TumorTrace: MRI-Based AI for Breast Cancer Detection: Documentation

1. Introduction

- **Background**: Brief about breast cancer detection challenges and the role of MRI imaging.
- **Project Goals**: Objectives of TumorTrace in leveraging AI for MRI-based cancer detection.
- **Overview of Workflow**: Summ
-
-
-
-
-
- ary of the steps involved: data collection, EDA, preprocessing, model training, evaluation, and deployment.

2. Exploratory Data Analysis (EDA)

- **Data Overview**
  - Details about the dataset (source, size, format).
  - Description of classes: tumor vs. no tumor.
  - Initial checks on data integrity (missing files, corrupt images).
- **Data Inspection**
  - Visualization of class distributions using bar charts.
  - Sample images from each class with descriptive labels.

  Example:

  - Tumor class image with annotations highlighting MRI-specific features.
  - No tumor class image with healthy tissue markings.
- **Preprocessing and Augmentation**
  - Resizing images to 224x224 to fit the input requirements of the neural network.

- Normalization using ImageNet statistics for faster convergence.
- Data augmentation techniques for generating diverse training samples.

```
transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomRotation(30),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485,      0.456,      0.406],
std=[0.229, 0.224, 0.225])
])
```

3. Model Training and Evaluation

- **Architecture Selection**
  - Why VGG16 was chosen (balance of simplicity and accuracy for image classification).
  - Explanation of custom modifications to suit binary classification.
- **Training Configuration**
  - Loss Function: Explanation of cross-entropy loss and why it's suitable for classification tasks.
  - Optimizer: Use of Adam optimizer and its parameters (learning rate, betas).
  - Batch Size and Epochs: Reasoning behind parameter choices.

  Example Code:

  ```
  criterion = nn.CrossEntropyLoss()
  optimizer = torch.optim.Adam(model.parameters(), lr=0.0001,
  weight_decay=1e-5)
  ```

- **Evaluation Metrics**
  - Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC, with detailed formulas and usage scenarios.

    - Visualizations: Confusion matrix and ROC curve to demonstrate performance.
- **Hyperparameter Tuning**
    - Grid search and validation to find the optimal learning rate, dropout, and batch size.
    - Observations from tuning: Improvements in precision/recall.
- **Results and Insights**
    - Quantitative results: Metrics on the test set.
    - Qualitative insights: Misclassified samples and possible reasons.

4. Model Development

Pretrained Model Customization

A pretrained **VGG16** model is used as the base, modified to adapt to the custom classification task. The model is tailored to handle two classes with the following adjustments:

- Replaced the fully connected layers to match the dataset's output requirements.
- Added dropout for regularization and ReLU activation functions for non-linearity.

Early Stopping

Implemented a custom EarlyStopping class to prevent overfitting by monitoring validation loss:

- Saves the model when a significant improvement in validation loss is detected.
- Stops training after a specified patience period.

Optimization

Training leverags:

- **Loss Function:** CrossEntropyLoss for classification tasks.
- **Optimizer:** SGD/Adam for gradient-based optimization.

- **Data Augmentation:** Enhancements like random flips, rotations, and normalization during preprocessing.

5. Deployment using Gradio

The final trained model is deployed using **Gradio**, allowing an interactive web interface:

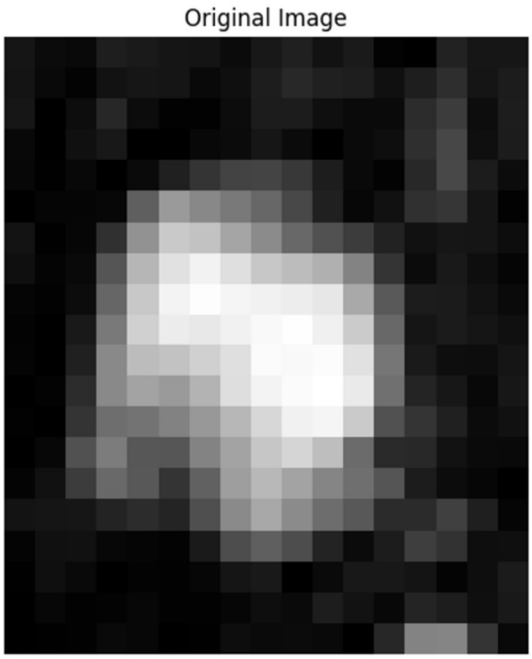
- **Input:** Users upload MRI images.
- **Output:** The model predicts and displays class probabilities or labels.

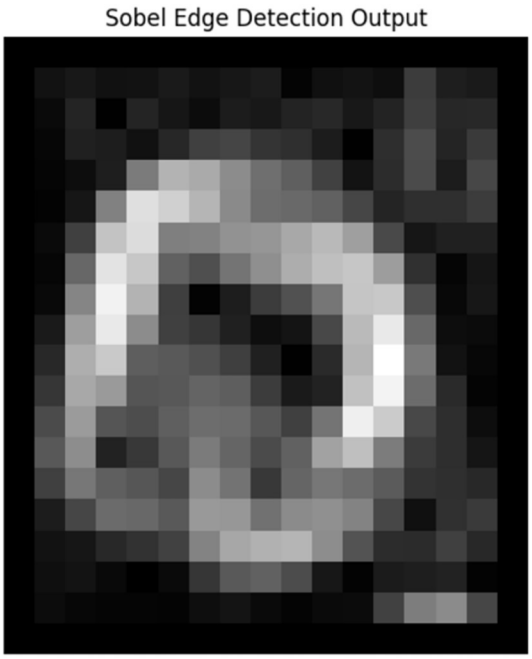
Key Features:

- Real-time predictions with visual feedback.
- Easy-to-use interface for both developers and end-users.

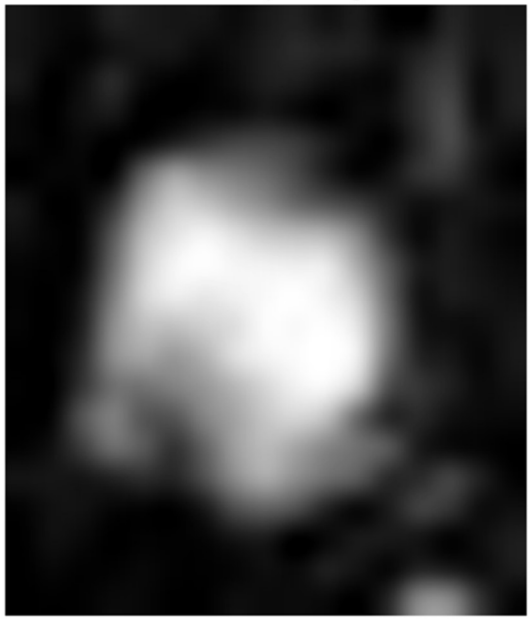
Results:

```
Sobel edge detection output:
 [[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0  18  24  17  18  26  18  23  28   4  16  19  14  61  30  26   0]
 [  0  10  37   1  36  22  12  28  24  35  40  24  35  63  39  40   0]
 [  0   6  32  29  17  39  64  69  53  47  28   1  46  76  37  58   0]
 [  0   5  13  30 123 179 170 140 111  95  65  19  45  75  28  71   0]
 [  0   3  22 129 223 208 181 138 109 106  98  70  37  47  47  60   0]
 [  0  10  64 195 220 127 130 146 150 168 185 159  74  22  33  32   0]
 [  0   6 103 227 199  97  79 116 143 173 191 198 157  49   5  21   0]
 [  0   9 133 242 179  63   3  29  61  82 118 197 200  78   8  23   0]
 [  0  26 158 233 140  64  54  33  14  20  72 186 233 104  13  11   0]
 [  0  40 174 201  94  89  80  63  32   1  42 181 255 122  18   7   0]
 [  0  54 168 154  85  88 100  94  60  26  34 193 243 108  42   4   0]
 [  0  75 155  85  77  95 109 107  87  64 116 239 203  72  46  13   0]
 [  0  88 141  34  56  88 123 101  75  97 162 191 124  59  46  21   0]
 [  0  64 119  95  86  70 140 115  56 101 119 108  91  51  47  40   0]
 [  0  28  70 107 104  89 153 151 113 139 144 131  71  16  49  58   0]
 [  0  18  22  40  50  65 131 167 177 181 141  83  44  43  64  41   0]
 [  0  15  19   9   0  10  54  89  98  76  23   3  27  30  34   3   0]
 [  0  12   7   5   5   4  14  20  10   4  10  11  65 125 139  70   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
Sobel edge detection output saved as 'sobel_edge_detection_output.png'
```

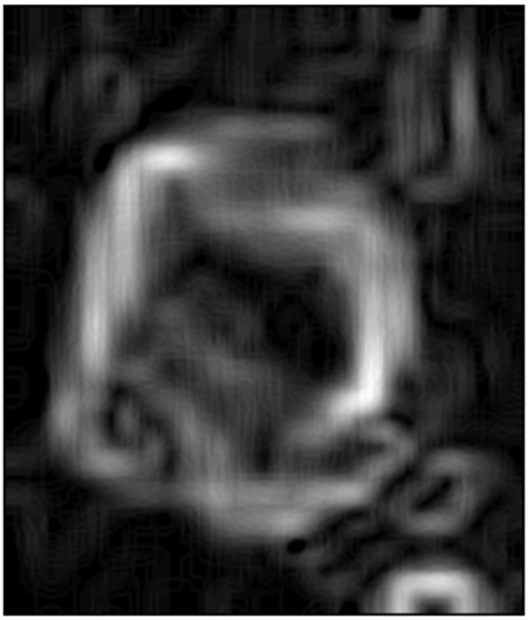




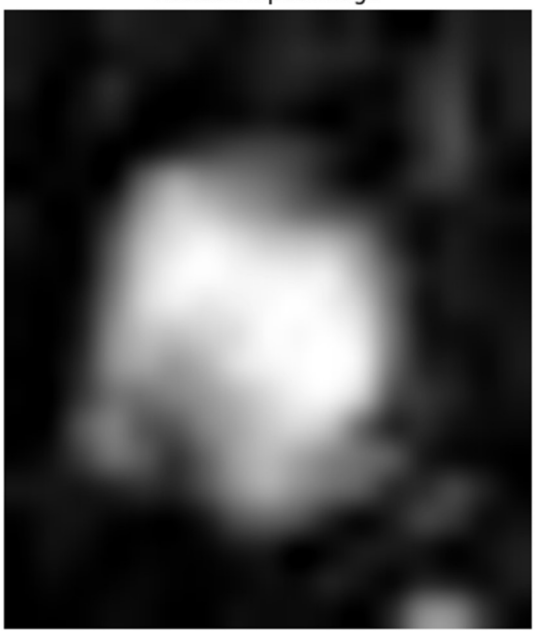
Resized Input Image



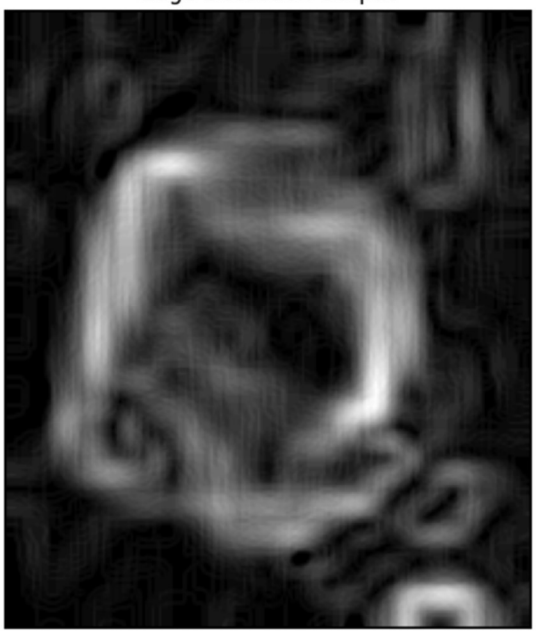
Edge Detection Result

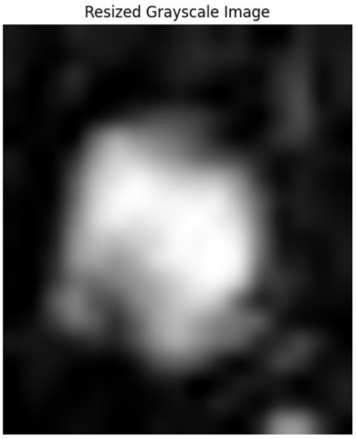


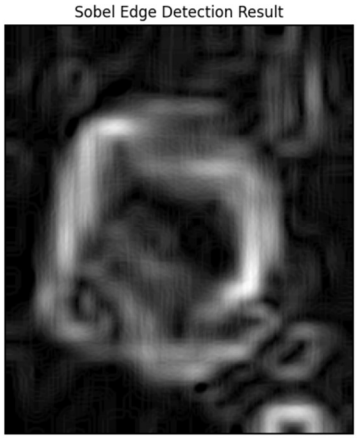
Resized Input Image



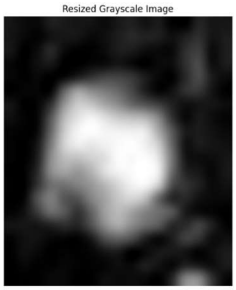
Edge Detection Output

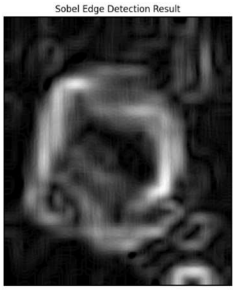



Resized Grayscale Image

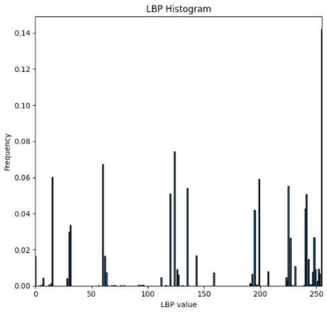

Sobel Edge Detection Result

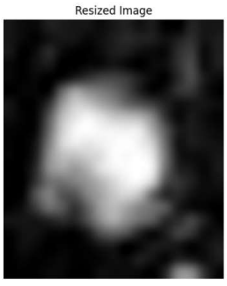

LBP Result

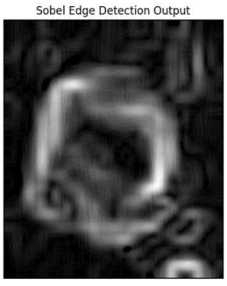

Resized Grayscale Image

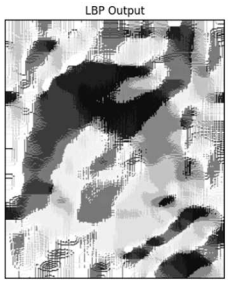

Sobel Edge Detection Result

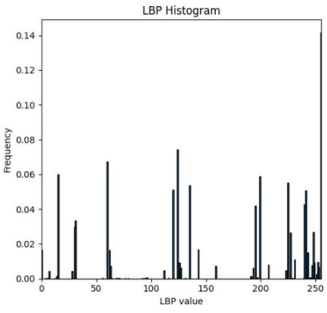

LBP Result


LBP Histogram
Frequency
LBP value
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0
50
100
150
200
250


Resized Image


Sobel Edge Detection Output

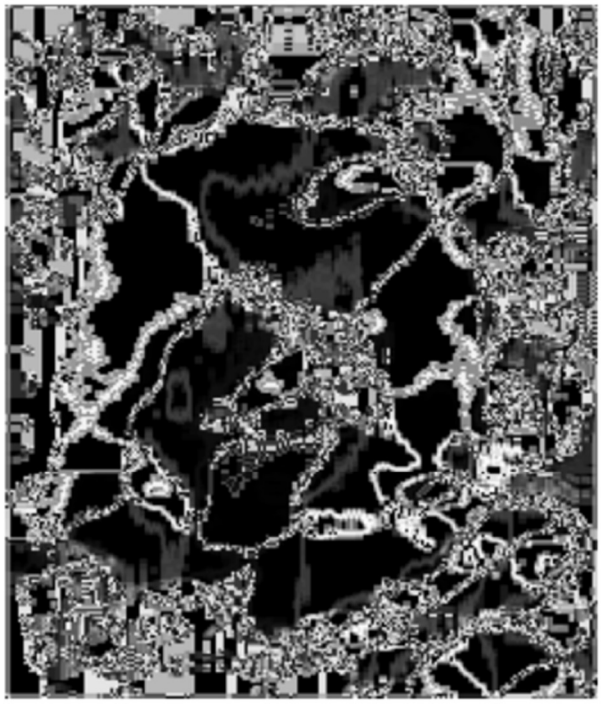

LBP Output


LBP Histogram
Frequency
LBP value
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0
50
100
150
200
250

LBP Image

LBP Mean-based Image

LBP Variance-based Image



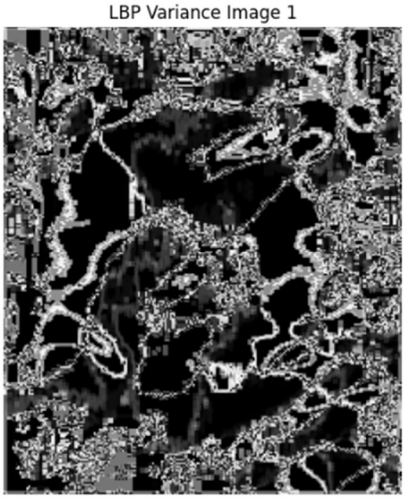
LBP Median-based Image

Original Image 1

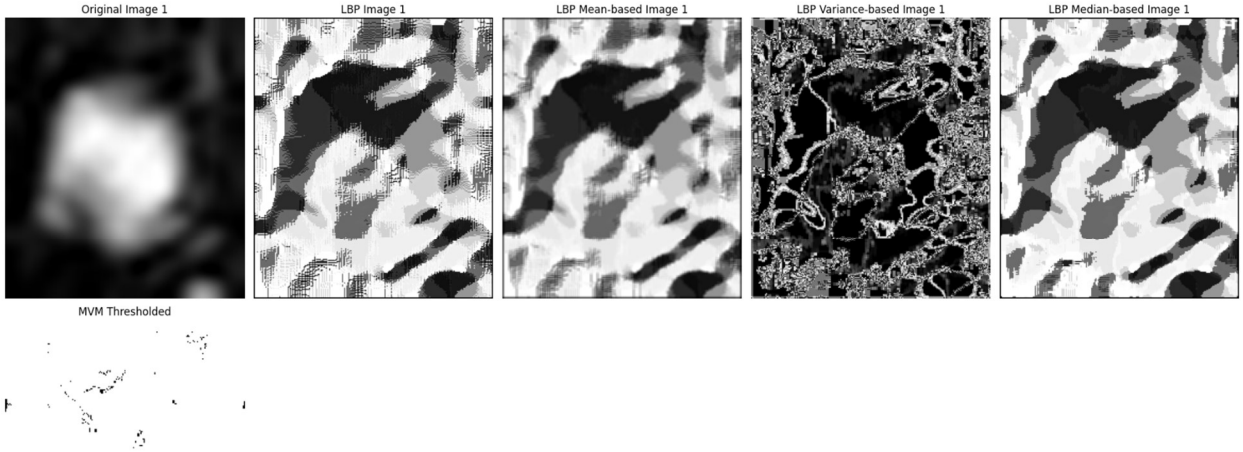
LBP Image 1

LBP Mean Image 1

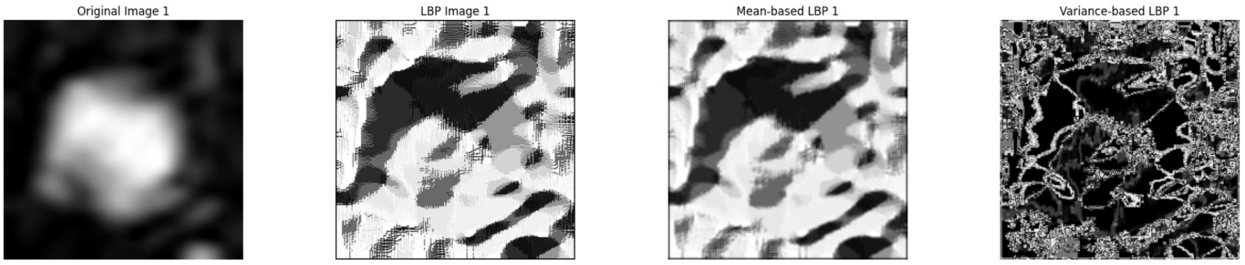
LBP Variance Image 1

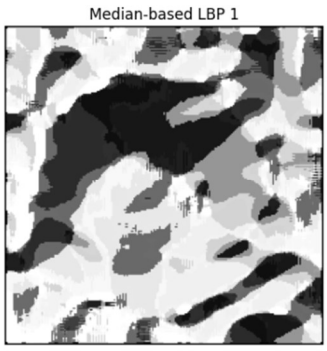


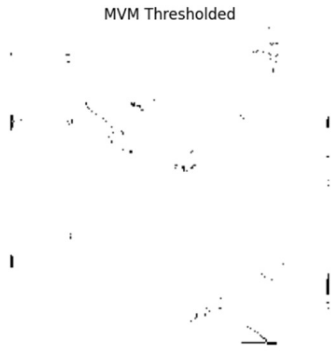
LBP Median Image 1

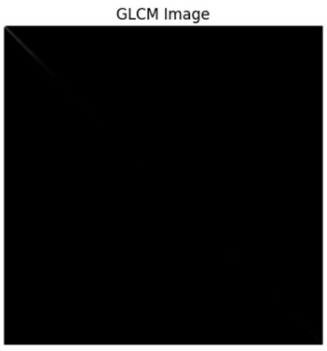

Original Image 1
LBP Image 1
LBP Mean-based Image 1
LBP Variance-based Image 1
LBP Median-based Image 1

MVM Thresholded


Original Image 1
LBP Image 1
Mean-based LBP 1
Variance-based LBP 1


Median-based LBP 1


MVM Thresholded


GLCM Image