```




GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Building Resnet and VGG16 Model:

```
[ ] )  
    return model  
  
[ ] K.clear_session()  
    gc.collect()  
    pretrained_resnet_base = tf.keras.applications.resnet_v2.ResNet50V2(  
        include_top=False,  
        input_shape=(224, 224, 3),  
        weights="imagenet"  
    )  
    pretrained_resnet_base.trainable = False  
  
[ ] learn_control = ReduceLROnPlateau(monitor='val_acc', patience=5,  
                                     verbose=1, factor=0.2, min_lr=1e-7)  
  
# Checkpoint  
filepath="weights.best.hdf5"  
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')  
  
[ ] pretrained_resnet_base = tf.keras.applications.resnet_v2.ResNet50V2(  
    include_top=False,  
    input_shape=(224, 224, 3),  
    weights="imagenet"  
)  
    pretrained_resnet_base.trainable = False  
  
[ ] resnet50_model = build_model(pretrained_resnet_base, lr=1e-4)  
    resnet50_model.summary()
```

```
Jupyter VGG16 Last Checkpoint: last month  
File Edit View Run Kernel Settings Help  
+ - - - - - Code  
JupyterLab Python 3 (ipykernel)  
  
return model  
  
[9]: K.clear_session()  
     gc.collect()  
     pretrained_vgg_base = tf.keras.applications.VGG16(  
         include_top=False,  
         input_shape=(224, 224, 3),  
         weights="imagenet"  
     )  
  
     pretrained_vgg_base.trainable = False  
  
WARNING:tensorflow:From C:\Users\Varsha\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\backend.py:277: The name tf.reset_default_graph is deprecated. Please use tf.compat.v1.reset_default_graph instead.  
  
WARNING:tensorflow:From C:\Users\Varsha\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.  
  
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5  
58889256/58889256 [=====] - 19s 0us/step  
  
[10]: learn_control = ReduceLROnPlateau(monitor='val_acc', patience=5,  
                                       verbose=1, factor=0.2, min_lr=1e-7)  
  
# Checkpoint  
filepath="weights.best.hdf5"  
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')  
  
[11]: pretrained_vgg_base = tf.keras.applications.VGG16(  
        include_top=False,  
        input_shape=(224, 224, 3),  
        weights="imagenet"  
    )  
  
    pretrained_vgg_base.trainable = False  
  
[12]: vgg_model = build_model(pretrained_vgg_base, lr=1e-4)  
       vgg_model.summary()
```



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

DecisionTreeClassifier:

```
import pandas as pd

df = pd.read_excel('C:\\Users\\Varsha\\Desktop\\symptoms.xlsx')

print(df)

def changeYesNo(string):
    for d in df.index:
        if df.loc[d, string] == 'Yes':
            df.loc[d, string] = 1
        else:
            df.loc[d, string] = 0

changeYesNo('SmallLump')
changeYesNo('ChangesinSizeShape')
changeYesNo('SkinChanges')
changeYesNo('NippleChanges')
changeYesNo('LumpinUnderarm')
changeYesNo('Swelling')
changeYesNo('LargeLumpMass')
changeYesNo('BonePain')
changeYesNo('Cough')
changeYesNo('AbdominalPain')
changeYesNo('Headaches')
changeYesNo('WeightLossFatigue')
print(df)

feature_cols = ['SmallLump', 'ChangesinSizeShape', 'SkinChanges', 'NippleChanges',
                'LumpinUnderarm', 'Swelling', 'LargeLumpMass', 'BonePain',
                'Cough', 'AbdominalPain', 'Headaches', 'WeightLossFatigue']
X = df[feature_cols]
Y = df['Stage']
```

```
X = df[feature_cols]
Y = df['Stage']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,
                                                    shuffle=False)

import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from joblib import dump, load

# Saving the model

clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

print("Expected Output Predicted Output")
for d in range(len(y_test)):
    print(y_test.iloc[d], y_pred[d])

from joblib import dump, load
dump(clf, 'decisiontree.joblib')
clf = load('decisiontree.joblib')
```



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

User Interface:

```
app = Flask(__name__)

def changeYesNo(s):
    if s == 'Yes':
        return 1
    else:
        return 0

@app.route('/', methods=['GET'])
def hello_world():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    SmallLump = request.form['SmallLump']
    ChangesinSizeShape = request.form['ChangesinSizeShape']
    SkinChanges = request.form['SkinChanges']
    NippleChanges = request.form['NippleChanges']
    LumpinUnderarm = request.form['LumpinUnderarm']
    Swelling = request.form['Swelling']
    LargeLumpMass = request.form['LargeLumpMass']
    BonePain = request.form['BonePain']
    Cough = request.form['Cough']
    AbdominalPain = request.form['AbdominalPain']
    Headaches = request.form['Headaches']
    WeightLossFatigue = request.form['WeightLossFatigue']

    SmallLump = changeYesNo(SmallLump)
```

Loading...

```
Edit Selection View Go Run ... Search
app.py X
> Users > Varsha > Desktop > flask > app.py > ...
55 def predict():
81
82     arr = np.array([SmallLump, ChangesinSizeShape, SkinChanges, NippleChanges, LumpinUnderarm, Swelling, LargeLumpMass, BonePain, Cough, A
83     print(arr)
84
85     clf = load('decisiontree.joblib')
86     stage = clf.predict(arr)
87     file = request.files['file1']
88     file.save('static/file1.png') # Save as PNG
89     model = load_model("trained.h5")
90     img_path = 'static/file1.png' # File path with correct extension
91     img = cv2.imread(img_path)
92     tempimg = image.load_img(img_path, target_size=(224, 224))
93     img_array = image.img_to_array(tempimg)
94     img_array = preprocess_input(img_array)
95     img = np.expand_dims(img_array, axis=0)
96
97     # Make the prediction
98     prediction = model.predict(img)
99
100    # Check if the maximum predicted probability is greater than or equal to 0.5
101    if np.max(prediction) >= 0.5:
102        prediction_label = "Malignant"
103    else:
104        prediction_label = "Benign"
105
106    classification = 'Prediction : %s' % (prediction_label)
107
```



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

User Interface:

```
app.py
C: > Users > Varsha > Desktop > flask > app.py > predict
55 def predict():
107
108
109     if prediction_label == "Malignant":
110         phase = ''
111         for ele in stage:
112             phase += ele
113         classification = "Prediction: Malignant and Stage: %s" % (phase)
114
115         # Add recommendations based on cancer stage
116         recommendations = get_recommendations_based_on_stage(phase)
117         classification += "\n\nRecommendations:\n%s" % (recommendations)
118
119     return render_template("/predict.html", prediction_label=classification)
120
121 def get_recommendations_based_on_stage(stage):
122
123     recommendations = {
124         'Stage 1': 'Surgery, such as lumpectomy or mastectomy, often accompanied by radiation therapy; additional treatments like hormone t
125         'Stage 2': 'Combined treatment approach involving surgery, chemotherapy, radiation, and hormone therapy; specifics depend on tumor
126         'Stage 3': 'Combined treatment approach involving surgery, chemotherapy, radiation, and hormone therapy; specifics depend on tumor
127         'Stage 4': 'Metastatic or advanced breast cancer management with systemic therapies like chemotherapy, hormone therapy, targeted th
128     }
129
130     return recommendations.get(stage, 'No specific recommendations available for this stage.')
131
132 if __name__ == "__main__":
133     app.run(debug=True)
134
```



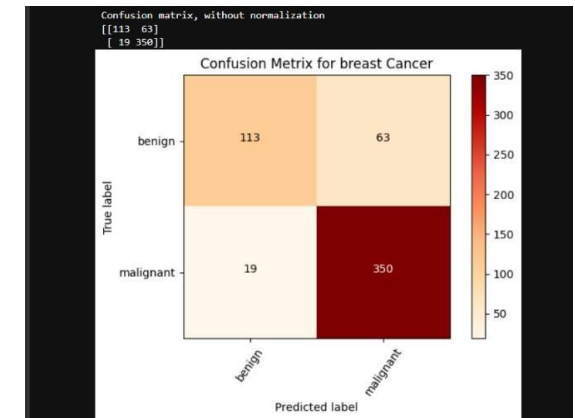
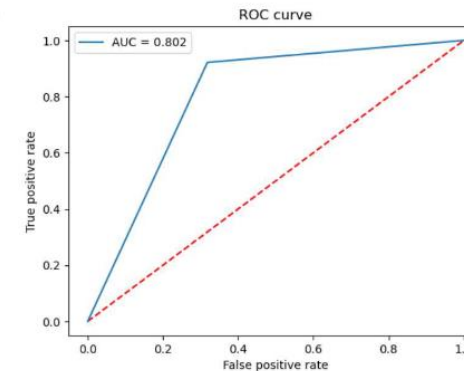
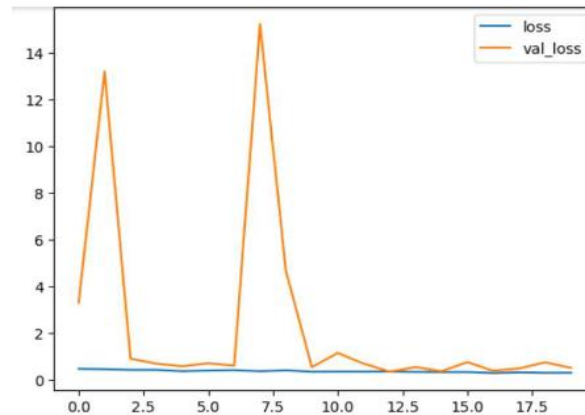
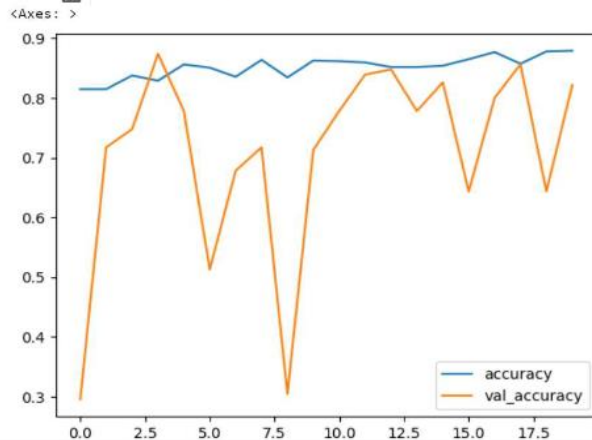
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Results and Discussions:

Densenet201:

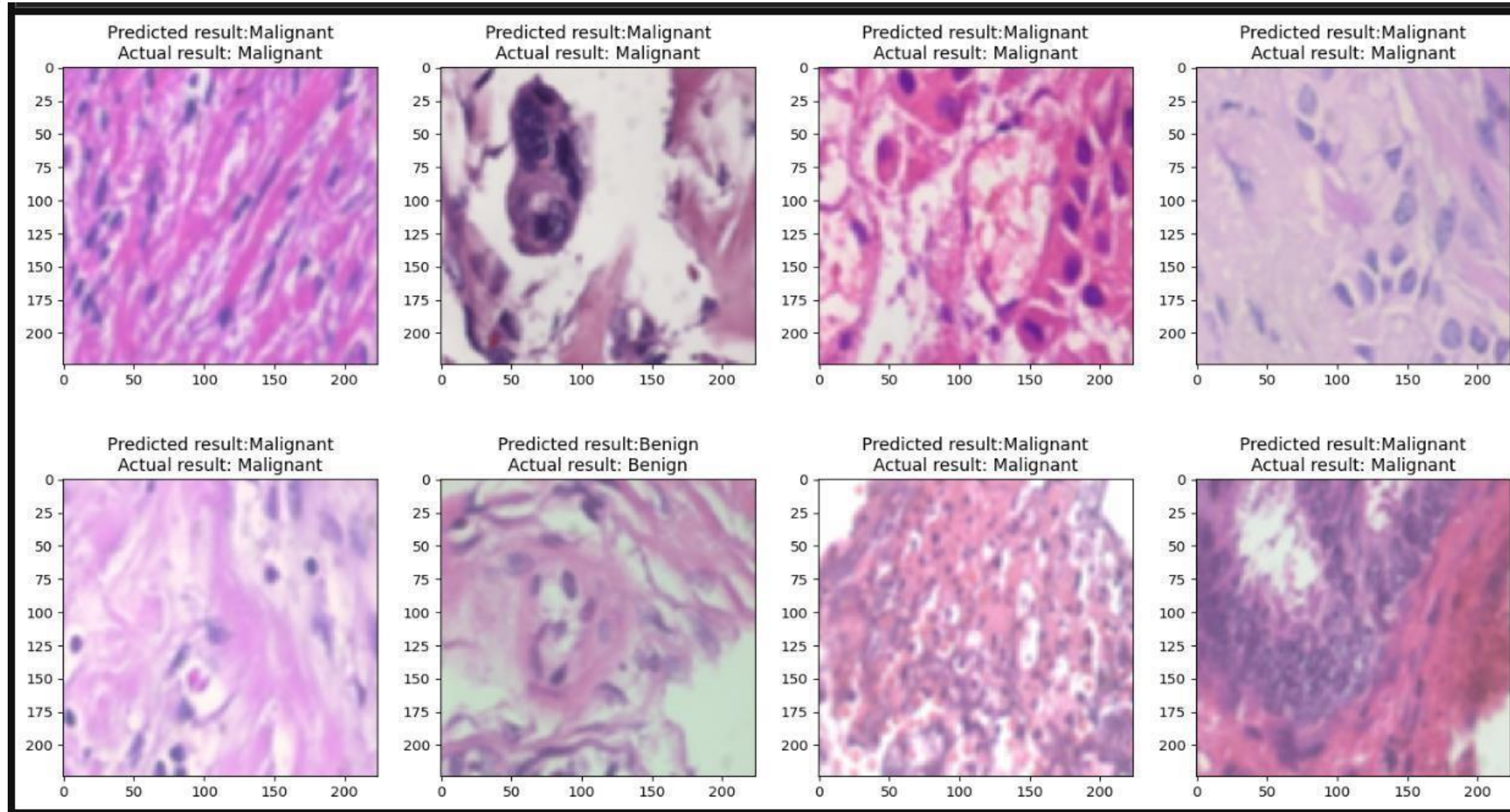
```
model.load_weights("C:\\Users\\Varsha\\Desktop\\DataSources\\BreakHis 400X")  
  
Y_val_pred = model.predict(x_val)  
accuracy_score(np.argmax(y_val, axis=1), np.argmax(Y_val_pred, axis=1))  
  
8/8 [=====] - 20s 2s/step  
0.9
```





GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering



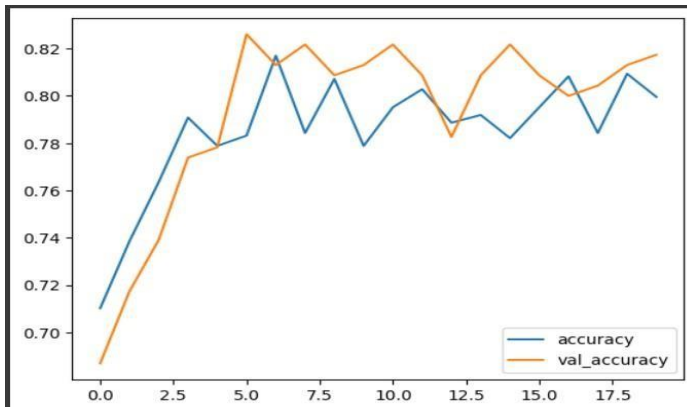


GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

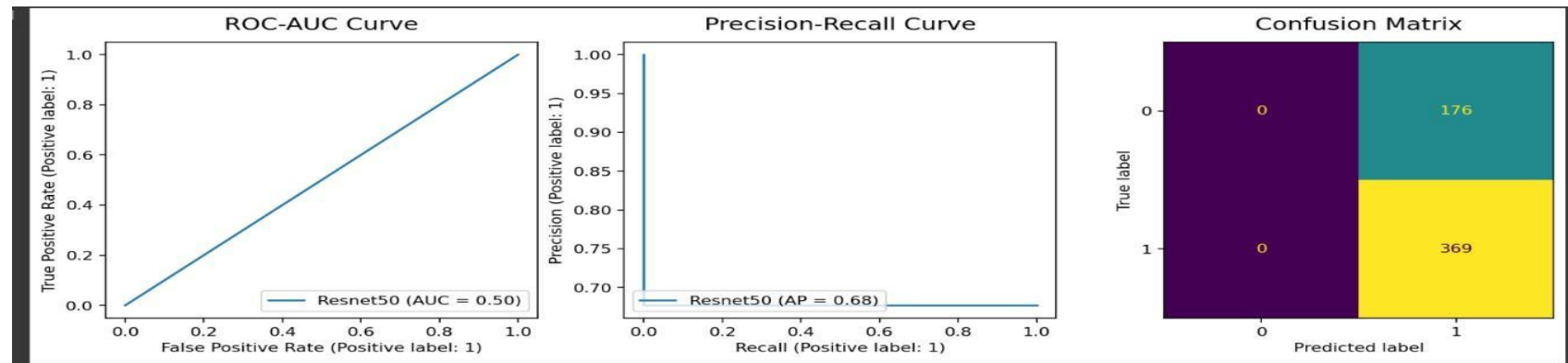
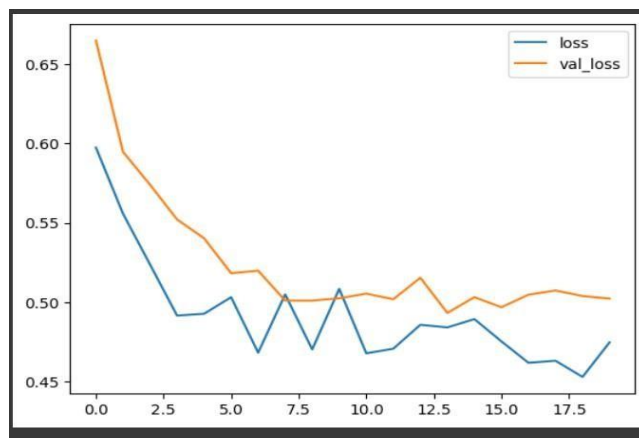
Resnet50

Accuracy:



```
[ ] Y_pred = resnet50_model.predict(X_test)
accuracy_score(np.argmax(Y_test, axis=1), np.argmax(Y_pred, axis=1))

18/18 [=====] - 78s 4s/step
0.8165137614678899
```





GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

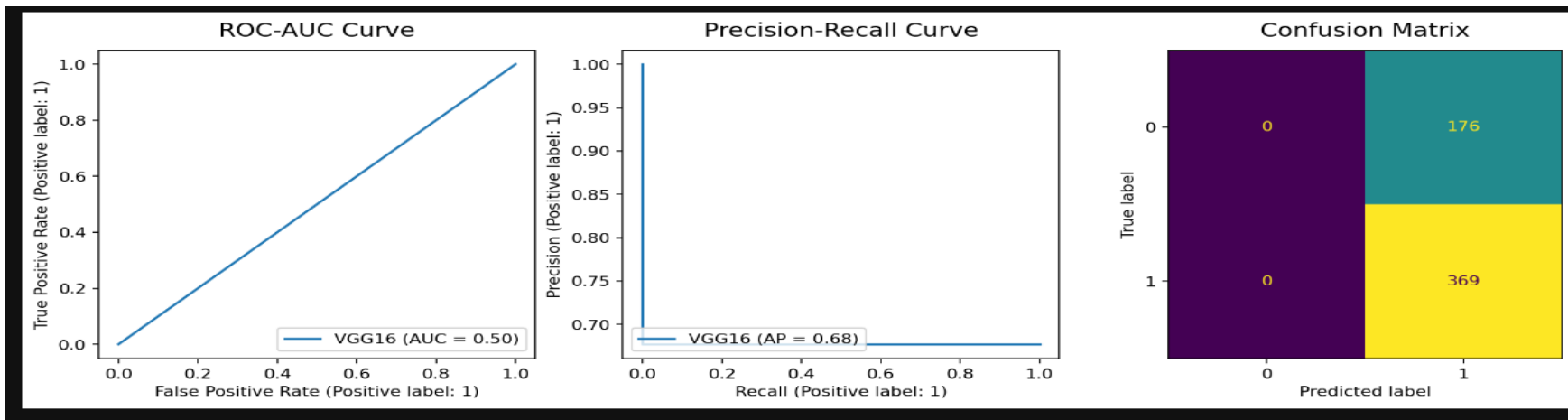
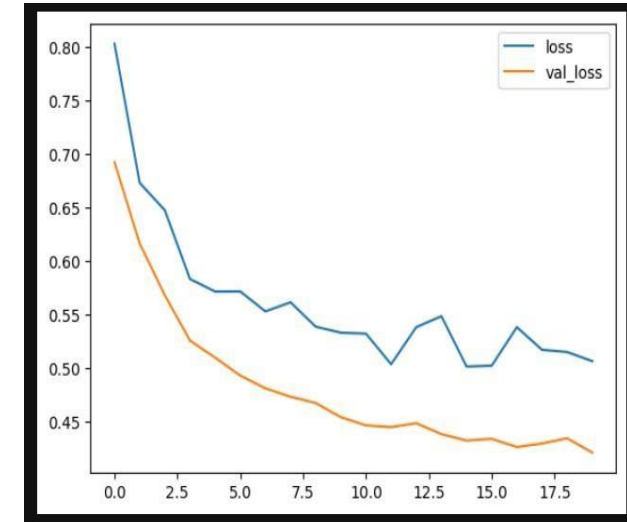
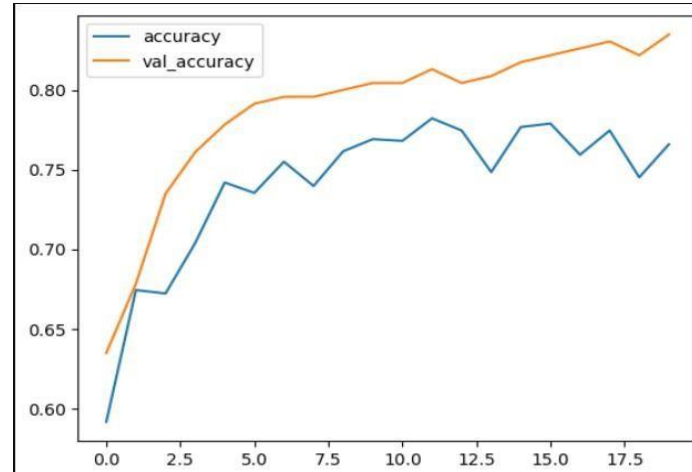
vgg16

Accuracy:

```
Y_pred = vgg_model.predict(X_test)
accuracy_score(np.argmax(Y_test, axis=-1), np.argmax(Y_pred, axis=-1))

18/18 [=====] - 77s 4s/step

0.8330275229357799
```





GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Results on diverse Datasets:

Type	Densenet201	Resnet50	vgg16
Mammography	<pre>1/1 [=====] 1/1 [=====] Validation Accuracy: 0.8</pre>	<pre>2/2 [=====] 1/1 [=====] ResNet50 Accuracy: 0.57142</pre>	<pre>VGG16 Accuracy: 0.5714285969734192</pre>
MRI	<pre>9/9 [=====] Validation Accuracy: 0.525</pre>	<pre>ResNet50 Accuracy: 0.4250000</pre>	<pre>VGG16 Accuracy: 0.487500</pre>



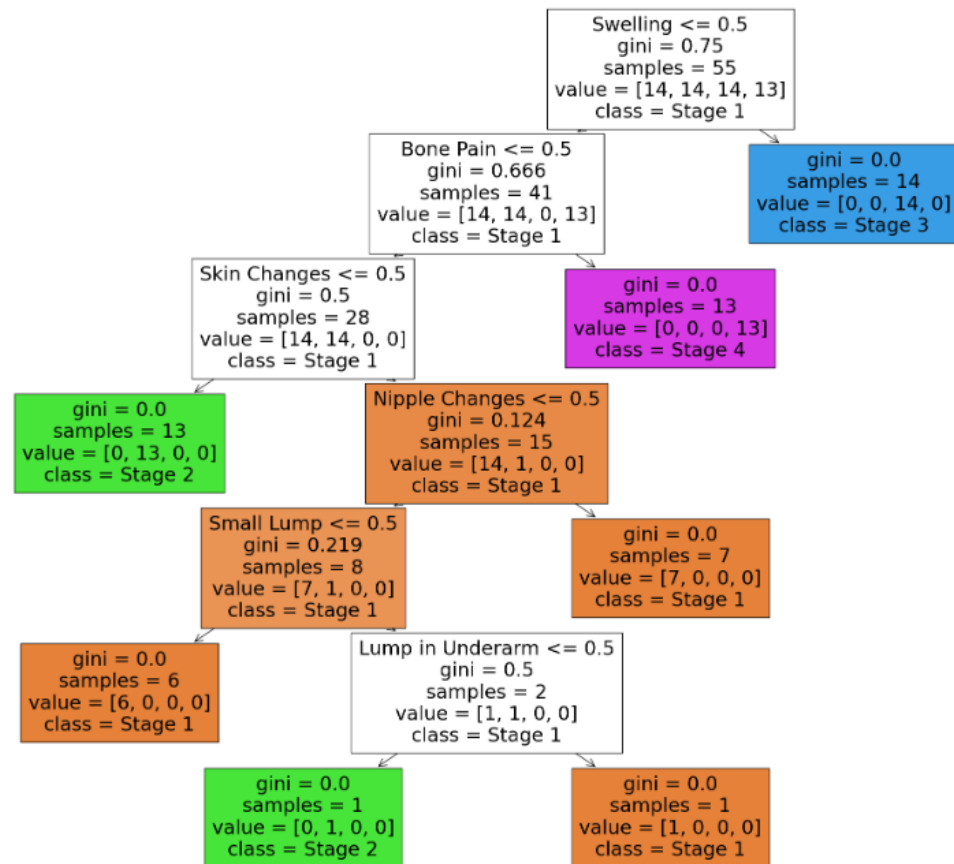
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Results for DecisonTreeClassifier:

Expected Output	Predicted Output
Stage 4	Stage 4
Stage 3	Stage 3
Stage 1	Stage 1
Stage 2	Stage 2
Stage 4	Stage 4
Stage 3	Stage 3
Stage 1	Stage 1
Stage 2	Stage 2
Stage 4	Stage 4
Stage 3	Stage 3
Stage 1	Stage 1
Stage 2	Stage 2
Stage 4	Stage 4
Stage 3	Stage 3

Accuracy: 1.0





GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

User Interface:

Breast Cancer Detection

Upload Image

Small Lump :

No

Changes in Size/Shape:

No

Skin Changes :

No

Nipple Changes :

No

LumpinUnderarm :

No

Swelling :

No

Large Lump/Mass :

No

Bone Pain:

No

Cough :

No

Abdominal Pain :

No

Headaches :

No

Weight Loss or Fatigue:

No

Choose File

No file chosen

Predict Cancer

For reference purposes only. Consult a professional.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Discussion

Integration of Heterogeneous Data Sets:

Heterogeneous data sources, including mammograms and mri images, were integrated by standardizing formats, addressing missing values, and normalizing data for consistency. Feature extraction techniques such as texture analysis and edge detection were employed across modalities, followed by fusion methods to create a unified representation. Challenges included aligning disparate data formats, selecting relevant features, and ensuring consistent normalization.

Stage Detection and Classification:

Stages 0-IV indicate the severity of cancer progression based on symptoms, crucial for determining appropriate interventions like surgery, chemotherapy, or radiation therapy. Understanding cancer stages aids in prognosis and treatment efficacy assessment.

User Interface Development:

The user-friendly interface allowed users to upload mammograms and ultrasound images, select classification models, and visualize results including predicted cancer stages and feature importance.

Patient Education:

Integrated educational resources provided patients with information on cancer stages, treatment options, and preventive measures, empowering informed decision-making for their healthcare journey.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

TestCases

	Test Cases	Expected Result	Actual Result	Test Result (Pass/Fail)
1	Check for Breast Cancer using Malign Dataset in Model	Presence of Breast Cancer	Breast Cancer is present	Pass
2	Check for Breast Cancer using Benign dataset in Model	Breast Cancer should not be present	Breast cancer is not present	Pass
3	Check for Breast Cancer by uploading in UI (1 st Malign image)	Breast Cancer should be present And correct stage is given	Breast cancer is present and correct stage is given acc to symptoms	Pass
4	Check for Breast Cancer by uploading in UI (2 nd Malign image)	Breast Cancer should be present And stage is given	Breast cancer is present and correct stage is given acc to symptoms	Pass



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Testcases 1 &2:

In CNN MODEL

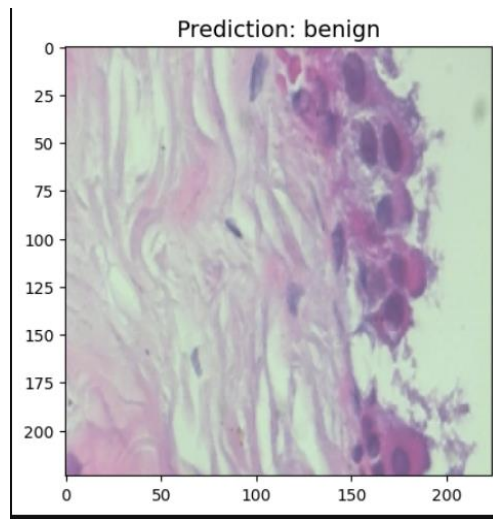


Fig. Sample Output of Normal Breast

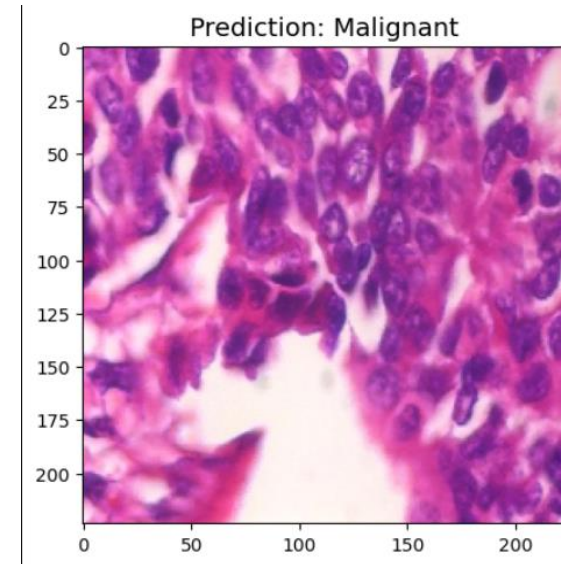


Fig. Sample Output of Cancerous Breast

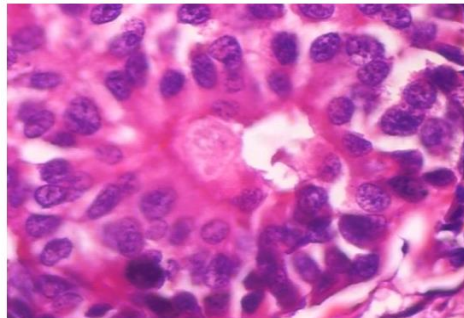


GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Test Case 3 and 4:

User Interface:



Prediction: Malignant and Stage: Stage 2 Recommendations: Combined treatment approach involving surgery, chemotherapy, radiation, and hormone therapy; specifics depend on tumor size and lymph node involvement.

Fig. Sample Output showing Malign and its stage as Prediction with recommendations.

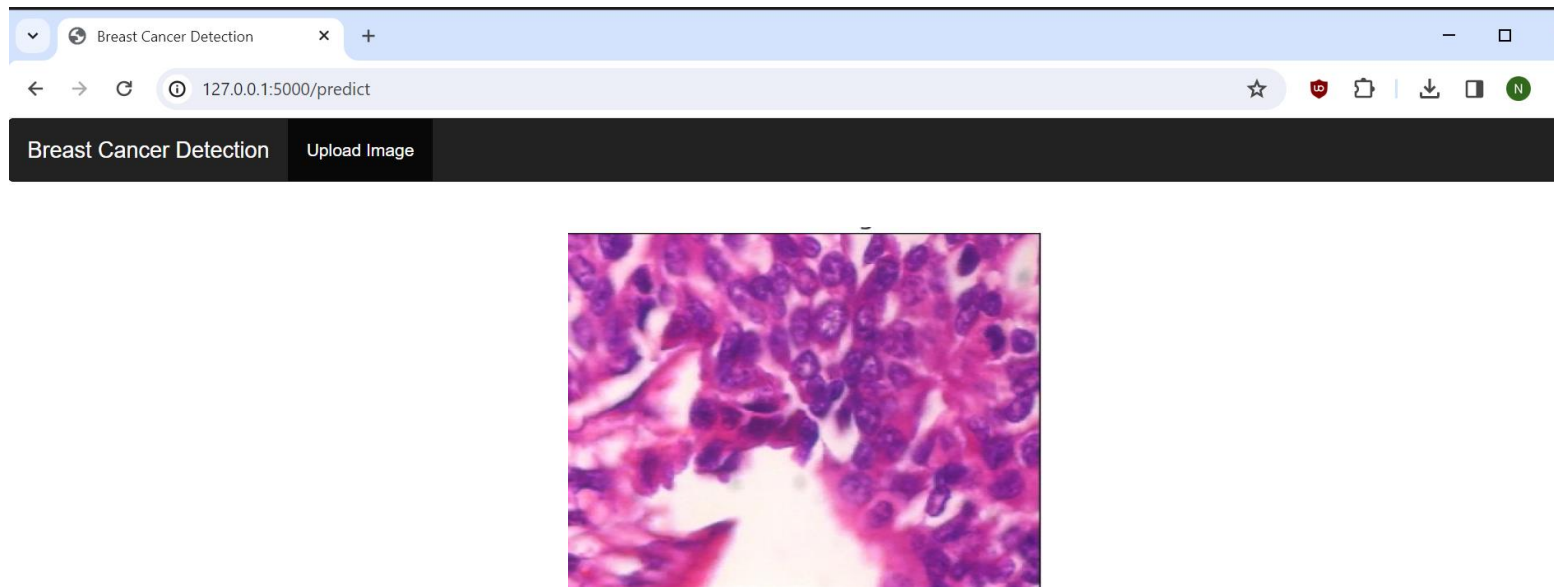


GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Test Case 3 and 4:

User Interface:



Prediction: Malignant and Stage: Stage 3 Recommendations: Combined treatment approach involving surgery, chemotherapy, radiation, and hormone therapy; specifics depend on tumor size and lymph node involvement.

Fig. Sample Output showing Malign and its stage as Prediction with recommendations.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Proposed Main Paper:-

This comprehensive review paper delves into the analysis of breast cancer diagnosis, focusing on key approaches such as convolutional neural networks and multi-modal integration. The examination encompasses modalities comparison, classifier evaluations, and advancements in deep learning. Within this context, the paper highlights both the progress made, and the challenges faced in the field. Addressing issues such as model interpretability and dataset limitations, the review aims to enhance accuracy for early breast cancer diagnosis. The exploration extends to advanced neural network architectures, transfer learning techniques, and the integration of multimodal data, providing insights into potential solutions. The synthesis of existing knowledge within the paper is strategically positioned to foster collaboration and innovation in breast cancer diagnosis. By bringing together insights into advancements, challenges, and emerging opportunities, the review aims to inspire collective efforts within the healthcare community.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

Conclusion:

In conclusion, our project on breast cancer classification utilizing deep learning methodologies marks a significant advancement in the realm of medical imaging and machine learning. Through the integration of diverse datasets encompassing biopsy images, mammograms, and mri data, alongside the development of sophisticated deep learning models to achieve optimal performance . The culmination of our efforts is manifested in a user-friendly interface empowering clinicians and patients to leverage insights for informed decision-making and proactive health management. The suggested model accurately predicts whether a given sample of image has cancer or is normal using a number of convolutional layers. This is extremely useful in the medical field for detecting Cancer in patients early and accurately. Early diagnosis is critical for saving a person's life by ensuring that the patient receives effective and timely treatment. Detection of Cancer phase/stage using Decision Tree also helps in the diagnosis and treatment effectively.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

References:

- [1] Murtaza, Ghulam, et al. "Deep learning-based breast cancer classification through medical imaging modalities: state of the art and research challenges." *Artificial Intelligence Review* 53.3 (2020): 1655-1720.
- [2] Vijayakumar, K., Vinod J. Kadam, and Sudhir Kumar Sharma. "Breast cancer diagnosis using multiple activation deep neural network." *Concurrent Engineering* 29.3 (2021): 275-284.ent care.
- [3] Yang, Zhanbo, et al. "EMS-Net: Ensemble of multiscale convolutional neural networks for classification of breast cancer histology images." *Neurocomputing* 366 (2019): 46-53.
- [4] Yala, Adam, et al. "A deep learning mammography-based model for improved breast cancer risk prediction." *Radiology* 292.1 (2019): 60-66.
- [5] Amrane, Meriem, et al. "Breast cancer classification using machine learning." 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT). IEEE, 2018.
- [6] Akselrod-Ballin, Ayelet, et al. "Predicting breast cancer by applying deep learning to linked health records and mammograms." *Radiology* 292.2 (2019): 331-342.
- [7] Krithiga, R., and P. Geetha. "Breast cancer detection, segmentation and classification on histopathology images analysis: a systematic review." *Archives of Computational Methods in Engineering* 28.4 (2021): 2607-2619.
- [8] Hu, Chuhan, et al. "Classification of breast cancer histopathological image with deep residual learning." *International Journal of Imaging Systems and Technology* 31.3 (2021): 1583-1594.
- [9] Sarosa, Syam Julio A., Fitri Utaminigrum, and Fitra A. Bachtiar. "Mammogram breast cancer classification using gray-level co-occurrence matrix and support vector machine." (SIET). IEEE, 2018.
- [10] Buda, Mateusz, et al. "A data set and deep learning algorithm for the detection of masses and architectural distortions in digital breast tomosynthesis images." *JAMA network open* 4.8 (2021): e2119100-e2119100.



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
Department of Computer Science and Engineering

THANK YOU!