Final Project Report

Project Title : Deep Learning Techniques for Breast Cancer Risk Prediction

10	**28/11/2024**	**10 . Appendix** **10.1 Source Code** **10.2 GitHub & Project Demo Link**

1. Introduction :

a) Deep Learning :

- In the fast-evolving era of artificial intelligence, Deep Learning stands as a cornerstone technology, revolutionising how machines understand, learn, and interact with complex data.
- At its essence, Deep Learning AI mimics the intricate neural networks of the human brain, enabling computers to autonomously discover patterns and make decisions from vast amounts of unstructured data.
- This transformative field has propelled breakthroughs across various domains, from computer vision and natural language processing to healthcare diagnostics and autonomous driving.

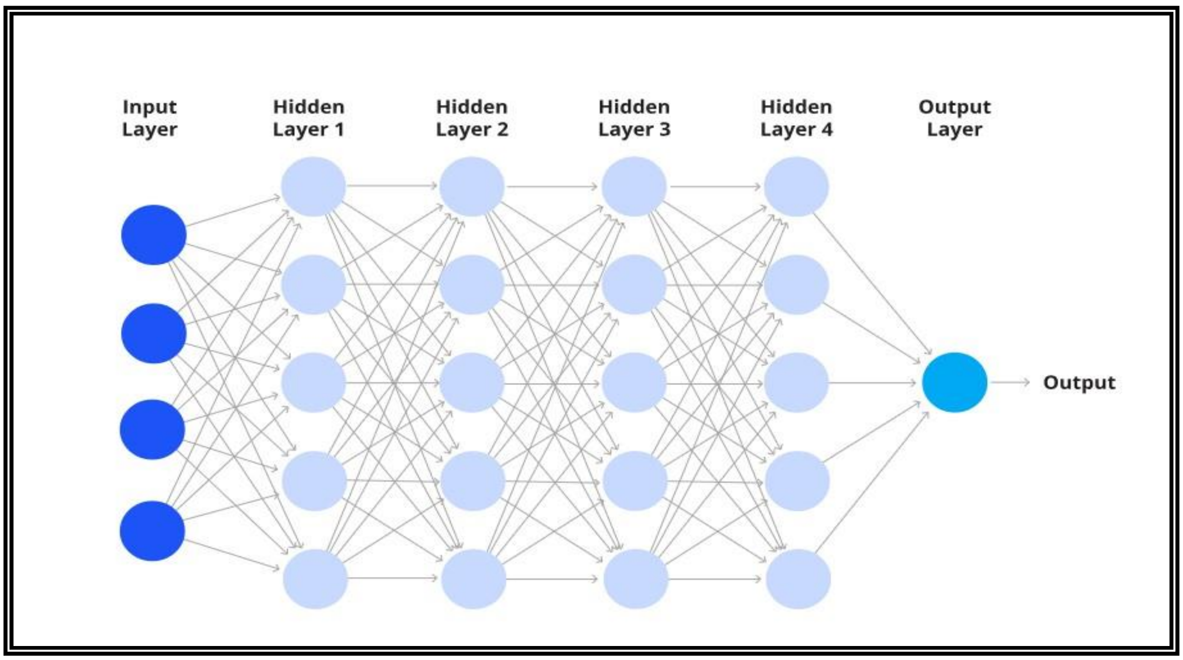


Fig 1 - Working of Deep Neural Networks

b) Types of Deep Learning Approaches/Techniques :

1) Convolutional Neural Networks (CNNs) :

- Designed for image and video analysis.

- Utilize convolutional layers to extract features from images, making them ideal for tasks like object recognition, image classification, and image segmentation.

2) Recurrent Neural Networks (RNNs) :

- Process sequential data, such as time series or natural language.
- Use recurrent connections to remember past information, making them suitable for tasks like language translation, speech recognition, and text generation.

3) Generative Adversarial Networks (GANs) :

- Comprise two neural networks: a generator and a discriminator.
- The generator creates new data samples, while the discriminator evaluates their authenticity.
- Used for tasks like image generation, style transfer, and data augmentation.

4) Long Short-Term Memory (LSTM) Networks :

- A special type of RNN capable of learning long-term dependencies.
- Used for tasks that require remembering information over extended periods, like language modelling and machine translation.

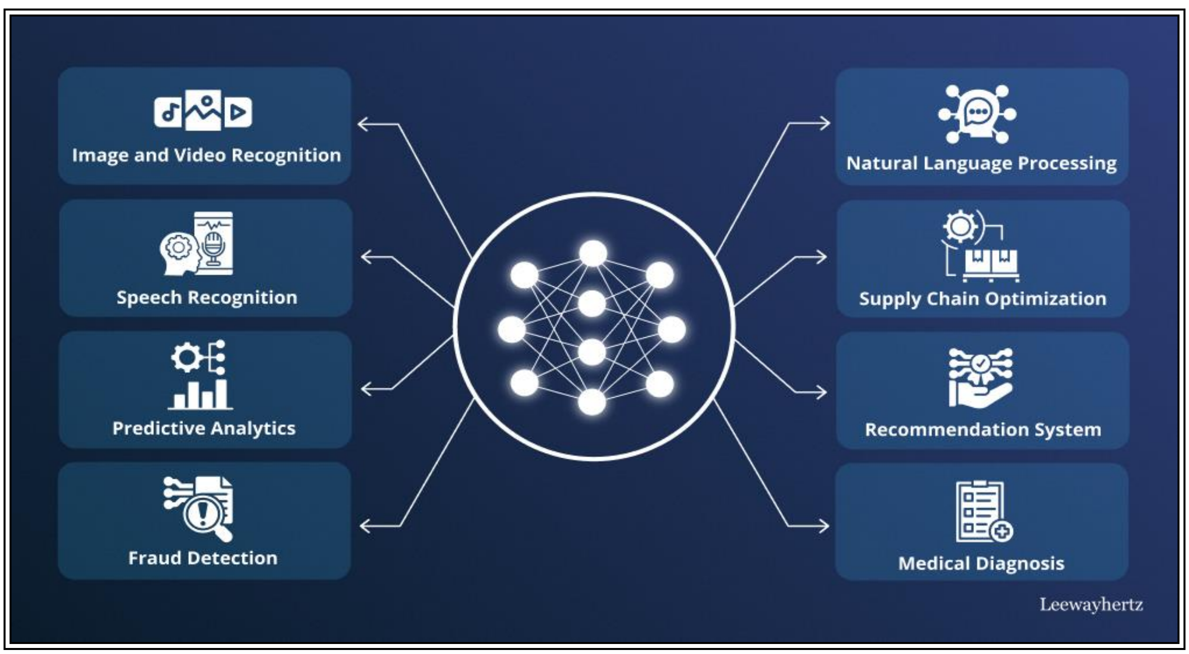


Fig 2 - Usage of Deep Learning in various fields

1.1 Project Overview :

- Breast cancer remains one of the most common and fatal cancers globally, necessitating the development of advanced diagnostic tools.
- This project leverages deep learning techniques, particularly Convolutional Neural Networks (CNNs), to predict breast cancer risks from histopathology images.
- The system aims to assist in early detection, leading to improved patient outcomes.

1.2 Objectives :

The primary objective of using Convolutional Neural Networks (CNNs) for breast cancer risk prediction is to develop a robust and accurate model that can aid in early detection and diagnosis. Some specific objectives are as follows :

1. **Accurate Risk Prediction :** Develop a CNN model capable of accurately predicting the risk of developing breast cancer based on mammogram images.

2. **Feature Extraction :** Identify subtle patterns that may not be easily detectable by human radiologists.

3. **Early Detection :** Improve early detection of breast cancer by identifying high-risk individuals who may benefit from more frequent screening , reducing the mortality rate.

4. **Personalized Risk Assessment :** Develop a personalized risk assessment tool that considers individual factors like age, family history, and genetic predisposition.

5. **Clinical Integration :** Integrate the CNN model into clinical workflows to assist radiologists in decision-making , Developing user-friendly interfaces to facilitate the adoption of the technology.

2. Project Initialization and Planning Phase :

2.1 Define Problem Statement :

Breast cancer is a leading cause of cancer-related deaths among women globally. Breast cancer diagnosis often relies on manual assessment, which is prone to error and inefficiency. In underserved regions, the lack of advanced tools limits early detection. This project addresses these issues by providing an automated deep learning-based system for accurate risk prediction .

Key considerations for the problem include :

The model must handle the variability in imaging data (e.g., resolution, contrast, noise) and account for differences across populations .

The system should aim for high sensitivity and specificity to reduce false positives and negatives .

The solution should integrate seamlessly into clinical workflows, providing interpretable and actionable results .

2.2 Project Proposal (Proposed Solution) :

To address the problem of breast cancer risk prediction, we propose a deep learningbased solution utilising **Convolutional Neural Networks (CNNs)**. CNNs are well-suited for medical imaging tasks due to their ability to automatically extract hierarchical features, capturing both low-level patterns and high-level structural information critical for disease diagnosis.

> Proposed Solution :

1. Data Collection and Preprocessing :

a) Datasets : Using publicly available datasets such as :

- Mammographic Image Analysis Society (MIAS) dataset
- Histopathological images from Breast Cancer Histology Challenge (BreakHis)

b) Data Preprocessing :

- Normalize images to ensure consistent intensity values.
- Augment data with techniques such as rotation, flipping, zooming, and contrast adjustment to handle class imbalances and increase generalizability.

2. Model Architecture :

a) Base Architecture :

- Employ state-of-the-art CNN architectures such as ResNet, InceptionNet, or EfficientNet.
- Alternatively, design a custom CNN architecture tailored to the specifics of breast imaging.

b) Input Layers:

- Utilize 2D greyscale or RGB channels for mammograms and ultrasound scans.
- Use 3D CNNs for volumetric imaging (e.g., MRI scans) if applicable.

c) Feature Extraction:

■ Leverage convolutional layers to capture spatial and texture-based patterns indicative of malignancy.

d) Classification Layers:

■ Use fully connected layers to output probabilities for risk categories (e.g., benign or malignant). **3. Training Process :**

a) Loss Function:

■ Use a categorical cross-entropy or focal loss function to handle imbalanced classes.

b) Optimization:

■ Train the model using optimizers like Adam or SGD with momentum for faster convergence.

c) Validation:

■ Use k-fold cross-validation to ensure model robustness and reduce over-fitting. **d) Metrics**:

■ Evaluate the model using sensitivity, specificity, F1 score, and area under the ROC curve (AUC).

4. Model Interpretability :

- Integrate techniques such as:

Attention mechanisms to highlight critical features driving predictions. **5.**

Deployment and Clinical Integration :

a) Deployment :

■ Develop a user-friendly interface for clinicians to upload and analyse images. **b) Integration** :

■ Ensure compliance with healthcare data privacy regulations (e.g., HIPAA, GDPR).

2.3 Initial Project Planning :

Sprint Planning : The project was divided into phases , including data collection , preprocessing , model building , and deployment.

a) Dataset Details --

Dataset : Breast Histopathology Images

Source : Kaggle

File Size : The dataset is approximately 3 GB in size .

Details : Total **277,524 labelled images** in PNG format, categorised as benign (class 0) or malignant (class 1) .

b) Image Preprocessing :

Normalization: Scale pixel values to a standard range (e.g., 0 to 1).

Resizing: Resize images to a uniform input size .

c) Model Building :
Design and train the deep learning model to predict breast cancer risk.

Model Architecture:

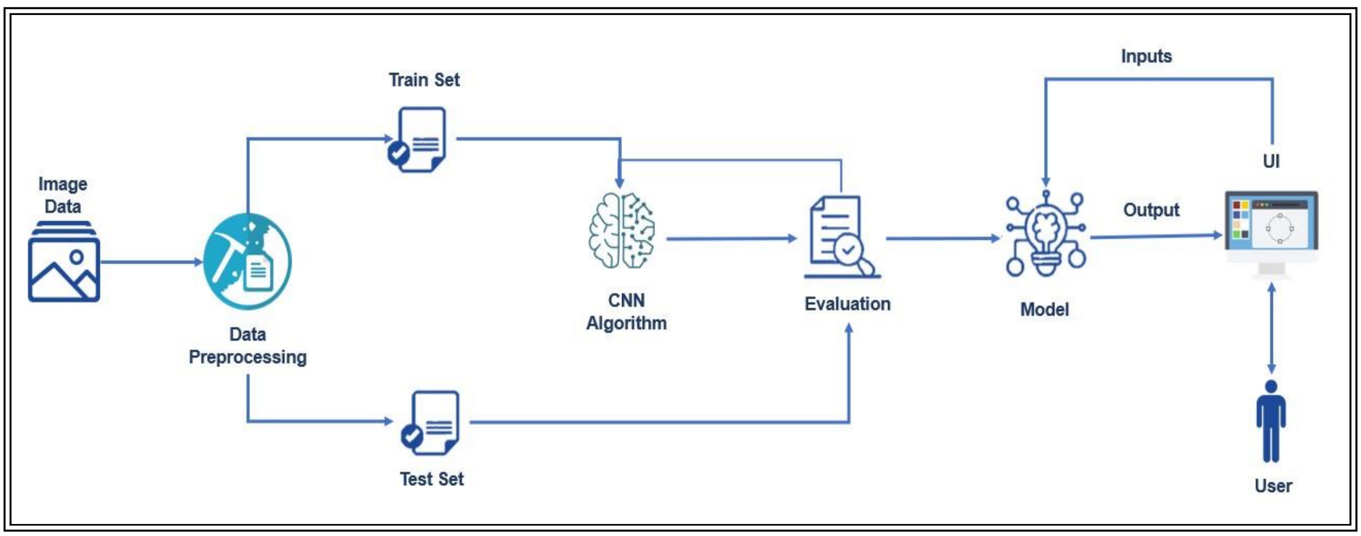


Fig 3 -- Model Architecture of the Project

d) Deployment :
Deploy the trained model in a real-world environment to assist clinicians in predicting breast cancer risk.

- **Key Tasks :**

1) Dataset preparation

2) image augmentation

3) model training

4) optimization.

3. Data Collection and Preprocessing Phase :

3.1 Data Collection Plan and Raw Data Sources Identified :

Dataset : Breast Histopathology Images Dataset from Kaggle.

The dataset is approximately 3 GB in size.

3.2 Data Quality Report Key data issues and resolutions :

• **Incomplete Metadata :** Collaborated with dataset contributors to fill gaps.

• **Class Imbalance :** Used data augmentation and oversampling for minority classes.

• **Duplicate Entries :** Removed duplicates using image hashing techniques. 3.3 Data Preprocessing . • **Steps :** **o** Resizing images to 128x128 pixels. **o** Normalizing pixel

values to [0, 1]. **o** Augmenting data using flipping, rotation, and zooming techniques. **o**

Tools Used: TensorFlow's ImageDataGenerator and Python libraries like NumPy.

4. Model Development Phase

4.1 Model Selection Report

• **Selected Model :** Convolutional Neural Networks (CNNs)

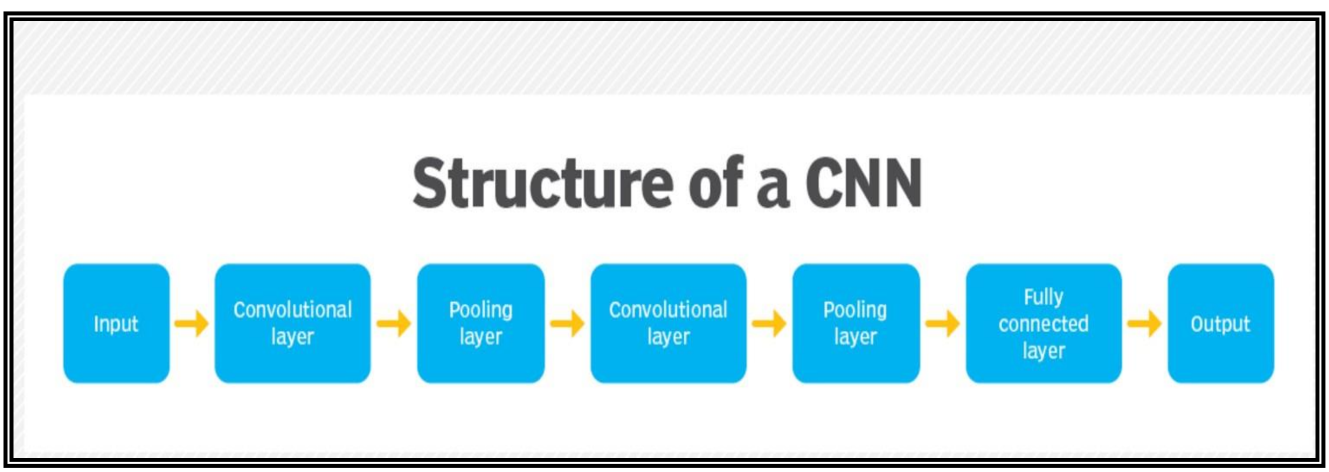


Fig 3 -- Working Of CNN Model

- **Rationale :** Superior performance on image-based tasks, ability to extract complex patterns, and high accuracy in medical image analysis.

4.2 Initial Model Training Code, Model Validation, and Evaluation Report

- **Process followed :**

1) Loaded dataset and applied preprocessing.

2) Built and compiled CNN architecture with convolutional, max-pooling, and dense layers.

3) Achieved training accuracy of 82.88% after 6 epochs.

4) Evaluation: Early stopping was used to prevent over fitting.

5. Model Optimization and Tuning Phase

5.1 Tuning Documentation --

- Hyper parameters Tuned : **o Batch Size :** 32 for balanced computational efficiency and generalisation. **o Epochs :** Set to 20 for comprehensive training. **o Optimizer :** Adam for adaptive learning rates and momentum.

5.2 Final Model Selection Justification --

- There are Various Deep Learning Approaches for building a model but CNN fits perfect as our dataset is in the image (PNG) Format .

- For the breast cancer risk prediction project we have utilised the Convolutional Neural Network (CNN) to train and test the model as it is one of the most efficient deep learning technique for analysing images .

- Convolutional Neural Networks (CNNs) are widely used for analyzing medical imaging data, including mammograms and histopathological images .

- The hierarchical structure of CNNs allows them to learn both low-level and high-level features, making them suitable for identifying subtle differences in breast tissue .

6. Results :

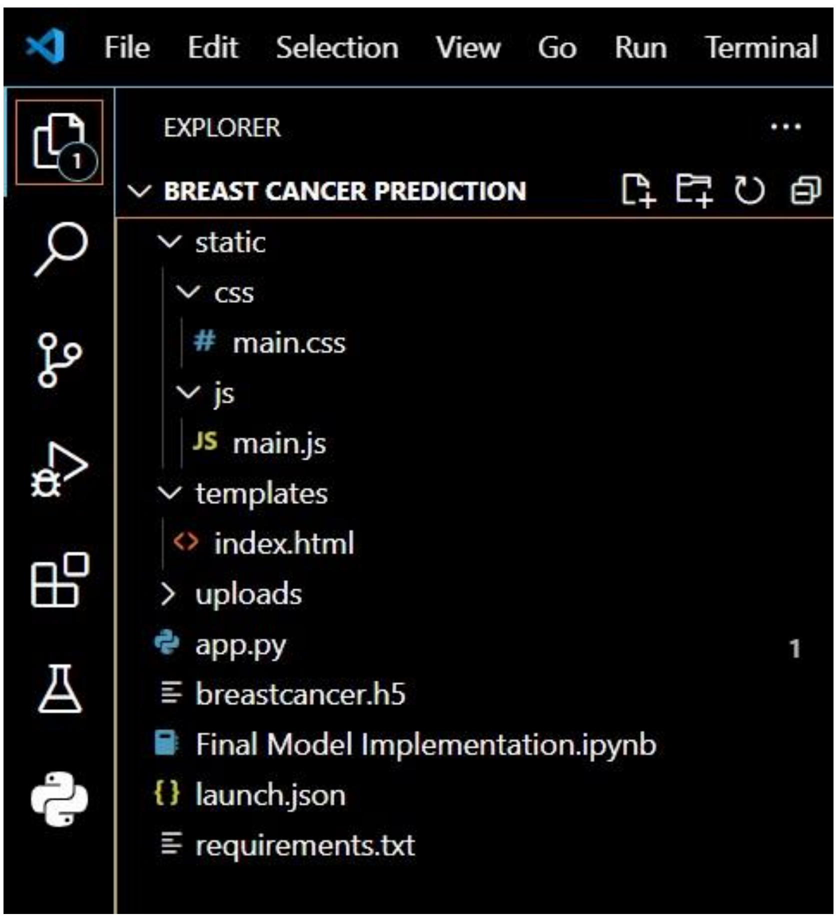
Model Accuracy obtained : 82.88 %

```
# Printing Accuracy
val_loss, val_accuracy = model.evaluate(val_generator)
print(f"Validation Accuracy: {val_accuracy * 100:.2f}%")
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - accuracy: 0.8318 - loss: 0.4277
Validation Accuracy: 82.88%
```

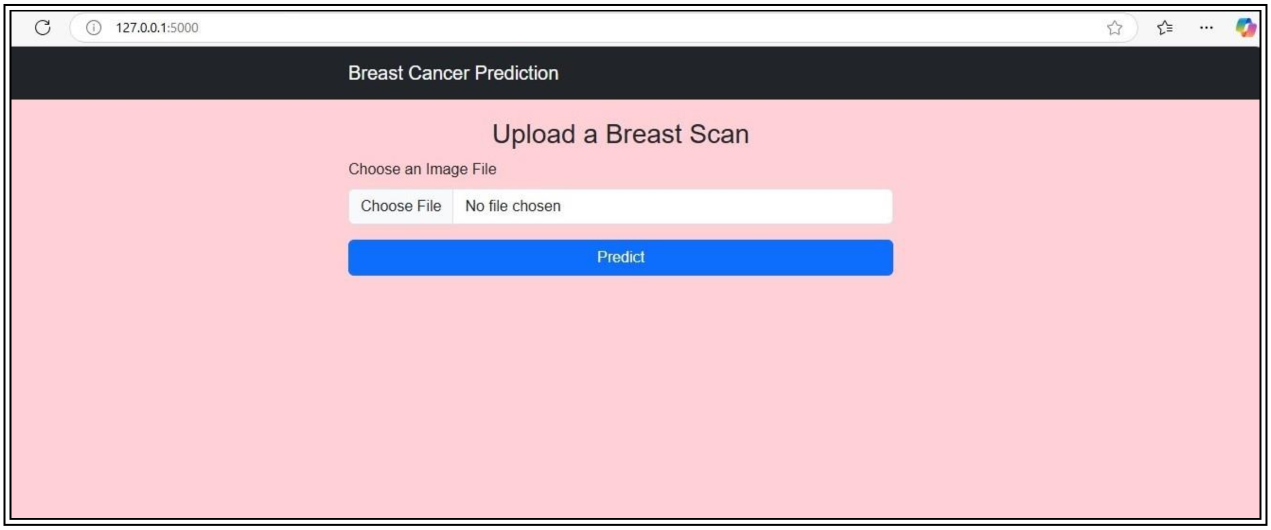
6.1 Output Screenshots :

Flask App : Directory Structure of project in VS Code :

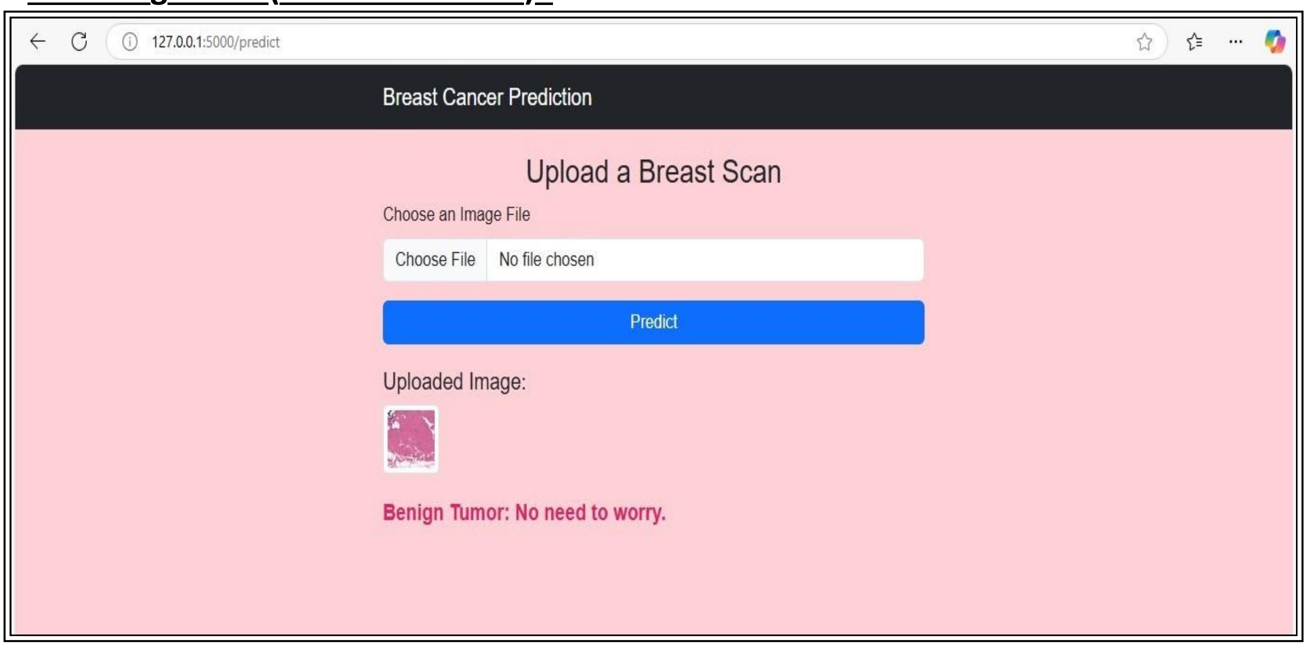


Flask App : User Interface (UI)

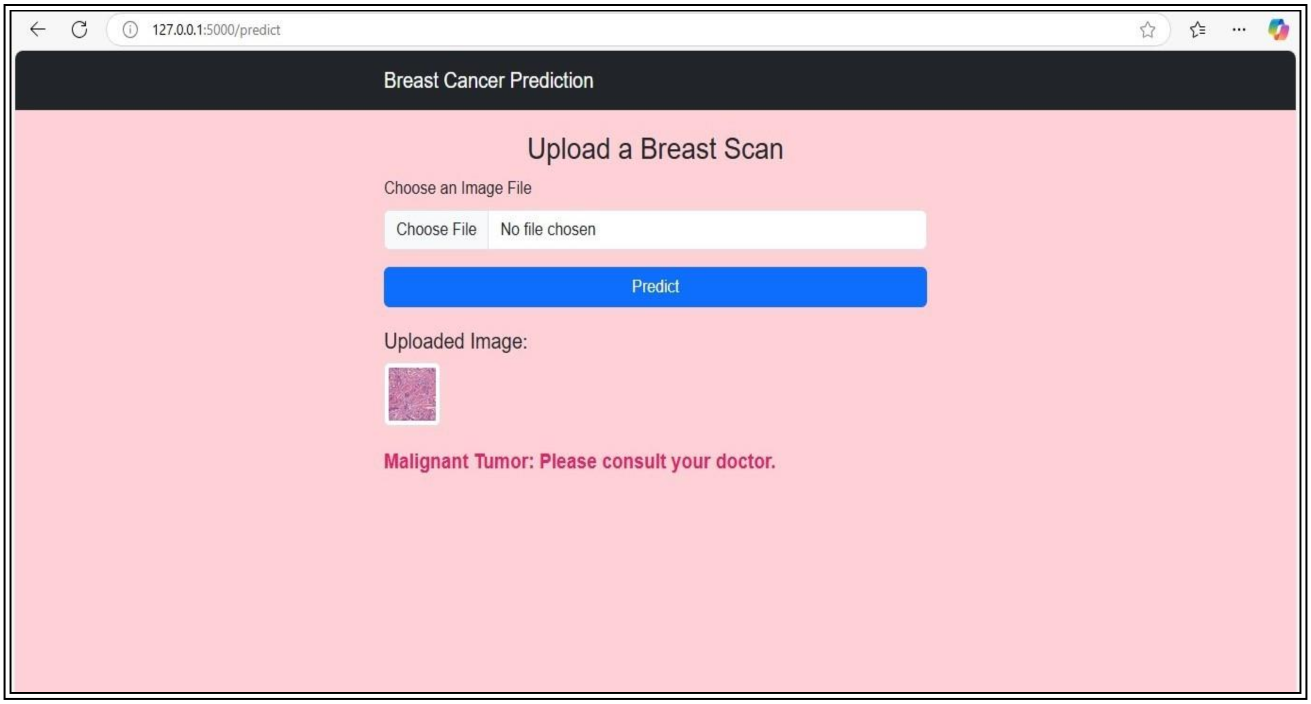
Home Page :



For Benign case (Non - Cancerous) :



For Malignant case (Cancerous) :



7. Advantages & Disadvantages

7.1 Advantages --

The advantages of using CNN model are listed below :

1) Automatic Feature Extraction: CNNs automatically learn hierarchical features from imaging data, eliminating the need for manual feature engineering.

2) High Accuracy: They excel in capturing spatial and texture patterns, leading to high predictive accuracy in medical imaging tasks.

3) Scalability: CNNs can handle large and complex datasets like mammograms, ensuring robust learning.

4) Adaptability: They can be fine-tuned for different types of imaging modalities, such as ultrasound, MRI, or mammograms.

5) Data Augmentation Compatibility: CNNs effectively leverage augmented datasets, improving performance in scenarios with limited data.

6) Integration with Explainability: Techniques like Grad-CAM enhance interpretability, crucial for trust in medical decision-making.

7) **Transfer Learning Benefits**: Pre-trained CNN models reduce training time and improve performance on specific tasks.

8) **Efficient Pattern Recognition**: They are highly adept at detecting subtle anomalies indicative of breast cancer.

9) **End-to-End Learning**: CNNs enable simultaneous feature extraction and prediction, simplifying the workflow.

10) **Non-Invasive Diagnosis Support**: CNNs facilitate non-invasive early risk prediction, aiding timely intervention.

7.2 Disadvantages --

The disavantages of using CNN model are listed below :

1) **Data Dependency**: CNNs require large, labeled datasets for effective training, which may not always be available in medical domains.

2) **Overfitting Risk**: They can overfit small or imbalanced datasets without proper regularization techniques.

3) **Computational Intensity**: CNNs demand significant computational power and memory, especially for training large models.

4) **Susceptibility to Noise**: CNNs may misinterpret noisy or low-quality imaging data, leading to unreliable predictions.

5) **Bias and Fairness Issues**: They can inherit and amplify biases present in the training data, potentially leading to unfair predictions across demographic groups.

6) **Training Time**: The training process for CNNs can be time-consuming, particularly for deep architectures.

7) **Dependency on Preprocessing**: While robust, CNN performance can be influenced by the quality of image preprocessing, such as normalization and resizing.

8) **Difficulty in Generalization**: CNNs trained on specific datasets may struggle to generalize well to unseen data or different imaging devices.

8. Conclusion :

In conclusion, the application of convolutional neural networks (CNNs) for breast cancer risk prediction demonstrates significant potential in enhancing early detection and diagnostic accuracy. By leveraging the hierarchical feature extraction capabilities of CNNs, this approach effectively identifies patterns and anomalies within imaging data, such as mammograms and histopathology images that are critical for assessing risk.

This project demonstrates the feasibility of using deep learning to improve breast cancer risk prediction. With further refinements, the system can be a valuable tool in clinical settings, enabling early detection and personalized care.

Overall, CNNs represent a transformative tool in breast cancer risk prediction, with the potential to significantly improve patient outcomes through timely and accurate diagnosis.

9. Future Scope :

Convolutional Neural Networks (CNNs) have shown immense potential in revolutionizing the field of medical image analysis, particularly in breast cancer detection and risk prediction.

As we move forward, the future of CNN-based breast cancer risk prediction holds promising prospects.

1) Enhancing model interpretability and accuracy through Explainable AI (XAI) :
a) Larger and Diverse Datasets
b) Advanced Architectural Designs **c)** Transfer Learning

2) Incorporating Multimodal Data :
a) Fusing Imaging and Clinical Data
b) Developing Multimodal Fusion Techniques

3) Real-time and Personalized Risk Assessment :
a) Efficient Inference
b) Personalized Risk Models

4) Explainable AI for Clinical Decision Support :
a) Visualizing Feature Importance
b) Interactive Visualization Tools

5) Ethical Considerations and Bias Mitigation :
a) Addressing Bias
b) Transparency and Accountability

10. Appendix :

10.1 Source Code : **Flask App : app.py Code :** from flask

```
import Flask, render_template, request, url_for from PIL

import Image import numpy as np import tensorflow as tf

import os import

uuid

# Initialize the Flask app app =

Flask(__name__)


# Use the existing uploads folder in the static directory UPLOAD_FOLDER =

os.path.join('static', 'uploads')

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER


# Load the model at the start of the application

model_path = 'breastcancer.h5' if

os.path.exists(model_path):

  try:

    print("Loading model...")

    model = tf.keras.models.load_model(model_path)

    print("Model loaded successfully.")

  except Exception as e:

    print(f"Error loading model: {e}")

    model = None else:

  print(f"Model not found at {model_path}.")
```

```
   model = None
@app.route('/') def
index():
   """Render the home page."""
   return render_template('index.html', prediction=None, image_url=None)
@app.route('/predict', methods=['POST'])
def predict():
   """Handle image uploads, make predictions, and display the image."""
   try:
      # Check if an image file was uploaded
      if 'input_file' not in request.files or request.files['input_file'].filename == '':
         return render_template('index.html', prediction="No file selected. Please upload an
image.", image_url=None)
 file = request.files['input_file']
 # Save the image to the upload folder with a unique name
      filename = f"{uuid.uuid4().hex}.jpg"
      filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)        file.save(filepath)

      # Open the image using PIL
      img = Image.open(filepath).convert('RGB')  # Ensure 3 channels (RGB)        img =
img.resize((128, 128))  # Resize to target size

      # Convert the image to a numpy array
      img_array = np.array(img) / 255.0  # Normalize to [0, 1]
      img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension

      # Make prediction
```

```python
        if model:
            prediction = model.predict(img_array)[0][0]
            if prediction > 0.5:
                result = "Malignant Tumor: Please consult your doctor."
            else:
                result = "Benign Tumor: No need to worry."
        else:
            result = "Error: Model not loaded."
       # Generate a relative URL for the uploaded image
        image_url = url_for('static', filename=f'uploads/{filename}')
      # Return the prediction and the uploaded image URL
        return render_template('index.html', prediction=result, image_url=image_url)

    except Exception as e:
        print(f"Error occurred during prediction: {e}")
        return render_template('index.html', prediction=f"Error: {str(e)}", image_url=None)

if __name__ == '__main__':
    app.run(debug=True)
```

Templates : index.html Code :

```html
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Breast Cancer Prediction</title>
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">     <style>

        body {

            background-color: #ffd0d6;

            font-family: Arial, sans-serif;

        }

        .navbar {

            margin-bottom: 20px;

        }

        .container {

            max-width: 600px;

        }

        .preview {

            margin-top: 20px;

        }

        #result {

            font-size: 1.2rem;            font-
weight: bold;            margin-top: 20px;

            color: #e91e63;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-dark bg-dark">

        <div class="container">
```

```
      <a class="navbar-brand" href="#">Breast Cancer Prediction</a>        </div>
  </nav>

  <div class="container">
    <h3 class="text-center">Upload a Breast Scan</h3>
    <form id="upload-form" method="POST" action="/predict" enctype="multipart/formdata">
      <div class="mb-3">
        <label for="input_file" class="form-label">Choose an Image File</label>
        <input class="form-control" type="file" name="input_file" id="input_file"
accept=".jpg,.jpeg,.png" required>
      </div>
      <button type="submit" class="btn btn-primary w-100">Predict</button>        </form>
    {% if image_url %}
    <div class="preview">
      <h5>Uploaded Image:</h5>
      <img src="{{ image_url }}" alt="Uploaded Image" class="img-thumbnail">
    </div>
    {% endif %}
    {% if prediction %}
    <div id="result">{{ prediction }}</div>
    {% endif %}
  </div>
</body>
</html>
```

Static : - main.css Code :

```
.img-preview {
  width: 256px;
```

```
    height: 256px;
    position: relative;
    border: 5px solid #0c0e0f;
    box-shadow: 0px 4px 8px 0px #070f11fe;
    margin-top: 1em;
    margin-bottom: 1em;
}
.img-preview > div {
    width: 100%;
    height: 100%;
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
}
input[type="file"] {
    display: none;
}
.upload-label {
    display: inline-block;
    padding: 12px 30px;
    background: #da5798;
    color: #f2f2f2;
```

```
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}
.upload-label:hover {
    background: #bf5091;
    color: #39D2B4;
}
.loader {
    border: 8px solid #9a2980; /* pink */
    border-top: 8px solid #cd5ca2; /* pink */
    border-radius: 50%;
    width: 3rem;
    height: 3rem;
    animation: spin 1s linear infinite;
}
@-webkit-keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
@keyframes spin {
    0% { transform: rotate(0deg); }
```

```
    100% { transform: rotate(360deg); }
}
```

Static :- main.js Code :

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();
// Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            try {
                reader.onload = function (e) {
                    $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                    $('#imagePreview').hide();
                    $('#imagePreview').fadeIn(650);
                }
                reader.readAsDataURL(input.files[0]);
} catch (error) {
                alert("Error reading file. Please try again.");
                console.error("FileReader error:", error);
```

```
        }
      }
    }
  $("#imageUpload").change(function () {
      var file = this.files[0];
      var fileType = file.type;
      var validTypes = ["image/jpeg", "image/png", "image/jpg"];
      if (!validTypes.includes(fileType)) {
        alert("Invalid file type. Please upload a JPG or PNG image.");
        $(this).val('');
        return;
      }
      if (file.size > 2 * 1024 * 1024) { // Limit size to 2MB
        alert("File size too large. Please upload an image less than 2MB.");
        $(this).val('');
        return;
      }
      $('.image-section').show();
      $('#btn-predict').show();
      $('#result').text('');
      $('#result').hide();
      readURL(this);
```

```
    });

    // Predict

    $('#btn-predict').click(function () {

        if ($('#upload-file').length === 0) {

            alert("Form not found!");

            return ;

}

        var form_data = new FormData($('#upload-file')[0]);

// Show loading animation

        $(this).hide();

        $('.loader').show();

 // Make prediction by calling api /predict

        $.ajax({

            type: 'POST',

            url: '/predict?_=' + new Date().getTime(),

            data: form_data,

            contentType: false,

            cache: false,

            processData: false,

            async: true,

            success: function (data) {

                // Get and display the result
```

```
            $('.loader').hide();
            $('#result').fadeIn(600);
            $('#result').text(' Result:  ' + data);
            console.log('Success!');
        },
        error: function (xhr, status, error) {
            $('.loader').hide();
            alert("An error occurred: " + error);
            $('#btn-predict').show();
            console.error('Error:', error);
        }
    });
  });
});
```

10.2 GitHub & Project Demo Link --

Github Link :

https://drive.google.com/drive/folders/1Z7cRbTSo51pCtRpBsyIsrxCvMXn_P2Fs?usp=drive_link

1) **Project Demo Link (google drive) :** This Folder contains the audio file as well as video file of the project .

https://github.com/sakshishirke0502/Breast_Cancer_prediction-_using_Deep_learning_Techniques