GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
Department of Computer Science and Engineering
Major Project With Seminar

Smart Platform for Breast Cancer Classification using Deep Learning techniques

GUIDED BY: Mr G.Mallikarjuna Rao, Professor

BATCH : B6[CSE-B]

TEAM MEMBERS:

M.BHAVITA-20241A0591
M.SAHITHI-20241A0595
N.VARSHA-20241A0599
R.NIKITHA-20241A05B1

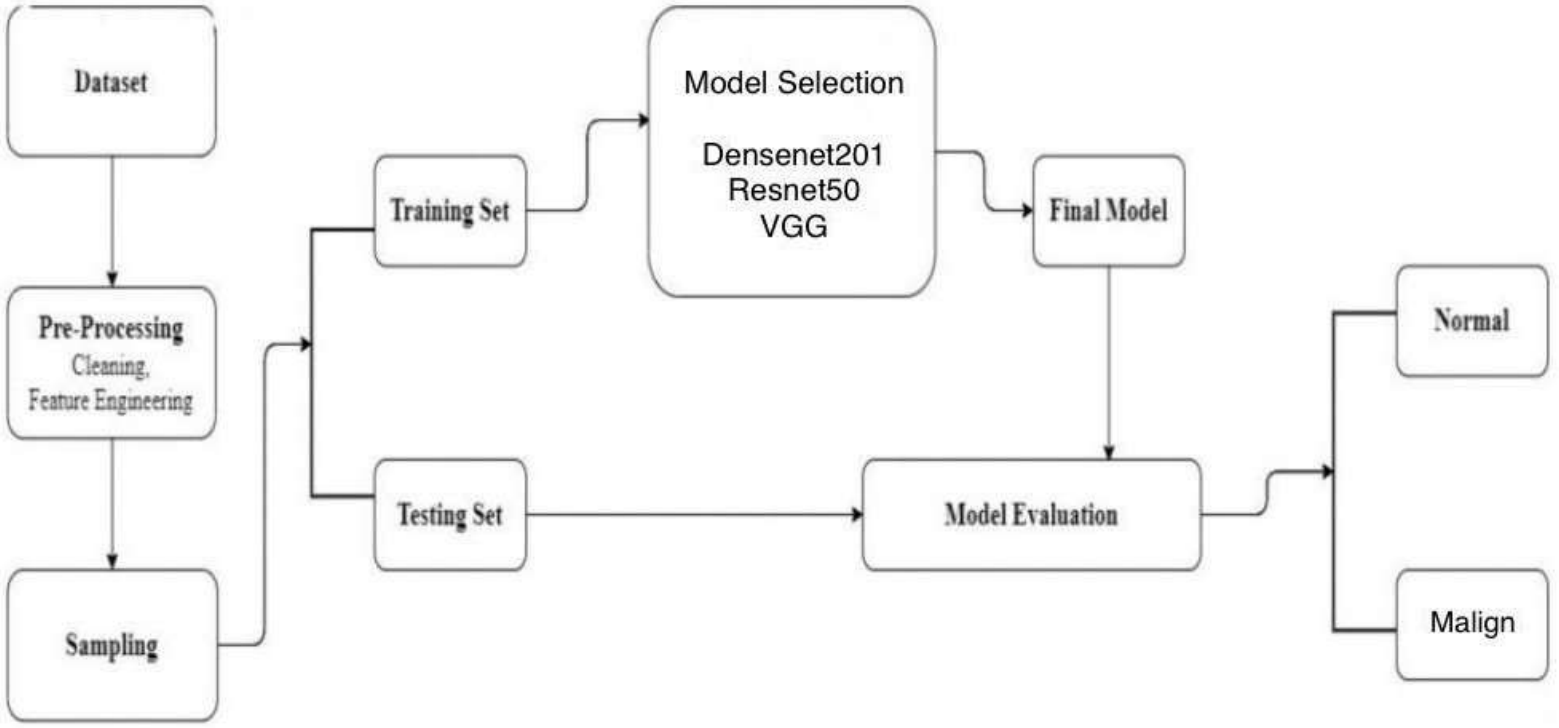
ABSTRACT

Breast cancer is prevalent and potentially a life-threatening disease that demands early detection for effective treatment and improved patient outcomes. Machine learning and deep learning techniques aid in breast cancer prediction by training algorithms on medical images, identifying specific features and classifying disease presence or absence.The project entails a multi-stage process. First, a diverse and well-curated dataset of medical images is collected, encompassing both cancerous and non-cancerous cases. Through meticulous preprocessing and data augmentation, the dataset's quality and diversity are optimized, preparing it for subsequent analysis. Various deep learning architectures, including Convolutional Neural Networks (CNNs) and possibly more specialized models for medical image analysis, are explored to extract intricate patterns and features from the images.Upon determining the optimal model, the system is deployed to classify new, unseen medical images. The model's predictions aid healthcare professionals in making informed decisions about breast cancer diagnosis and treatment planning.

Proposed System

The Breast Cancer Histopathological 400X (BreakHis 400X) dataset from Kaggle, which contains 1693 microscopic biopsy pictures of breast tumors, will be employed in the proposed approach.Both benign and malignant tumor samples can be found in the dataset.To enable an unbiased assessment of the model's performance, the dataset will be divided into training,validation, and test sets.The planned system's backbone architecture will be DenseNet201.It has shown outstanding performance in a variety of computer vision applications, making it suitable for medical picture analysis, including the categorization of breast cancer.A number of criteria, including accuracy, precision, and recall, will be used to assess the proposed system.These metrics offer a thorough evaluation of the model's capability to distinguish between benign and malignant breast cancers.DenseNet201 being a deeper and more densely connected architecture compared to other architecture offers **Improved Classification Accuracy**. Its ability to capture intricate patterns and feature reuse can lead to improved breast cancer classification accuracy.It’s connectivity pattern allows for **efficient feature extraction** and propagation throughout the network. leading to faster **training times and inference.**It offers Reduced Risk of Overfitting with its dense connections,especially when dealing with relatively small medical datasets like the BreakHis 400X dataset.We can also develop an ensemble model with existing and proposed models for higher accuracy.

Proposed Architecture



Design Methodology

Basic steps in constructing a model:

1.Data Collection
Data's quantity and quality will determine how precise the model is.
Use pre-collected data, by using datasets from UCI, Kaggle etc

2.Data Preparation
Accumulate data and prepare it for training.
Clean up the data (remove duplicates, fix errors, handle missing numbers, normalisation, convert data types, etc.)
Create data visualisations to assist in identifying significant correlations between class imbalances, variables or other exploratory analysis.
Make separate sets for training and evaluating.

3.Choose a Model
There are various algorithms for various tasks. Pick the best option

Methodology

4. Train the Model

Training's goal is to deliver an accurate answer or prediction as frequently as is practical.
Every period in the process is a training one.

5.Analyse the model

Use a measure or group of measures to "measure" the model's objective performance.
Although the model is currently being tuned, this unseen data is intended to be reasonably reflective of model performance in the real world.

6.Parameter Tuning

By modifying the parameters, you can make the model run better.
Initialization settings, learning rate, training step count, and distribution, among these simple model hyperparameters, may be used.

Module Description

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.

Module Description

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 is a deep learning architecture primarily designed for image classification and computer vision tasks. Its key purpose is to enhance information flow through the network by utilizing dense connections between layers. This approach not only improves gradient flow during training but also mitigates the vanishing gradient problem in very deep networks. DenseNet-201 achieves high accuracy while maintaining parameter efficiency, making it suitable for resource-constrained environments.

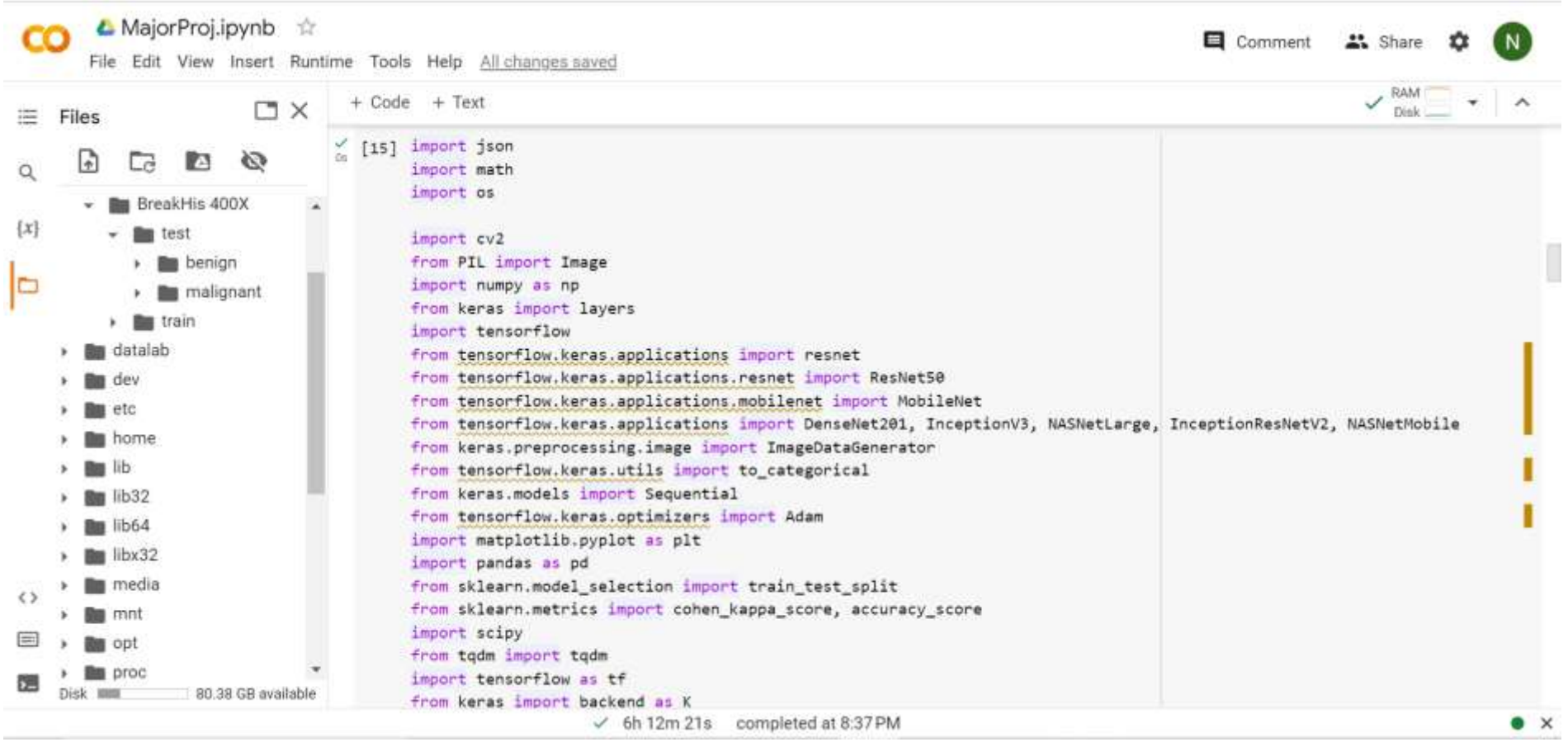
Module Description

Module-3 :Testing and Training

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. This process ensures that the model can make accurate predictions on new, real-world examples.
It aims to enhance performance on the BreakHis dataset, evaluating metrics like accuracy, precision,and recall for breast cancer classification.

Implementation and Execution:

Densenet201



MajorProj.ipynb

File Edit View Insert Runtime Tools Help Last edited on September 21

Comment Share N

+ Code + Text Connect

```
benign_train = np.array(Dataset_loader('/content/BreakHis 400X/train/benign', 224))
malign_train = np.array(Dataset_loader('/content/BreakHis 400X/train/malignant', 224))
benign_test = np.array(Dataset_loader('/content/BreakHis 400X/test/benign', 224))
malign_test = np.array(Dataset_loader('/content/BreakHis 400X/test/malignant', 224))

benign_train_label = np.zeros(len(benign_train))
malign_train_label = np.ones(len(malign_train))
benign_test_label = np.zeros(len(benign_test))
malign_test_label = np.ones(len(malign_test))

X_train = np.concatenate((benign_train, malign_train), axis = 0)
Y_train = np.concatenate((benign_train_label, malign_train_label), axis = 0)
X_test = np.concatenate((benign_test, malign_test), axis = 0)
Y_test = np.concatenate((benign_test_label, malign_test_label), axis = 0)

s = np.arange(X_train.shape[0])
np.random.shuffle(s)
X_train = X_train[s]
Y_train = Y_train[s]

s = np.arange(X_test.shape[0])
np.random.shuffle(s)
X_test = X_test[s]
Y_test = Y_test[s]
```

MajorProj.ipynb ☆

File Edit View Insert Runtime Tools Help Last edited on September 21

+ Code + Text

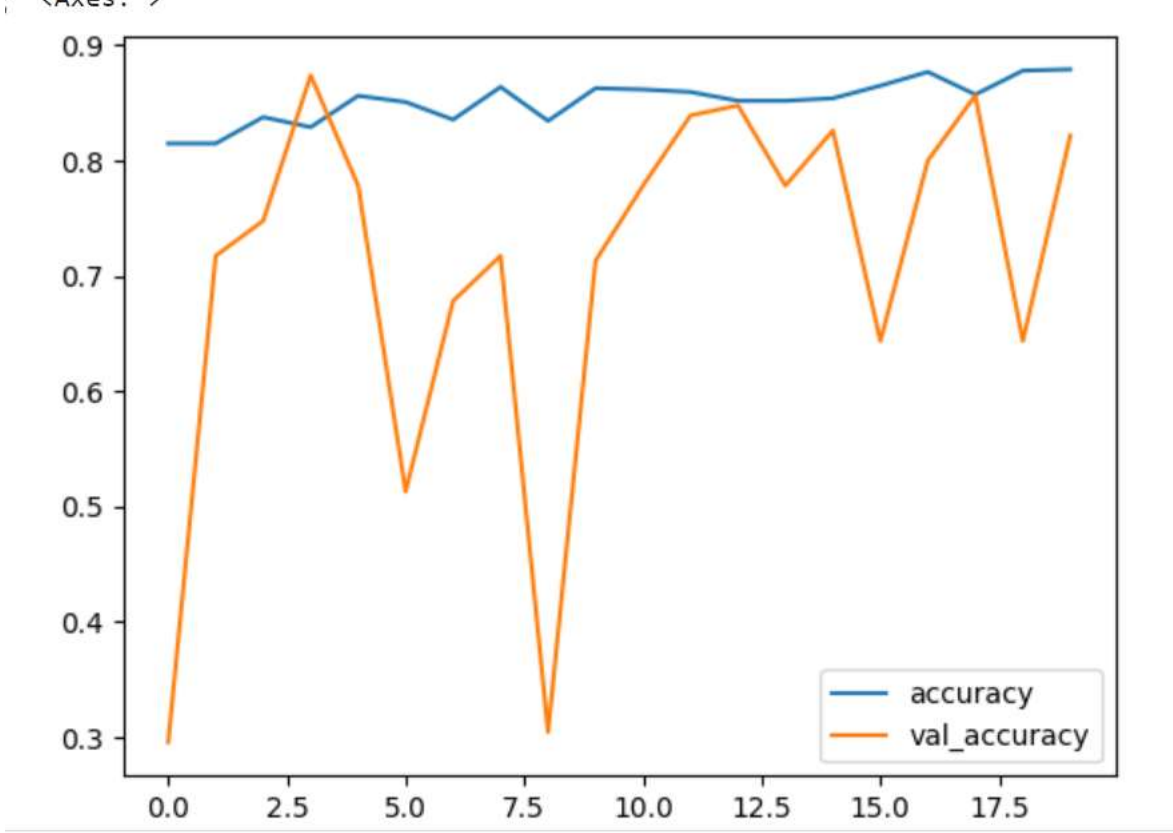
```
[ ]  def build_model(backbone, lr=1e-4):
         model = Sequential()
         model.add(backbone)
         model.add(layers.GlobalAveragePooling2D())
         model.add(layers.Dropout(0.5))
         model.add(layers.BatchNormalization())
         model.add(layers.Dense(2, activation='softmax'))


         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)
     )
```

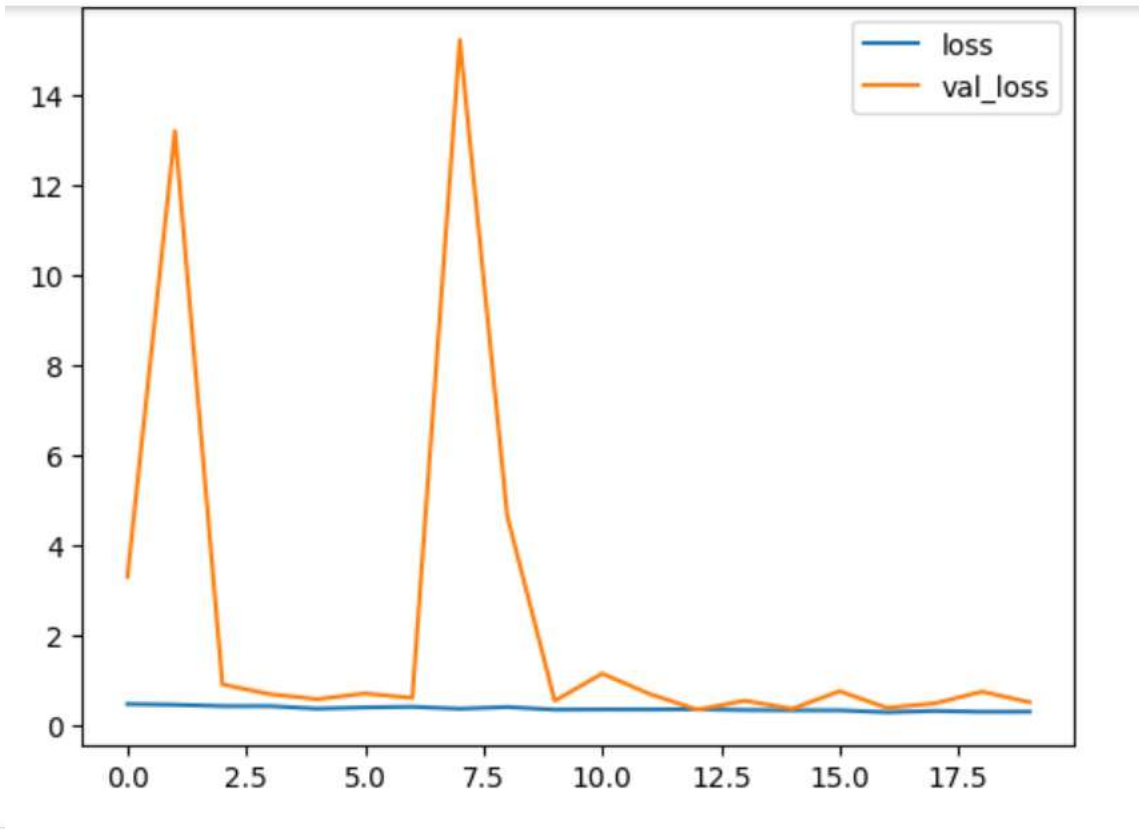
```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/de
f_kernels_notop.h5
74836368/74836368 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              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
_________________________________________________________________
```

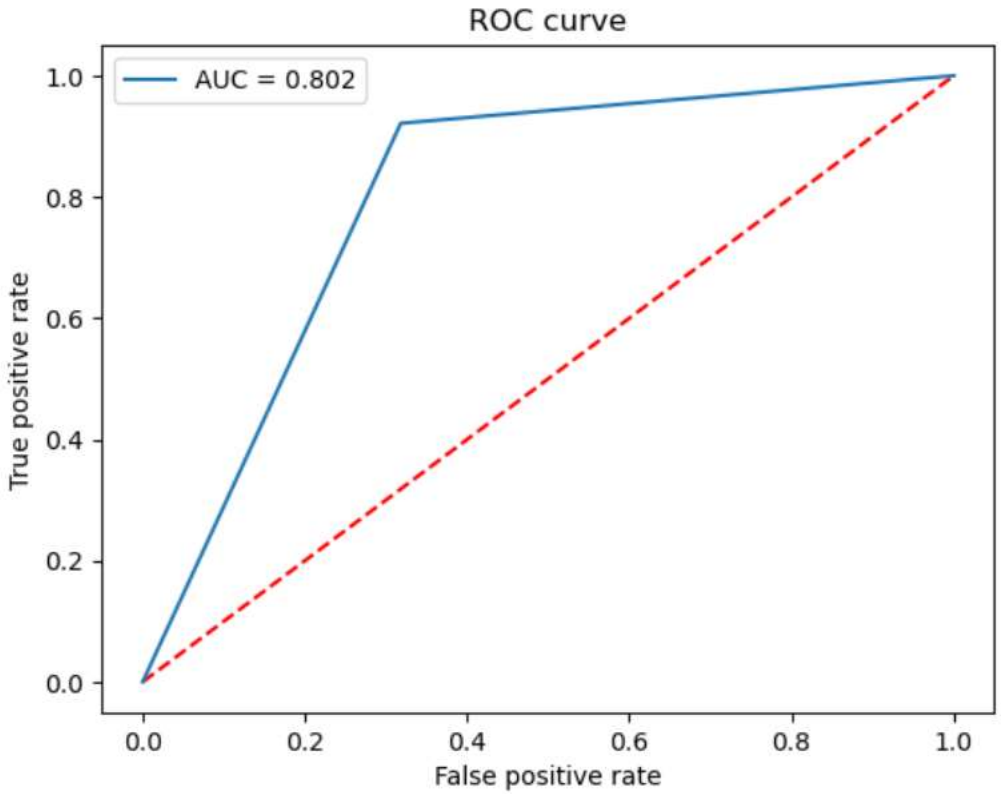
```
[ ]  model.load_weights("/content/BreaKHis 400X")

     Y_val_pred = model.predict(x_val)
     accuracy_score(np.argmax(y_val, axis=1), np.argmax(Y_val_pred, axis=1))
```

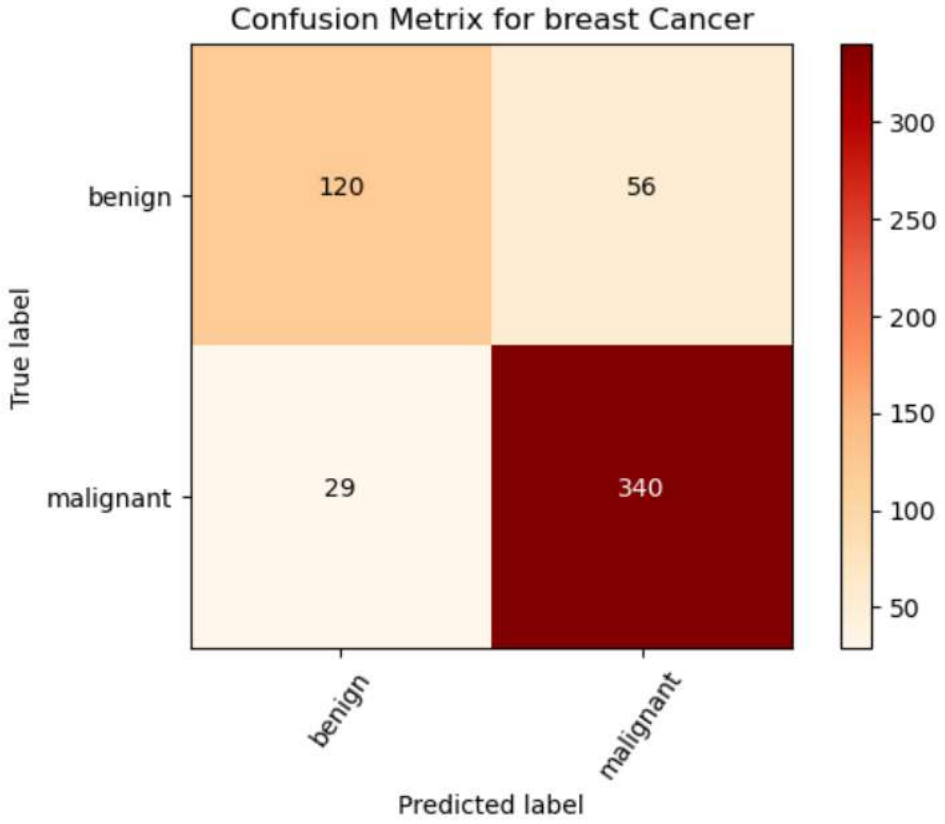
```
8/8 [==============================] - 53s 6s/step
0.8739130434782608
```

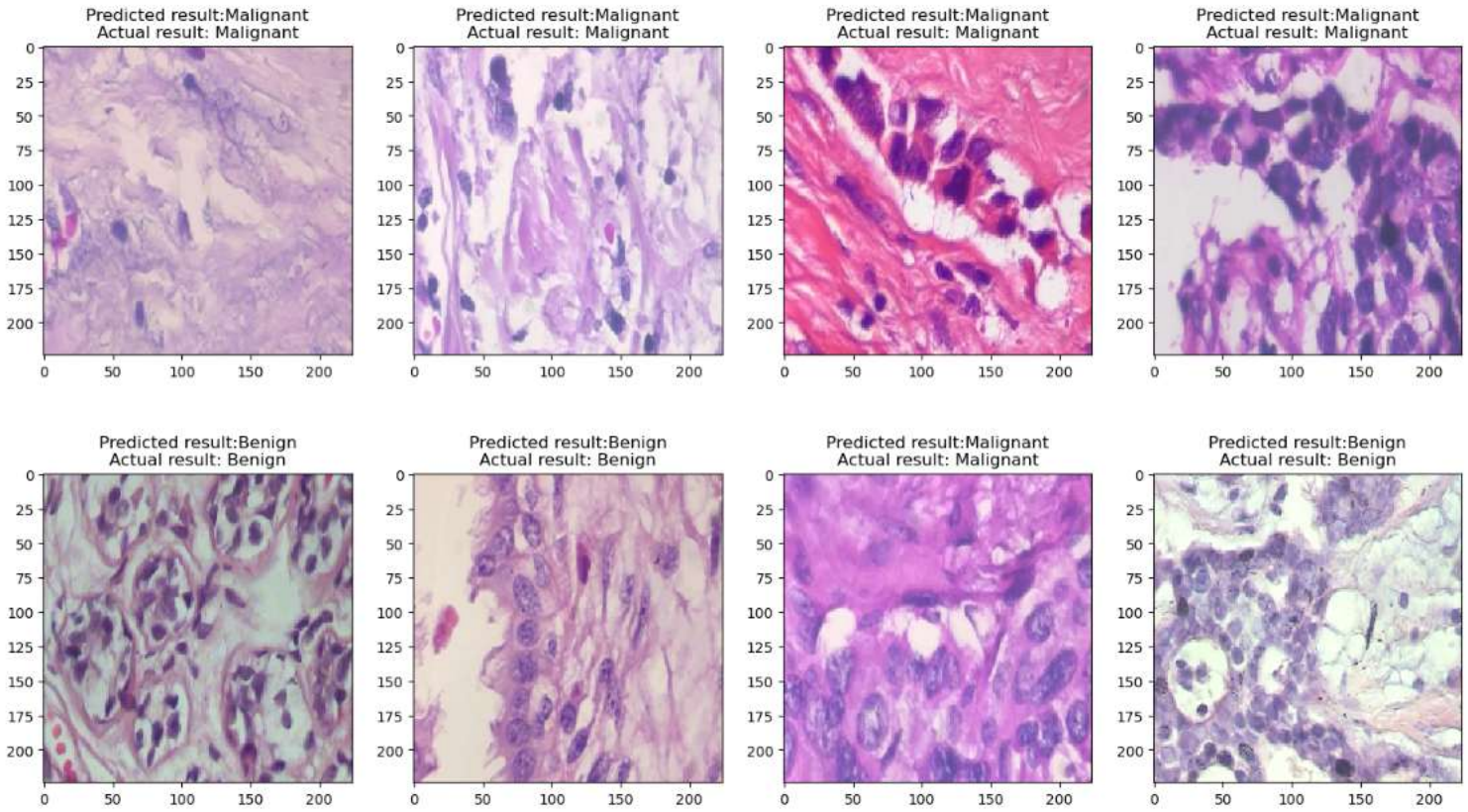
```
<Axes: >
```






ROC curve
AUC = 0.802
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


Confusion Metrix for breast Cancer
benign
malignant
True label
120
56
29
340
benign
malignant
Predicted label
300
250
200
150
100
50


Predicted result:Malignant
Actual result: Malignant
Predicted result:Malignant
Actual result: Malignant
Predicted result:Malignant
Actual result: Malignant
Predicted result:Malignant
Actual result: Malignant
Predicted result:Benign
Actual result: Benign
Predicted result:Benign
Actual result: Benign
Predicted result:Malignant
Actual result: Malignant
Predicted result:Benign
Actual result: Benign

Resnet50

```
[ ] 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
```

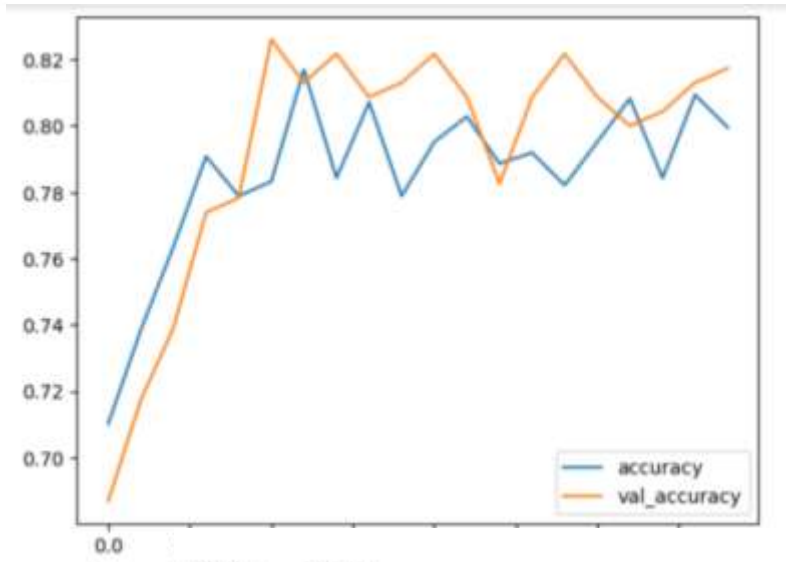
```
resnet50_model = build_model(pretrained_resnet_base ,lr = 1e-4)
resnet50_model.summary()
```

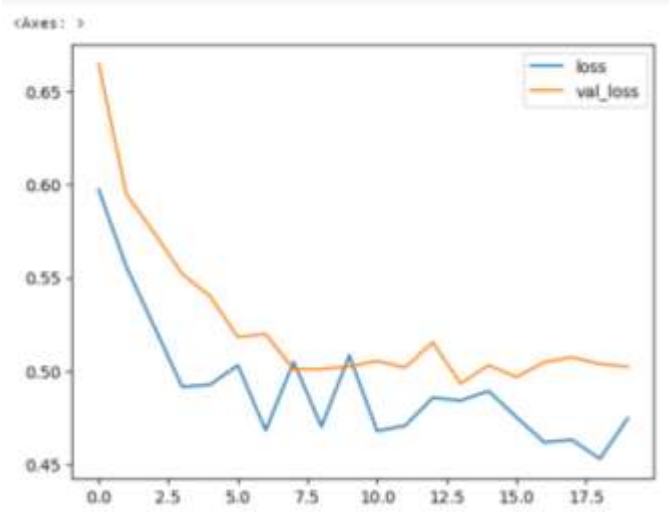
```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50v2 (Functional)     (None, 7, 7, 2048)        23564800

 global_average_pooling2d_1  (None, 2048)              0
  (GlobalAveragePooling2D)

 dropout_1 (Dropout)         (None, 2048)              0

 batch_normalization_1 (Bat  (None, 2048)              8192
 chNormalization)

 dense_1 (Dense)             (None, 2)                 4098

=================================================================
Total params: 23577090 (89.94 MB)
Trainable params: 8194 (32.01 KB)
Non-trainable params: 23568896 (89.91 MB)
_________________________________________________________________
```

Accuracy:

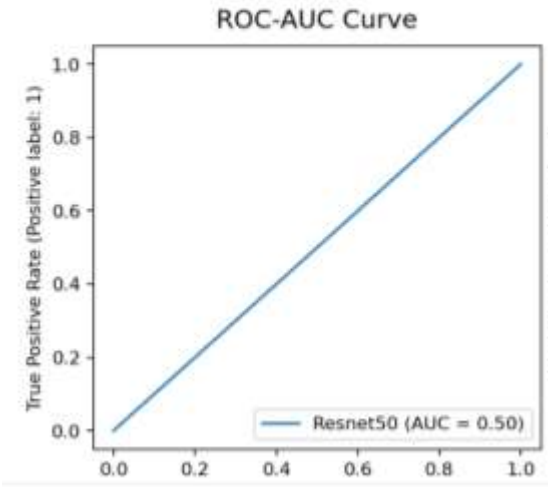
```
Y_pred = resnet50_model.predict(X_test)
accuracy_score(np.argmax(Y_test, axis=1), np.argmax(Y_pred, axis=1))
```

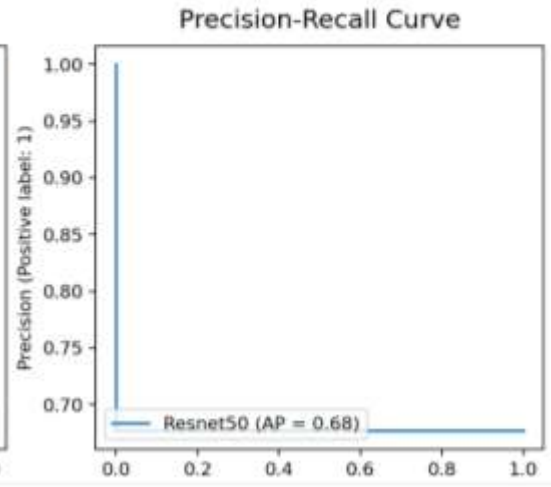
```
18/18 [==============================] - 78s 4s/step
0.8165137614678899
```

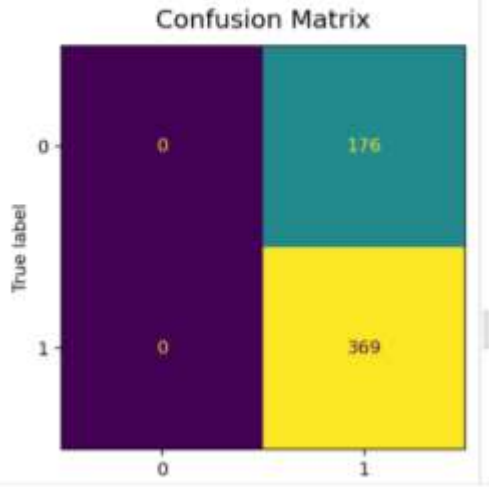





```
ROC-AUC:    0.84915
Accuracy:   0.81651
Loss:       0.50265
```







VGG16

```
]: vgg_model = build_model(pretrained_vgg_base, lr=1e-4)
   vgg_model.summary()
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_ra
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 global_average_pooling2d (  (None, 512)               0
 GlobalAveragePooling2D)

 dropout (Dropout)           (None, 512)               0

 batch_normalization (Batch  (None, 512)               2048
 Normalization)

 dense (Dense)               (None, 2)                 1026

=================================================================
Total params: 14717762 (56.14 MB)
Trainable params: 2050 (8.01 KB)
Non-trainable params: 14715712 (56.14 MB)
_________________________________________________________________
```

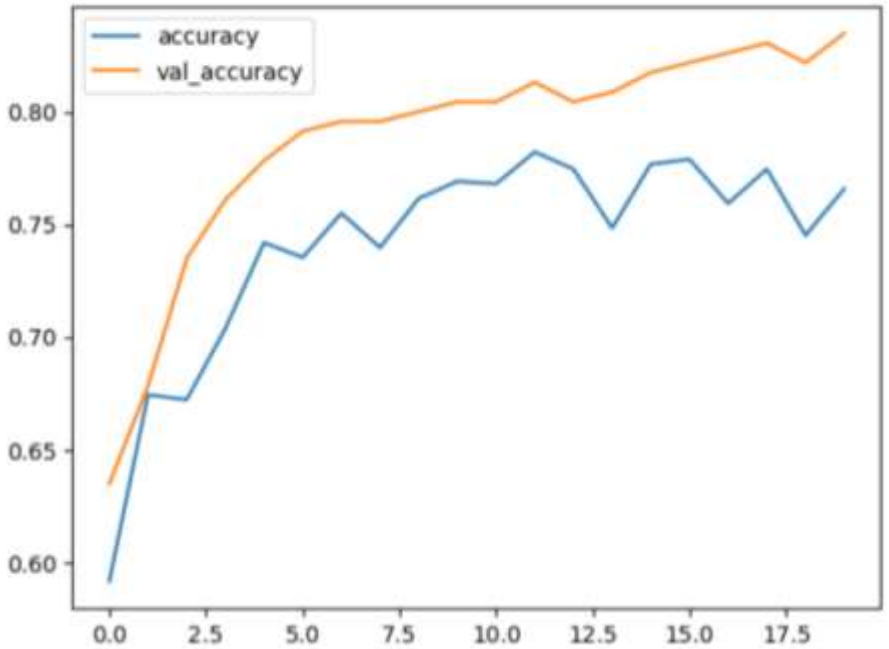
Accuracy:

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

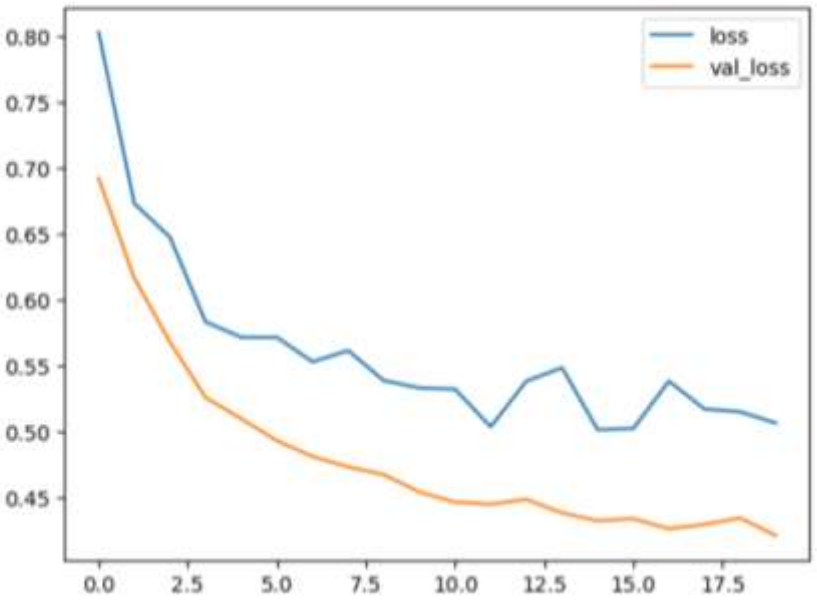
```
18/18 [==============================] - 77s 4s/step
```

```
[19]: 0.8330275229357799
```

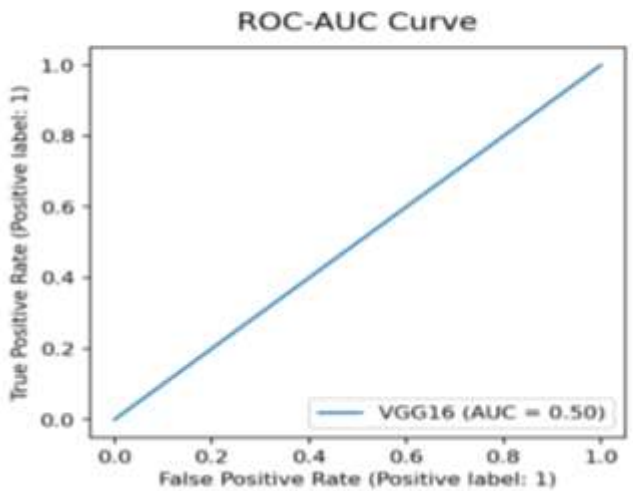
<Axes: >

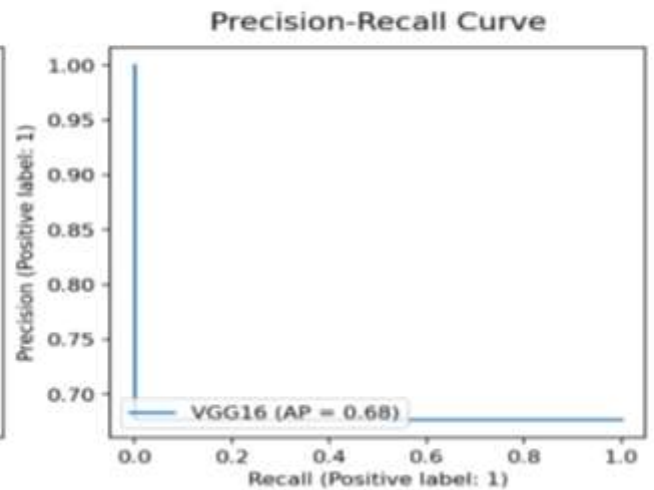


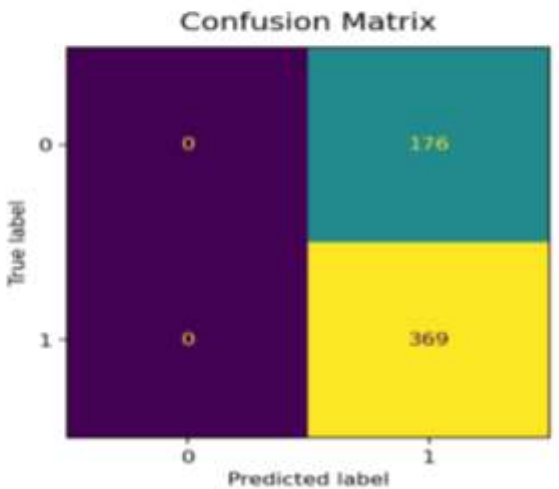
[17]: <Axes: >




```
ROC-AUC:    0.88275
Accuracy:   0.83303
Loss:       0.43996
```

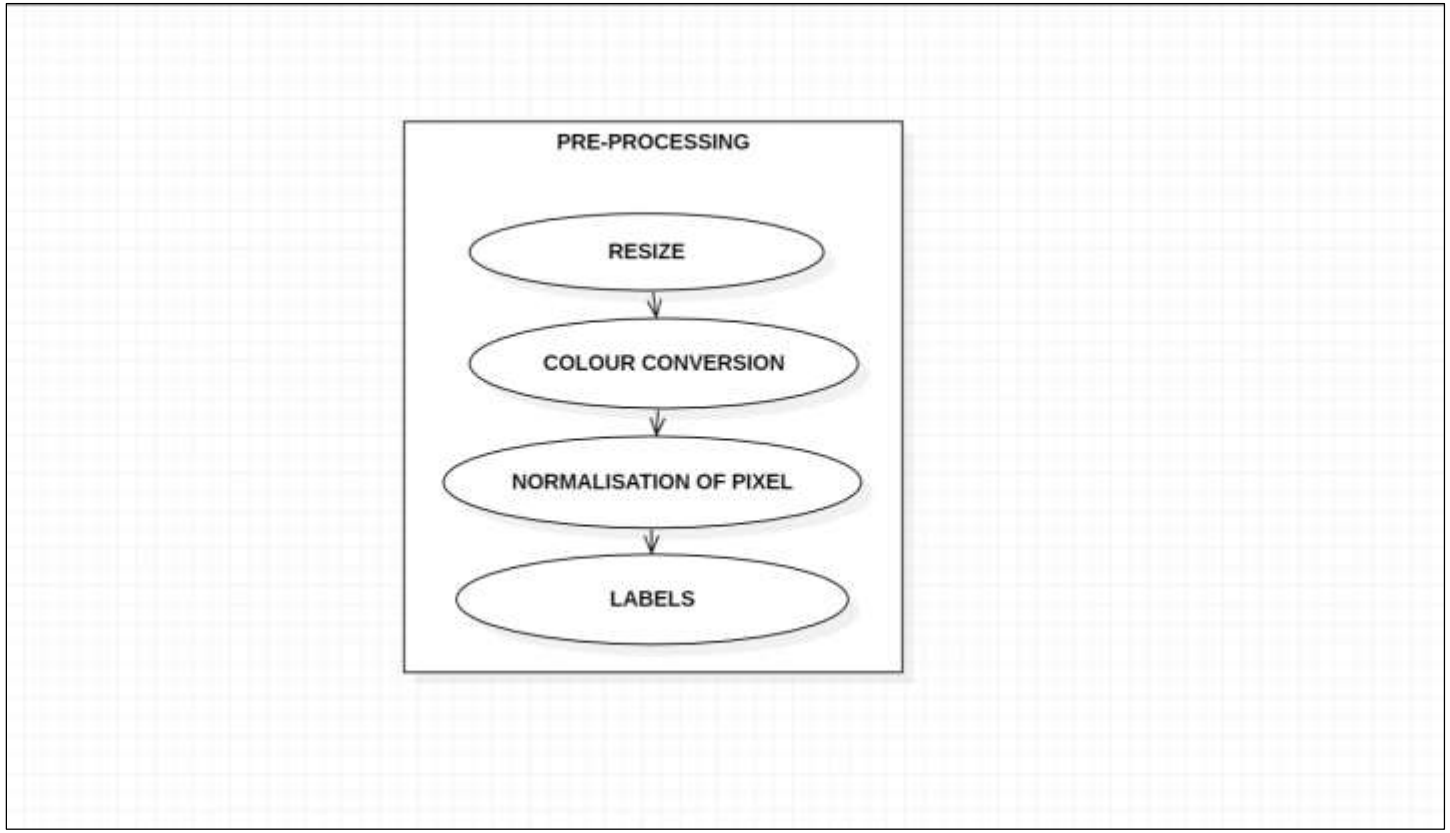






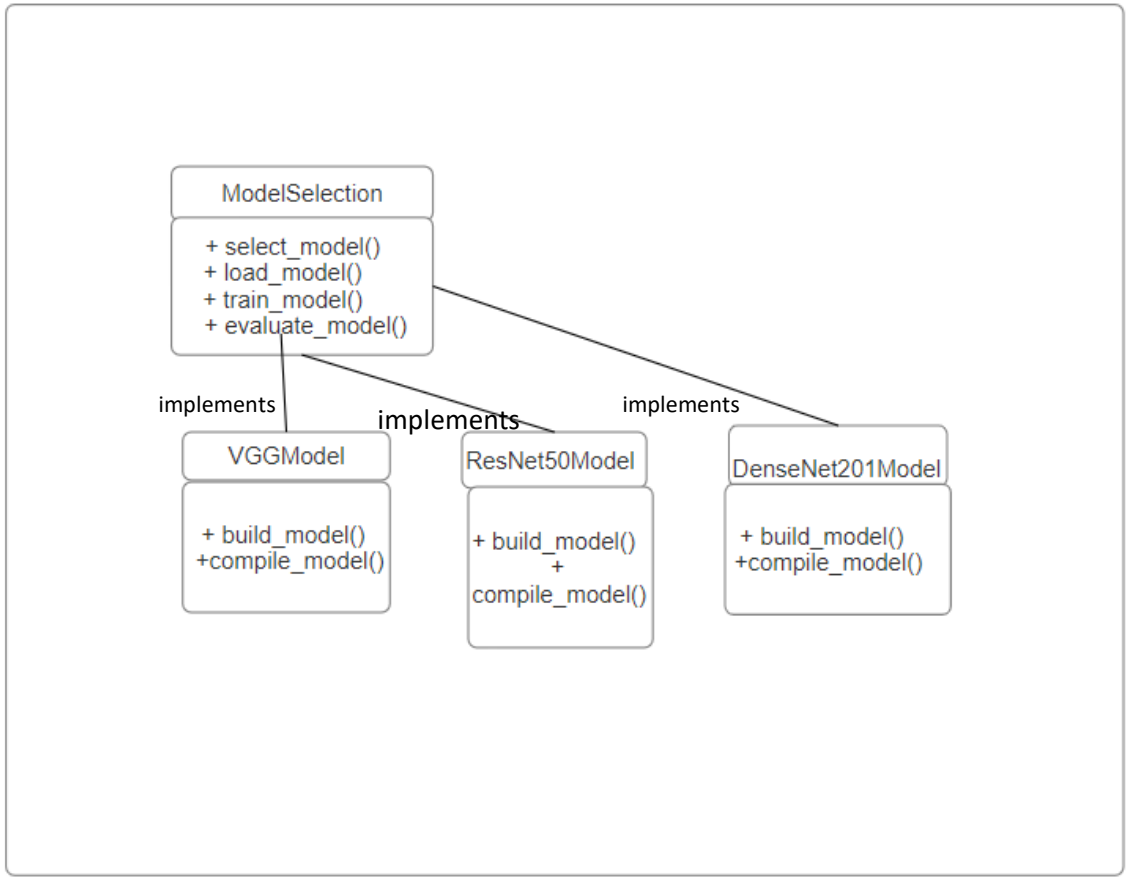
UML Diagrams:

Module -1 : Pre-Processing

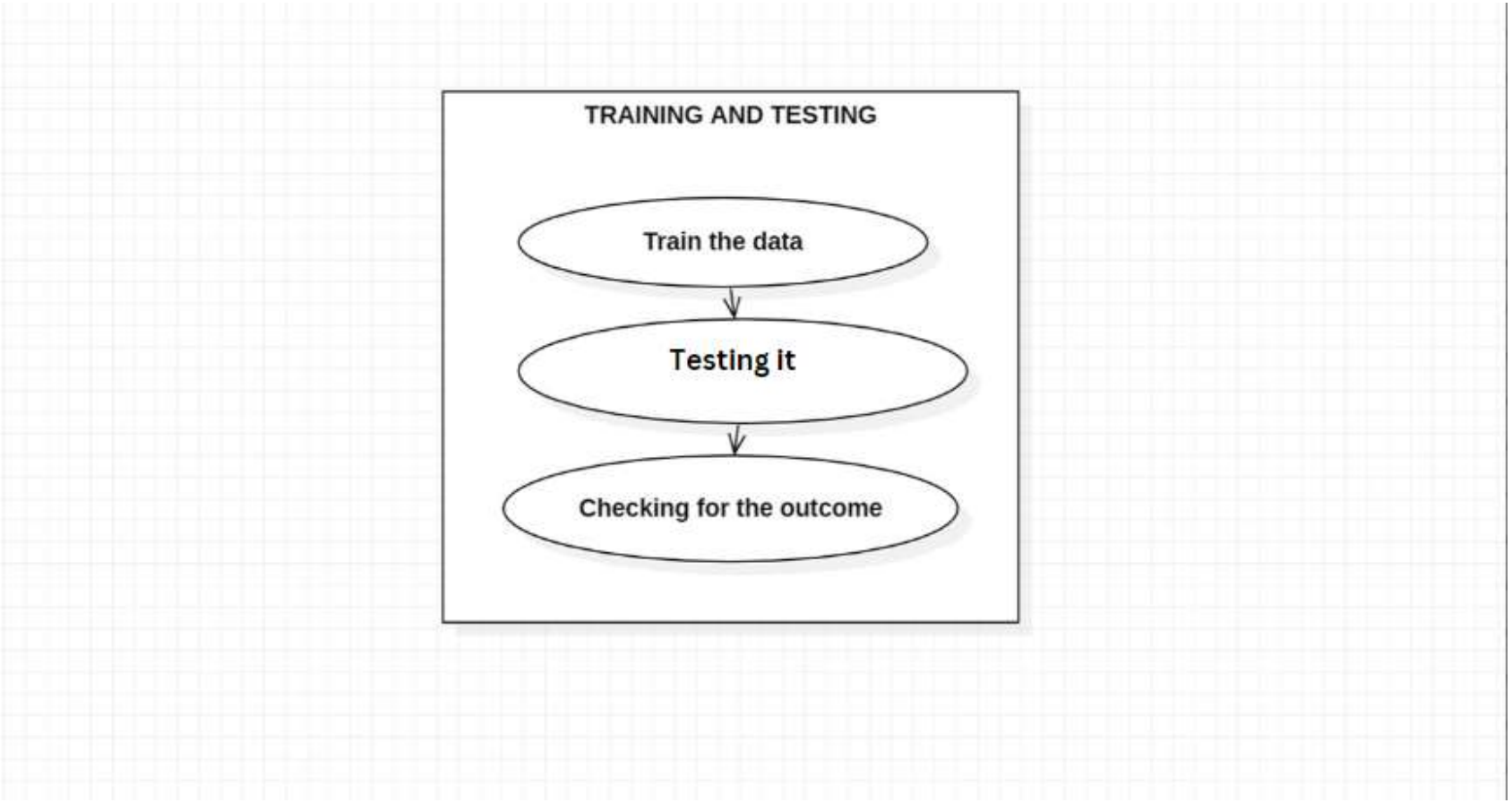


UML Diagrams:

Module -2 : Model Selection



Module-3 :Testing and Training



Scope Limitations:

Data Availability and Quality: The success of machine learning models heavily relies on the quality and quantity of data. Limited access to diverse and well-annotated medical images can be a significant constraint.

Computational Resources: Training deep learning models, especially on large medical image datasets, can be computationally intensive. Limited access to high-performance computing resources may impact the scale and speed of your project.

Ethical and Legal Considerations: Handling medical data involves strict ethical and legal considerations. Compliance with regulations such as HIPAA (Health Insurance Portability and Accountability Act) and ensuring patient privacy can impose limitations on data sharing and processing.

Real-world Generalization: Achieving a high accuracy within the training dataset doesn't always guarantee successful generalization to real-world scenarios. The model's performance on unseen data, especially from different healthcare institutions, is a challenge.

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] Amrane, Meriem, et al. "Breast cancer classification using machine learning." 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT). IEEE, 2018.

[3] Sarosa, Syam Julio A., Fitri Utaminingrum, and Fitra A. Bachtiar. "Mammogram breast cancer classification using gray-level co-occurrence matrix and support vector machine." 2018 international conference on sustainable information engineering and technology (SIET). IEEE, 2018.

[4] Zebari, Dilovan Asaad, et al. "Systematic review of computing approaches for breast cancer detection based computer aided diagnosis using mammogram images." Applied Artificial Intelligence 35.15 (2021): 2157-2203.

[5] Nawaz, Majid, Adel A. Sewissy, and Taysir Hassan A. Soliman. "Multi-class breast cancer classification using deep learning convolutional neural network." Int. J. Adv. Comput. Sci. Appl 9.6 (2018): 316-332.

Review Paper

Introduction:

Breast cancer is a major global health concern, and early detection is crucial for improving patient outcomes and reducing treatment costs.Deep learning models have demonstrated impressive performance in classifying cancerous to non-cancerous tumors. These models mimic the human brain's ability to process large amounts of data and computation power, outperforming humans in tasks like object recognition, image segmentation, and face recognition. CNNs, particularly in the biomedical field, have shown remarkable progress in computer vision, making them useful for tasks like histology image classification.

Literature review:

In previous research done till now there where different imaging modalities as Mammography, MRI, Ultrasound, Tomography and Histopathology were compared as to which would be better for image diagnosis through various parameters.Different techniques like SVM,Random Forest, Naïve Bayes and KNN,ANN and Models like ResNeXt, Dual Path Net, SENet, and NASNet were compared for classification and reviewed incontext of their performance.

Problem Identification and Challenges

Problem Identification:

Breast cancer diagnosis is traditionally reliant on mammography and histopathology, which are subject to limitations such as false positives and interobserver variability. The introduction of machine learning aims to address these limitations, offering automated, objective, and potentially more accurate diagnostic tools.

Challenges:
Data Quality: Obtaining diverse and high-quality datasets for model training is challenging.
Interpretability: Complex machine learning models need to be interpretable for clinical use.
Clinical Validation: Models must be rigorously validated on diverse patient populations.
Ethical Concerns: Data privacy and ethical regulations require strict adherence.

Proposed Methodology:
This section presents the proposed approach for breast cancer detection, emphasizing the utilization of deep learning CNN models which are Inception V3,VGG16, ResNet-50,and compare them with the proposed DenseNet-201 model on various parameters.Comparative analysis is done to demonstrate its validity.

Experimental Evaluation:
A rigorous experimental evaluation is conducted to assess the performance of the proposed approach. BreakHis 400X dataset is utilized to validate the effectiveness of the Densenet201 Architecture in detecting Breast Cancer. Performance metrics are employed for comprehensive evaluation.

Discussion:
The discussion section provides a comprehensive reflection on the project's outcomes, encompassing model performance, future prospects, and the overarching focus on improving patient care in breast cancer diagnosis.

Conclusion:

In conclusion, the project demonstrates the potential of machine learning in breast cancer diagnosis, highlighting significant improvements in accuracy and efficiency. The ethical considerations and resource optimization strategies ensure responsible and accessible deployment. As we move forward, further research and development will continue to enhance the model's capabilities, ultimately contributing to improved patient care in breast cancer diagnosis.

References:

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.

Amrane, Meriem, et al. "Breast cancer classification using machine learning." 2018 Electric Electronics, Computer Science, Biomedical Engineering's' Meeting (EBBT). IEEE, 2018.

Zebari, Dilovan Asaad, et al. "Systematic review of computing approaches for breast cancer detection based computer aided diagnosis using mammogram images." Applied Artificial Intelligence 35.15 (2021): 2157-2203.

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.

Conclusion

There are various models present in Machine Learning which we can use for breast cancer classification. We analysed the models and studied their characteristics. We also studied different research papers in order to choose our model(Densenet201) or technique.We compare different CNN models on our dataset and analyse their accuracy. The proposed project harnesses the power of machine learning and deep learning techniques to revolutionize breast cancer detection and diagnosis. By leveraging curated datasets of histopathology images,meticulous preprocessing, training and testing, the project aims to create a sophisticated deep learning model capable of accurately classifying breast cancer cases.

Thank You !